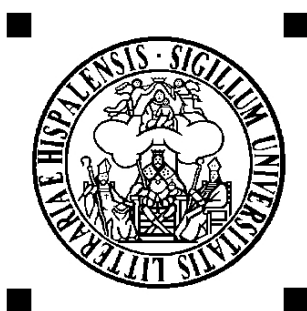


Un Entorno Metodológico de Ingeniería de Requisitos para Sistemas de Información

Departamento de Lenguajes y Sistemas Informáticos

Universidad de Sevilla



Sevilla, septiembre de 2000

**Memoria de la tesis doctoral dirigida por el doctor José Miguel Toro Bonilla
y desarrollada por Amador Durán Toro para optar al grado de
Doctor en Informática por la Universidad de Sevilla**

Don Miguel Toro Bonilla, catedrático del Área de Lenguajes y Sistemas Informáticos,

HACE CONSTAR

que don Amador Durán Toro, Licenciado en Informática por la Universidad de Sevilla, ha realizado bajo mi supervisión el trabajo de investigación titulado

Un Entorno Metodológico de Ingeniería de Requisitos para Sistemas de Información

Una vez revisado, autorizo el comienzo de los trámites para su presentación como Tesis Doctoral al tribunal que ha de juzgarlo.

Fdo. Miguel Toro Bonilla
Catedrático de Universidad
Área de Lenguajes y Sistemas Informáticos

Sevilla, Mayo de 2000

Yo, Amador Durán Toro, con Documento Nacional de Identidad número 27.308.303-S,

DECLARO BAJO JURAMENTO

ser el autor del trabajo que se presenta en la memoria de esta tesis doctoral que tiene por título

**Un Entorno Metodológico de
Ingeniería de Requisitos
para Sistemas de Información**

Fdo. Amador Durán Toro
Profesor Asociado del Dpto. de
Lenguajes y Sistemas Informáticos
de la Universidad de Sevilla
Área de Lenguajes y Sistemas Informáticos

Sevilla, Septiembre de 2000

*A Paqui con todo mi amor,
por su apoyo y, sobre todo,
por su eterna paciencia*

Agradecimientos

A mi director de tesis, Miguel Toro, por iniciarme en el mundo de la investigación y convencerme de continuar mi trabajo a pesar de todas las dificultades.

A Beatriz, Corchu y Antonio por su ayuda en mis investigaciones y por su continuo apoyo y ánimos durante todo el largo proceso, así como por las discusiones sobre mis ideas.

A mis compañeros de asignaturas de ingeniería del software, Fran y Manolo, por aguantar mis impertinentes, continuas y muchas veces desordenadas sugerencias.

A mis amigos de la Universidad de Extremadura, Fernando, Juanma y su familia, Juan y todos los demás, por tratarme como uno más en todo momento y por su sincera amistad.

A mis amigos de la Universidad de Granada, María José y Miguel, Patricia, Pepe y todos los demás, por ser los mejores.

A mis amigos de la Universidad Politécnica de Valencia, Pele, Pepe, Hilario y el resto de los *menhires*, por su interés y por su apoyo.

A Miguel Ángel Laguna, de la Universidad de Valladolid, y a Francisco José García Peñalvo, de la Universidad de Salamanca, por aplicar mis propuestas con su alumnos cuando aún no eran más que un conjunto desordenado de ideas y proporcionarme una realimentación importantísima.

A Jon Iturrioz y a Oscar Díaz por compartir sus conocimientos conmigo y apostar por una ingeniería de requisitos práctica.

A todos los integrantes del proyecto MENHIR, que han compartido sus conocimientos en las distintas áreas de la ingeniería de requisitos, y que sin su presencia, la presente tesis probablemente habría sido muy distinta.

Resumen

En esta tesis se describe un entorno metodológico para la ingeniería de requisitos de sistemas de información compuesto por:

- un modelo de procesos iterativo en el que se identifican tres actividades principales: elicitación, análisis y validación.
- una metodología para la elicitación de requisitos de sistemas de información, incluyendo las tareas a realizar, los productos a obtener y las técnicas a emplear, principalmente plantillas y patrones de requisitos, así como la posibilidad de introducir la reutilización en el proceso.
- una metodología para el análisis de requisitos de sistemas de información, incluyendo las tareas a realizar, los productos a obtener y las técnicas a emplear, basadas en el estándar UML y con relaciones de rastreabilidad hacia los productos de la actividad anterior que facilita la reutilización de elementos complejos.
- una metodología para la validación de requisitos de sistemas de información, incluyendo las tareas a realizar, los productos a obtener y las técnicas a emplear, basadas en la combinación de los *walkthrough* o recorridos con el prototipado de sistemas.
- un prototipo de una herramienta CASE de ingeniería de requisitos como complemento a la metodología descrita.

Índice General

1	Introducción	1
1.1	Motivación	1
1.1.1	Crisis del software	1
1.1.2	La necesidad de una ingeniería del software	3
1.1.3	La necesidad de una ingeniería de requisitos	3
1.1.4	Estructura de la tesis	7
1.2	Definición de ingeniería de requisitos	9
1.2.1	Las dimensiones de la ingeniería de requisitos	10
1.3	El concepto de requisito	11
1.3.1	Las dimensiones de los requisitos	12
1.3.2	Los requisitos como restricciones	15
1.3.3	Propiedades deseables de los requisitos	16
1.4	Clasificación de la investigación en ingeniería de requisitos .	23
1.4.1	Clasificación de la tesis	24
1.5	Bibliografía	25
2	Modelos de procesos de ingeniería de requisitos	29
2.1	Propuesta de modelo de procesos de ingeniería de requisitos	29
2.1.1	Ciclos de iteración	31
2.1.2	Actividades del proceso	32
2.1.3	Productos del proceso	37
2.2	Otros modelos de procesos	38

2.2.1	El modelo de Pohl	39
2.2.2	El modelo espiral	41
2.2.3	El modelo SWEBOK	43
2.3	Comparación de modelos de procesos de ingeniería de requisitos	46
2.4	El modelo de madurez de proceso REAIMS	47
2.5	Conclusiones	50
2.6	Bibliografía	50
3	Elicitación de Requisitos	55
3.1	Introducción	55
3.2	La elicitación de requisitos	56
3.3	Problemas de la elicitación de requisitos	57
3.3.1	Problemas de articulación	58
3.3.2	Problemas de comunicación	59
3.3.3	Problemas de limitaciones cognitivas	60
3.3.4	Problemas de conducta humana	61
3.3.5	Problemas técnicos	62
3.4	Técnicas de elicitación de requisitos	63
3.4.1	Entrevistas	63
3.4.2	Joint Application Development	66
3.4.3	Brainstorming	70
3.4.4	Casos de uso	72
3.5	Propuesta metodológica para la elicitación de requisitos . . .	76
3.5.1	Tarea 1: Obtener información sobre el dominio del problema y el sistema actual	77
3.5.2	Tarea 2: Preparar y realizar las sesiones de elicitación/negociación	78
3.5.3	Tarea 3: Identificar/revisar los objetivos del sistema .	78
3.5.4	Tarea 4: Identificar/revisar los requisitos de información	79

3.5.5	Tarea 5: Identificar/revisar los requisitos funcionales	80
3.5.6	Tarea 6: Identificar/revisar los requisitos no funcionales	81
3.5.7	Tarea 7: Priorizar objetivos y requisitos	82
3.5.8	El documento de requisitos del sistema	82
3.6	Plantillas y patrones lingüísticos para elicitación de requisitos	84
3.6.1	Plantilla para los objetivos del sistema	85
3.6.2	Plantilla para requisitos de almacenamiento de información	87
3.6.3	Plantilla para actores	89
3.6.4	Plantilla para requisitos funcionales	90
3.6.5	Plantilla para requisitos no funcionales	94
3.6.6	Plantilla para conflictos	95
3.7	Patrones de reutilización de requisitos-C	96
3.7.1	Patrones- R_C de requisitos de almacenamiento de información	97
3.7.2	Patrones- R_C de requisitos funcionales	99
3.7.3	Patrones- R_C de requisitos no funcionales	104
3.8	Trabajos relacionados	104
3.8.1	Metodologías de elicitación	104
3.8.2	Plantillas para requisitos	108
3.8.3	Patrones lingüísticos	108
3.8.4	Patrones de reutilización	112
3.9	Conclusiones	116
3.10	Bibliografía	119
4	Análisis de Requisitos	127
4.1	Introducción	127
4.2	Modelado de sistemas de información	128
4.2.1	Rastreabilidad entre requisitos-C y requisitos-D . . .	130
4.3	Técnicas estructuradas de modelado	131

4.3.1	Modelado estático	132
4.3.2	Modelado funcional	133
4.3.3	Modelado dinámico	134
4.3.4	Integración de modelos	137
4.4	Técnicas orientadas a objetos de modelado	139
4.4.1	Modelado estático	140
4.4.2	Modelado funcional	141
4.4.3	Modelado dinámico	153
4.4.4	Integración de modelos	157
4.5	Comparación de técnicas de modelado	158
4.6	Propuesta metodológica para el análisis de requisitos	159
4.6.1	Tarea 1: Analizar los requisitos de almacenamiento de información	159
4.6.2	Tarea 2: Analizar los requisitos funcionales	160
4.6.3	Tarea 3: Analizar los requisitos no funcionales	161
4.6.4	Tarea 4: Desarrollar prototipos	161
4.6.5	El documento de análisis del sistema	162
4.7	Plantillas y patrones lingüísticos para análisis de requisitos	162
4.7.1	Plantillas para especificación de tipos y asociaciones	164
4.7.2	Plantilla para especificación de operaciones de sistema	170
4.8	Patrones de reutilización de requisitos–D	175
4.8.1	Patrones– R_D para requisitos de almacenamiento de información	175
4.8.2	Patrones– R_D para requisitos funcionales	176
4.9	Conclusiones	181
4.10	Bibliografía	182
5	Validación de Requisitos	189
5.1	Introducción	189
5.2	Validación y verificación de requisitos	190
5.3	Derechos y responsabilidades de clientes y usuarios	193

5.3.1	Derechos de los clientes y usuarios de software . . .	193
5.3.2	Responsabilidades de los clientes y usuarios de software	196
5.3.3	El compromiso de la validación de requisitos	199
5.4	Técnicas de validación de requisitos	200
5.4.1	Walkthroughs	200
5.4.2	Prototipos orientados a clientes/usuarios	204
5.5	Propuesta metodológica para la validación de requisitos . .	205
5.5.1	Tarea 1: Validar los requisitos de almacenamiento de información y funcionales	205
5.5.2	Tarea 2: Validar los requisitos no funcionales	206
5.5.3	Tarea 3: Cerrar la versión de los requisitos	206
5.6	Conclusiones	207
5.7	Bibliografía	207
6	Conclusiones y trabajo futuro	213
6.1	Conclusiones	213
6.1.1	La distancia entre investigación y práctica	213
6.1.2	La dificultad de la experimentación	214
6.1.3	La formación en ingeniería de requisitos	214
6.1.4	Madurez del proceso de ingeniería de requisitos . . .	215
6.2	Trabajo futuro	218
6.2.1	Completar el entorno metodológico	218
6.2.2	Reutilización de requisitos	219
6.2.3	Sacar más rendimiento a los requisitos	220
6.2.4	Requisitos no funcionales y arquitectura de software	221
6.3	Bibliografía	222
A	La IR en algunas normas de desarrollo de software	225
A.1	Ingeniería de requisitos en la norma MIL–STD–498	225
A.2	Ingeniería de requisitos en las normas IEEE	234

A.3	Ingeniería de requisitos en la norma ESA PSS-05	239
A.4	Ingeniería de requisitos en Métrica	244
A.5	Conclusiones	248
A.6	Bibliografía	249
B	Metodología para la elicitación de requisitos de SI	253
B.1	Objetivo de la metodología	253
B.2	Tareas recomendadas	254
B.2.1	Tarea 1: Obtener información sobre el dominio del problema y el sistema actual	254
B.2.2	Tarea 2: Preparar y realizar las sesiones de elicitación/negociación	256
B.2.3	Tarea 3: Identificar/revisar los objetivos del sistema .	257
B.2.4	Tarea 4: Identificar/revisar los requisitos de almacenamiento de información	258
B.2.5	Tarea 5: Identificar/revisar los requisitos funcionales	259
B.2.6	Tarea 6: Identificar/revisar los requisitos no funcionales	260
B.3	Productos entregables	261
B.3.1	Documento de requisitos del sistema	262
B.4	Técnicas	267
B.4.1	Entrevistas	267
B.4.2	Joint Application Development	267
B.4.3	Brainstorming	268
B.4.4	Casos de uso	268
B.4.5	Plantillas y patrones de objetivos y requisitos	268
B.5	Ejemplo: gestión de un vídeo-club	268
B.5.1	Objetivos del sistema	268
B.5.2	Requisitos de almacenamiento de información	269
B.5.3	Requisitos funcionales	271
B.5.4	Requisitos no funcionales	290

B.6	Bibliografía	290
C	Metodología para el análisis de requisitos de SI	293
C.1	Objetivo de la metodología	293
C.2	Tareas recomendadas	294
C.2.1	Tarea 1: Analizar los requisitos de almacenamiento de información	294
C.2.2	Tarea 2: Analizar los requisitos funcionales	295
C.2.3	Tarea 3: Analizar los requisitos no funcionales	296
C.2.4	Tarea 4: Desarrollar prototipos	297
C.3	Productos entregables	298
C.3.1	Documento de análisis del sistema	298
C.4	Técnicas	305
C.4.1	Diagrama de tipos	305
C.4.2	Plantillas para especificación de tipos y asociaciones	305
C.4.3	Diagrama de estados	305
C.4.4	Diagrama de acción conjunta	306
C.4.5	Diagrama de secuencia	306
C.4.6	Plantillas para especificación de operaciones del sistema	306
C.4.7	Construcción de prototipos	306
C.5	Ejemplo: gestión de un vídeo-club	306
C.5.1	Diagramas de tipos	307
C.5.2	Tipos y asociaciones	310
C.5.3	Operaciones del sistema	344
C.5.4	Conflictos detectados en los requisitos-C	352
C.6	Bibliografía	353
D	Metodología para la validación de requisitos de SI	355
D.1	Objetivo de la metodología	355
D.2	Tareas recomendadas	356

D.2.1	Tarea 1: Validar los requisitos de almacenamiento de información y funcionales	356
D.2.2	Tarea 2: Validar los requisitos no funcionales	357
D.2.3	Tarea 3: Cerrar la versión de los requisitos	358
D.3	Productos entregables	359
D.4	Técnicas	359
D.4.1	Walkthrough	360
D.5	Bibliografía	360
E	Una herramienta CASE para la ingeniería de requisitos	361
E.1	Introducción	361
E.2	Modelo de los requisitos	362
E.2.1	Modelo de un proyecto de ingeniería de requisitos	362
E.2.2	Modelo de los requisitos–C	364
E.2.3	Modelo de los requisitos–D	366
E.2.4	Modelo de los conflictos	368
E.3	Arquitectura del prototipo	368
E.4	Interfaz de usuario del prototipo	370
E.4.1	Vista de proyecto	370
E.4.2	Vista de requisitos–C	371
E.4.3	Vista de requisitos–D	372
E.4.4	Vista de conflictos	373
E.4.5	Otros elementos de la interfaz de usuario	374
E.5	Otras herramientas CASE de requisitos	375
E.6	Bibliografía	378

Índice de Figuras

1.1	Resultados del informe del GAO	4
1.2	Resultados del informe CHAOS	4
1.3	La ingeniería de requisitos en el ciclo de vida del desarrollo de software	10
1.4	Dimensiones de la ingeniería de requisitos	11
1.5	Dimensiones de los requisitos	13
1.6	Los requisitos como restricciones	15
1.7	La ingeniería de requisitos como un proceso de comunicación	16
1.8	Requisitos innecesarios ($S_v' \subset S_v$)	18
1.9	Requisitos incompletos ($S_v \subset S_v'$)	19
1.10	Requisitos inconsistentes ($S_v = \emptyset$)	20
2.1	Modelo de procesos de ingeniería de requisitos: iteración de actividades	30
2.2	Modelo de procesos de ingeniería de requisitos: entradas– actividades–salidas	33
2.3	Modelo de procesos de ingeniería de requisitos de Pohl . . .	39
2.4	Modelo espiral de procesos de ingeniería de requisitos . . .	42
2.5	Modelo SWEBOK de procesos de ingeniería de requisitos . .	43
2.6	Niveles de madurez de proceso de ingeniería de requisitos .	48
3.1	Diagrama de casos de uso	73
3.2	Representación gráfica de las relaciones <i>includes</i> y <i>extends</i> . .	74
3.3	Representación gráfica de los paquetes de casos de uso . . .	75

3.4	Tareas de elicitación de requisitos	76
3.5	Rastreabilidad entre participantes, objetivos y requisitos–C .	79
3.6	Representación del ámbito en los diagramas de casos de uso	81
3.7	Estructura del Documento de Requisitos del Sistema	83
3.8	Matriz de rastreabilidad del DRS	83
3.9	La plantilla como elemento de elicitación y negociación . . .	85
3.10	Plantilla y patrones–L para objetivos	86
3.11	Plantilla y patrones–L para requisitos de almacenamiento de información	88
3.12	Plantilla y patrones–L para actores	90
3.13	Plantilla y patrones–L para requisitos funcionales (casos de uso)	91
3.14	Ejemplo de caso de uso de conexión de usuario (plantilla) . .	93
3.15	Ejemplo de caso de uso de conexión de usuario (Coleman) .	93
3.16	Plantilla y patrones–L para requisitos no funcionales	95
3.17	Plantilla para conflictos	95
3.18	Frecuencia de aparición de algunos patrones– R_C de requi- sitos de almacenamiento de información	97
3.19	Patrón– R_C <i>Cliente/Socio/Paciente</i> de requisito de almacena- miento de información	98
3.20	Patrón– R_C <i>Producto</i> de requisito de almacenamiento de in- formación	99
3.21	Patrón– R_C <i>Empleado</i> de requisito de almacenamiento de in- formación	99
3.22	Patrón– R_C <i>Crear/Dar de alta/Registrar</i> de requisito funcional .	100
3.23	Patrón– R_C <i>Eliminar/Dar de baja/Cancelar</i> de requisito funcional	101
3.24	Patrón– R_C <i>Modificar/Editar</i> de requisito funcional	102
3.25	Patrón– R_C <i>Consultar</i> de requisito funcional	103
3.26	Patrón– R_C <i>Sistema operativo</i> de requisito no funcional	104
3.27	Modelo de procesos de elicitación de requisitos de Volere . .	106
3.28	Plantilla para requisitos de Volere	109
3.29	Comparativas de plantillas para casos de uso	110

3.30	Guías de estilo CREWS	111
3.31	Guías de contenido CREWS	112
3.32	Aplicación del patrón <i>Especificar</i> en los casos de uso	114
3.33	La rastreabilidad como clave para la reutilización vertical	118
4.1	Dimensiones del modelado de sistemas de información	129
4.2	Modelos de caja negra (requisitos–C) y de caja gris (requisitos–D)	129
4.3	Técnicas de modelado estructuradas	132
4.4	Ejemplo de diagrama entidad–interrelación	132
4.5	Ejemplo de diagrama de flujo de datos	134
4.6	Ejemplo de diagrama de transición de estados	135
4.7	Ejemplo de diagrama de historia de vida de entidad	135
4.8	Técnicas de modelado orientadas a objetos	140
4.9	Ejemplo de diagrama de tipos	141
4.10	Ejemplo de diagrama de secuencia	142
4.11	Ejemplo de diagrama de colaboración	142
4.12	Ejemplo de statechart de Syntropy	144
4.13	Esquema de <i>operación de sistema</i> de Fusion	149
4.14	Representación gráfica de una acción conjunta en Catalysis	150
4.15	Representación de las acciones como interfaz del sistema	152
4.16	Ejemplo de Statechart (Catalysis)	156
4.17	Comparación de técnicas estructuradas y orientadas a objetos	158
4.18	Tareas de análisis de requisitos	159
4.19	Estructura del Documento de Análisis del Sistema	163
4.20	Plantilla de descripción de tipo de objetos	165
4.21	Plantillas de descripción de atributos	166
4.22	Plantillas de descripción de enlaces	167
4.23	Plantilla de descripción de invariante de tipo	168
4.24	Plantillas de descripción de asociación	169
4.25	Plantilla de descripción de rol	170

4.26	Esquema para la especificación de operaciones de sistema	171
4.27	Esquema para la especificación de operaciones de consulta al sistema	172
4.28	Plantilla de descripción de operación de sistema	173
4.29	Patrón- R_D <i>Cliente/Socio</i>	175
4.30	Patrón- R_D para la operación de sistema <i>Crear</i> <X>	176
4.31	Patrón- R_D para la operación de sistema <i>Eliminar</i> <X>	177
4.32	Patrón- R_D para la operación de sistema <i>Modificar</i> <X>	178
4.33	Patrón- R_D para la operación de sistema <i>Consultar</i> <X>	179
4.34	Diagrama de tipos para el ejemplo de consulta en OCL	180
5.1	La validación como acuerdo entre clientes y desarrolladores	199
5.2	Tareas de validación de requisitos	205
6.1	Guías REAIMS (nivel Inicial)	216
6.2	Guías REAIMS (niveles Repetible y Avanzado)	217
A.1	MIL-STD-498: sistema hardware-software	226
A.2	Estructura del OCD de la norma MIL-STD-498	228
A.3	Estructura del SSS de la norma MIL-STD-498	230
A.4	Estructura del SSDD de la norma MIL-STD-498	231
A.5	Estructura del SRS de la norma MIL-STD-498	232
A.6	MIL-STD-498: sistema sólo software con varios CSCIs	233
A.7	MIL-STD-498: sistema sólo software con un sólo CSCI	233
A.8	Procesos del ciclo de vida del software según la norma ISO/IEC/IEEE 12207	235
A.9	IEEE 12207: sistema con varios elementos software	236
A.10	Estructura de la SyRS de la norma IEEE-1233	237
A.11	Estructura de la SRS de la norma IEEE-830	238
A.12	Fases de ingeniería de requisitos en la norma ESA PSS-05	240
A.13	Estructura del URD de la norma ESA PSS-05	242
A.14	Estructura del SRD de la norma ESA PSS-05	243

A.15 Métrica 2.1: ingeniería de requisitos	244
A.16 Estructura del DRS de Métrica V2.1	245
A.17 Estructura del EFS de Métrica V2.1	246
A.18 Métrica 3: ingeniería de requisitos	248
B.1 Estructura del Documento de Requisitos del Sistema	262
B.2 Portada del Documento de Requisitos del Sistema	263
B.3 Lista de cambios del Documento de Requisitos del Sistema	264
B.4 Matriz de rastreabilidad del Documento de Requisitos del Sistema	266
B.5 Diagrama de subsistemas	271
B.6 Diagrama de casos de uso del subsistema Gestión de socios	271
B.7 Diagrama de casos de uso del subsistema Gestión de películas	272
B.8 Diagrama de casos de uso del subsistema Gestión de alquileres	273
C.1 Estructura del Documento de Análisis del Sistema	299
C.2 Portada del Documento de Análisis del Sistema	301
C.3 Lista de cambios del Documento de Análisis del Sistema	301
C.4 Matriz de rastreabilidad del Documento de Análisis del Sistema	304
C.5 Diagrama de <i>subsistemas</i>	307
C.6 Diagrama de tipos del subsistema <i>Películas y Cintas</i>	308
C.7 Diagrama de tipos del subsistema <i>Alquileres y Socios</i>	309
C.8 Diagrama de acción conjunta de <i>AltaDeSocio</i>	345
C.9 Diagrama de secuencia de <i>AltaDeSocio</i>	345
C.10 Interfaz de usuario de <i>AltaDeSocio</i>	346
C.11 Diagrama de acción conjunta de <i>AlquilarCinta</i>	348
C.12 Diagrama de secuencia de <i>AlquilarCinta</i>	348
C.13 Interfaz de usuario de <i>AlquilarCinta</i>	349
C.14 Interfaz de usuario del operación <i>ConsultarPelícula</i>	351

E.1	Subsistemas del modelo de requisitos	362
E.2	Estructura de los proyectos de ingeniería de requisitos . . .	363
E.3	Jerarquía de objetos de especificación	363
E.4	Estructura las especificaciones de requisitos–C	364
E.5	Estructura de los casos de uso	365
E.6	Jerarquía de acciones	365
E.7	Estructura de las especificaciones de requisitos–D	366
E.8	Transformación de asociaciones en asociaciones como tipo .	367
E.9	Clasificación de propiedades de tipos de objetos	367
E.10	Estructura de los conflictos	368
E.11	Arquitectura de la herramienta CASE	369
E.12	Vista de proyecto de la herramienta CASE	370
E.13	Vista de requisitos–C de la herramienta CASE	371
E.14	Vista de requisitos–D de la herramienta CASE	372
E.15	Vista de conflictos de la herramienta CASE	373
E.16	Ejemplo de diálogo compuesto	374
E.17	Matriz de rastreabilidad de Requisite Pro	376
E.18	Acceso a través de Internet de Requisite Pro	377
E.19	Foros de discusión de Requisite Pro	378

Capítulo 1

Introducción

1.1 Motivación

1.1.1 Crisis del software

Imaginemos por un momento una sala de un centro de cálculo a principios de los años 50 en algún organismo del gobierno de los Estados Unidos. La estancia está ocupada por un enorme ordenador cuyo desarrollo ha costado varios millones de dólares y cuyo mantenimiento tiene ocupadas a varias personas las 24 horas del día.

La siguiente media hora es el turno semanal de uno de los muchos científicos que trabajan para el Departamento de Defensa. Su trabajo consiste en calcular tablas numéricas para trayectorias balísticas usando ecuaciones relativamente complejas. Lleva toda la semana repasando su programa en su despacho para asegurarse que no contiene errores sintácticos ni semánticos, ya que un fallo en la compilación podría hacerle perder su preciosa media hora semanal.

Cuando comienza su tiempo, el ordenador comienza a leer las tarjetas perforadas que previamente había entregado al operador, compila el programa sin errores y comienza su ejecución. El científico se dispone a esperar a que la impresora comience a imprimir las tablas. Cuando faltan sólo pocas líneas para completarlas, el hardware falla y los encargados de mantenimiento deben parar la máquina para repararla. Quizás la semana que viene tenga más suerte.

Si comparamos esta situación con la actual, pocos son los aspectos comunes que podríamos encontrar. Sin embargo, podemos hacer un ejerci-

cio de análisis e identificar algunas de las características del desarrollo de software en los comienzos de la informática.

- *El hardware es mucho más caro que el software.* La máquina y su mantenimiento cuestan millones de dólares. Comparado con ese coste, el sueldo del científico que escribe el programa es ridículo, así que ¿por qué preocuparse por el coste del desarrollo de software?
- *El desarrollo del hardware es más complejo que el del software.* La tecnología hace que el hardware sea complejo de construir y mantener. El software habitual suelen ser programas no muy grandes (debido, entre otras limitaciones, a la escasa capacidad de memoria) y suelen estar escritos por una única persona, normalmente empleada en la organización que utiliza el hardware. Los requisitos¹ que tiene que cumplir el software son simples. Por lo tanto, ¿por qué preocuparse por la complejidad del software?
- *El hardware es poco fiable.* Debido a la tecnología que se utiliza para su implementación, en cualquier momento la máquina puede sufrir una avería, así que ¿para qué preocuparse por la calidad del software?

Esta despreocupada situación respecto al software cambia cuando, gracias a los avances en la tecnología, aumenta la capacidad de memoria y se reducen los costes de desarrollo y mantenimiento del hardware. Se empiezan a comercializar los primeros ordenadores y la demanda de software más complejo crece rápidamente, destapando la caja de Pandora de la *crisis del software*, término utilizado por primera vez en la conferencia organizada por la Comisión de Ciencia de la OTAN en Garmisch, Alemania, en octubre de 1968, para designar la gran cantidad de problemas que presentaba (y aún presenta) el desarrollo de software y el alto índice de fracasos en los proyectos de desarrollo.

¿Qué podía hacerse ante una situación en la que los proyectos software tenían un alto riesgo de fracasar? La respuesta parecía obvia: construir software de forma similar a como se construye hardware, aviones, barcos, puentes o edificios, es decir, aplicar los métodos de la ingeniería a la construcción de software.

¹En esta memoria se utilizará el término *requisito*, y no *requerimiento* para referirse al término inglés *requirement*, tal como se propone en [Fernández y Rueda 1998]. La definición del concepto se realizará en la sección 1.3, pág. 11.

1.1.2 La necesidad de una ingeniería del software

Esta nueva ingeniería, la *ingeniería del software*², se ha definido de muchas formas, dos de las cuales, [IEEE 1990] y [Davis 1995, pág. XI], son las siguientes:

ingeniería del software (1): *(a) la aplicación de un enfoque sistemático, disciplinado y cuantificable para el desarrollo, operación y mantenimiento del software; es decir, la aplicación de la ingeniería al software. (b) el estudio de los enfoques como los descritos en (a).*

ingeniería del software (2): *la aplicación inteligente de principios probados, técnicas, lenguajes y herramientas para la creación y mantenimiento, dentro de un coste razonable, de software que satisfaga las necesidades de los usuarios.*

Las razones que justifican la necesidad de una ingeniería del software parecen claras. Sin embargo, la ingeniería del software comprende una serie de actividades lo suficientemente diversas como para poder considerar la necesidad de otras ingenierías dentro de la propia ingeniería del software.

1.1.3 La necesidad de una ingeniería de requisitos

Desde 1968 se ha invertido un gran esfuerzo en determinar las causas y proponer soluciones para la crisis del software.

En 1979, la Oficina de Cuentas del gobierno norteamericano (*Government Account Office*, GAO) realizó un estudio [GAO 1979] seleccionando 9 proyectos de desarrollo de software para el gobierno norteamericano cuyos contratos sumaban una cantidad total de 6.800.000 dólares.

De esta cantidad, sólo 119.000 dólares correspondían a un proyecto que se había utilizado tal como se había entregado. Dicho proyecto se trataba de un preprocesador de COBOL, por lo que era un problema relativamente simple cuyos requisitos eran comprendidos por clientes y desarrolladores y que además no cambiaron durante el desarrollo.

El resto de los 6.8 millones de dólares se distribuyó como puede verse en la figura 1.1, en la que puede destacarse el enorme porcentaje de dinero invertido en proyectos cancelados o no satisfactorios.

²Término utilizado por primera vez en la conferencia de la OTAN anteriormente citada.

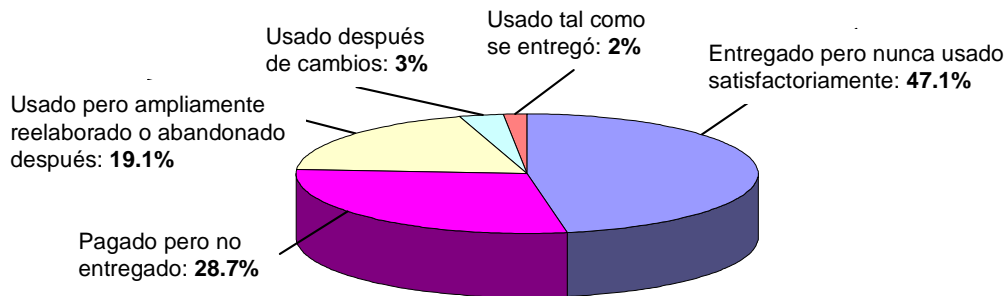


Figura 1.1: Resultados del informe del GAO

En 1995, el Grupo Standish realizó un estudio (el informe CHAOS) mucho más amplio y significativo que el del GAO cuyos resultados, a pesar de haber pasado más de 25 años, no reflejaban una mejoría sustancial [TSG 1995].

Los resultados generales, que pueden verse en la figura 1.2, al compararse con los de [GAO 1979] presentan una mejora en los proyectos que se entregan cumpliendo todos sus requisitos, 2% frente al 16.2% (sólo el 9% en grandes compañías), pero empeoran ligeramente respecto a los que se abandonan durante el desarrollo, 28.7% frente a 31.1%.

Sin incluir al 16.2% de los proyectos terminados correctamente, la media del gasto final fue del 189% del presupuesto original, el tiempo necesario para su realización del 222% del plazo original y se cumplieron una media del 61% de los requisitos iniciales, cifras que también empeoraban en el caso de grandes compañías.

Las encuestas realizadas a los directores de los proyectos que participaron en el estudio indicaron que, en su opinión, los tres principales factores de éxito eran:

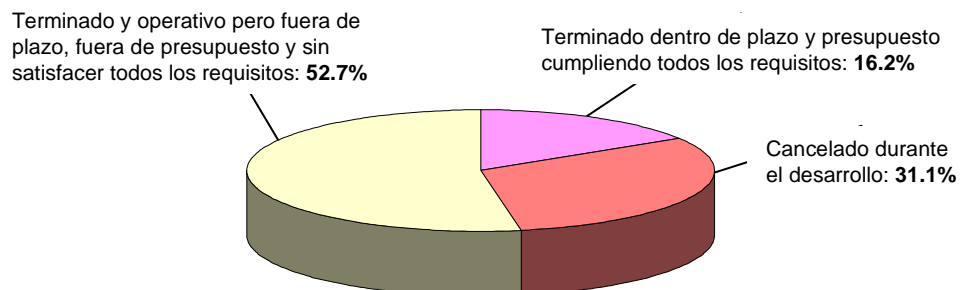


Figura 1.2: Resultados del informe CHAOS

- 1 Implicación de los usuarios
- 2 Apoyo de los directivos
- 3 Enunciado claro de los requisitos

mientras que los tres principales factores de fracaso eran:

- 1 Falta de información por parte de los usuarios
- 2 Especificaciones y requisitos incompletos
- 3 Especificaciones y requisitos cambiantes

En 1996, el proyecto ESPITI (*European Software Process Improvement Training Initiative*) [ESP 1996] realizó una investigación sobre los principales problemas en el desarrollo de software a nivel europeo. Los resultados, muy similares a los obtenidos en el informe CHAOS, indicaron que los mayores problemas estaban también relacionados con la especificación, la gestión y la documentación de los requisitos.

Estos informes ponen de manifiesto el hecho de que, a pesar de que las herramientas para construir software han evolucionado enormemente, se sigue produciendo software que no es satisfactorio para los clientes y usuarios. Esto indica que los principales problemas que han dado origen a la crisis del software residen en las primeras etapas del desarrollo, cuando hay que decidir las características del producto software a desarrollar.

En palabras de F. P. Brooks [Brooks 1995, pág. 199]:

*"La parte más difícil de construir de un sistema software es decidir qué construir. [...] Ninguna otra parte del trabajo afecta más negativamente al sistema final si se realiza de manera incorrecta. Ninguna otra parte es más difícil de rectificar después."*³

Otro hecho comprobado es que el coste de un cambio en los requisitos, una vez entregado el producto, es entre 60 y 100 veces superior al coste que hubiera representado el mismo cambio durante las fases iniciales de

³Salvando las distancias, podría decirse lo mismo de la elaboración de una tesis doctoral: *"La parte más difícil en la elaboración de una tesis doctoral es decidir qué investigar. Ninguna otra parte del trabajo afecta más negativamente al resultado final si se realiza de manera incorrecta. Ninguna otra parte es más difícil de rectificar después."*

desarrollo [Pressman 1997, Davis 1993], por lo que no es de extrañar que aquellos proyectos en los que no se determinan correctamente los requisitos y cambian frecuentemente durante el desarrollo, superen con creces su presupuesto inicial.

Todas estas circunstancias han convencido a la gran parte de la comunidad de la ingeniería del software de la necesidad, cada vez mayor, de una *ingeniería de requisitos*.

Ejemplos de este convencimiento son, entre otros:

- el hecho de que el *Software Engineering Institute* de la Universidad Carnegie–Mellon haya identificado la *gestión de requisitos* como una de las áreas de proceso clave (*key process area*) dentro del nivel 2 (*Repeatible*) del *Capability Maturity Model* (CMM) [Paulk et al. 1993].
- la organización de congresos específicos como:
 - El *IEEE International Symposium on Requirements Engineering* (RE), que se celebra los años impares desde 1993 organizado por IEEE, ACM, IFIP y varias asociaciones más.
 - La *IEEE International Conference on Requirements Engineering* (ICRE), que se celebra los años pares desde 1994 organizado por el IEEE.
 - El *Workshop em Engenharia de Requisitos* (WER), que se celebra anualmente desde 1998.
- la publicación de revistas especializadas como el *Requirements Engineering Journal*, que se publica trimestralmente desde 1996.
- la aparición bianual de números monográficos sobre ingeniería de requisitos en *IEEE Software* coincidiendo con la celebración del ICRE, en concreto los correspondientes a los meses de marzo de los años 1994, 1996 y 1998 y el del mes de mayo del año 2000.
- la financiación pública de proyectos europeos como:
 - NATURE (*Novel Approaches to Theories Underlying Requirements Engineering*).
 - REAIMS (*Requirements Engineering Adaptation and IMprovement for Safety and dependability*).
 - CREWS (*Cooperative Requirements Engineering With Scenarios*)

- la financiación de proyectos nacionales como el proyecto CICYT MENHIR (*Metodologías, Entornos y Nuevas Herramientas para la Ingeniería de Requisitos*)
- la red europea RENOIR (*Requirements Engineering Network Of International cooperating Research groups*)

Una vez asumida la necesidad de una ingeniería de requisitos, el reto al que se enfrenta la comunidad investigadora es identificar claramente los problemas a los que se enfrenta y dotar a dicha ingeniería de las herramientas teóricas y prácticas necesarias para solucionarlos, teniendo en cuenta la dificultad añadida de lo costoso de la obtención de datos empíricos tanto en ingeniería del software en general, como en ingeniería de requisitos en particular.

La motivación del trabajo que se describe en esta memoria es realizar aportaciones a la ingeniería de requisitos, especialmente en los aspectos metodológicos y de herramientas que soporten el proceso. Dentro de las posibles clases de sistemas informáticos⁴ (*sistemas de información, sistemas empotrados y sistemas de control* [Sawyer y Kontoya 1999]), este trabajo se centra en la ingeniería de requisitos de sistemas de información.

1.1.4 Estructura de la tesis

La estructura de esta tesis está organizada alrededor del modelo de procesos de ingeniería de requisitos que se propone en el capítulo 2, en el que se definen las actividades de *elicitación de requisitos*⁵, *análisis de requisitos* y *validación de requisitos*.

En este primer capítulo se ofrece una introducción general a la ingeniería de requisitos, sus conceptos básicos y se clasifica la investigación realizada en esta tesis según los criterios expuestos en [Zave 1997].

En el capítulo 2 se propone un modelo de procesos para la ingeniería de requisitos que servirá de base para el resto del trabajo. En dicho capítulo también se comparan otros modelos de procesos propuestos y se comenta el modelo de madurez de procesos de ingeniería de requisitos resultado del proyecto REAIMS [Sawyer *et al.* 1997, Sommerville y Sawyer 1997].

⁴Computer-based systems en inglés.

⁵Aunque no es una palabra castellana, en este trabajo se utilizará el término *elicitación* para referirse al término inglés *elicitation*, tal como se justifica en [Fernández y Rueda 1998].

En el capítulo 3 se estudian los problemas relacionados con las actividades de elicitación de requisitos, las técnicas más habituales y se describen la propuesta metodológica para la elicitación de requisitos, las plantillas y patrones de elicitación de requisitos y los patrones de reutilización de requisitos.

En el capítulo 4 se estudian las alternativas estructuradas y orientadas a objetos para el análisis de requisitos y se describe la metodología propuesta para el análisis de requisitos junto con las plantillas y patrones de reutilización para esta actividad de la ingeniería de requisitos.

En el capítulo 5 se describe la propuesta metodológica para la validación de requisitos y se estudian algunas técnicas para la validación de requisitos de sistemas de información.

Por último, en el capítulo 6 se exponen las conclusiones y se describen posibles líneas de trabajo futuro.

A estos cuatro capítulos les acompañan los siguientes apéndices:

A : *La ingeniería de requisitos en algunas normas de desarrollo de software*

En este apéndice se incluye un estudio comparativo de las actividades relacionadas con la ingeniería de requisitos en distintas normas oficiales de desarrollo de software.

B : *Metodología para la Elicitación de Requisitos de Sistemas de Información*

En este apéndice se incluye la metodología desarrollada para la actividad de elicitación de requisitos. La versión que se incluye en este trabajo es la evolución de la presentada en [Durán *et al.* 1998].

C : *Metodología para el Análisis de Requisitos de Sistemas de Información*

En este apéndice se incluye la metodología desarrollada para la actividad de análisis de requisitos, en la que se propone el uso de técnicas orientadas a objetos basadas en UML [Booch *et al.* 1999] siguiendo las propuestas descritas en [D'Souza y Wills 1999].

D : *Metodología para la Validación de Requisitos de Sistemas de Información*

En este apéndice se incluye la metodología desarrollada para la actividad de validación de requisitos, en la que se propone el prototipado como principal técnica de validación.

E : *Una Herramienta CASE para la Ingeniería de Requisitos*

En este apéndice se describe el prototipo de herramienta CASE de ingeniería de requisitos que se ha realizado para dar soporte a la metodología presentada en esta tesis.

1.2 Definición de ingeniería de requisitos

La ingeniería de requisitos es todavía una disciplina inmadura. De hecho, no hay una definición universalmente aceptada. Por ejemplo, en [Christel y Kang 1992] se pueden encontrar las siguientes:

ingeniería de requisitos (1): *Aplicación disciplinada de principios científicos y técnicas para desarrollar, comunicar y gestionar requisitos.*

ingeniería de requisitos (2): *El proceso sistemático de desarrollar requisitos mediante un proceso iterativo y cooperativo de analizar el problema, documentar las observaciones resultantes en varios formatos de representación y comprobar la precisión del conocimiento obtenido.*

En [Hsia et al. 1993] aparece la siguiente:

ingeniería de requisitos (3): *Todas las actividades relacionadas con: (a) identificación y documentación de las necesidades de clientes y usuarios. (b) creación de un documento que describe la conducta externa y las restricciones asociadas [de un sistema] que satisfará dichas necesidades. (c) análisis y validación del documento de requisitos para asegurar consistencia, compleción y viabilidad. (d) evolución de las necesidades.*

En [IEEE 1990] aparece la siguiente definición de *análisis de requisitos*⁶ que, tal como se propone en [Pohl 1997], puede interpretarse como otra definición de ingeniería de requisitos:

ingeniería de requisitos (4): *(a) el proceso de estudiar las necesidades del usuario para llegar a una definición de requisitos de sistema, hardware o software. (b) el proceso de estudiar y refinar los requisitos de sistema, hardware o software.*

En nuestra opinión, la ingeniería de requisitos puede considerarse como un proceso de *descubrimiento y comunicación* de las necesidades de clientes y usuarios y la gestión de los cambios en dichas necesidades.

Su situación dentro del ciclo de vida de desarrollo de software puede verse en la figura 1.3, aunque debe tenerse en cuenta que la ingeniería de requisitos continúa durante todo el proceso de desarrollo [Sawyer y Kontoya 1999].

⁶En las secciones 3.2 (pág. 56) y 4.1 (pág. 127) se discutirá la diferencia entre ingeniería de requisitos y análisis de requisitos y el porqué son consideradas a veces como sinónimos.

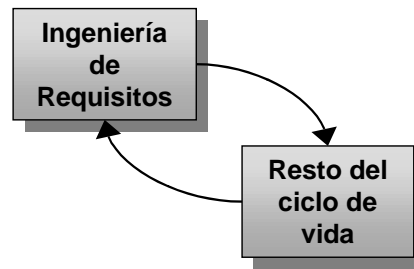


Figura 1.3: La ingeniería de requisitos en el ciclo de vida del desarrollo de software

Desde nuestro punto de vista, el aspecto más importante en la ingeniería de requisitos es la *comunicación*, como se discutirá más adelante. Esta característica es la que hace de la ingeniería de requisitos una disciplina especialmente compleja ya que hay un factor que, aunque lleva siendo estudiado mucho tiempo, apenas se conoce aún: el *factor humano*.

Este factor es el responsable de que la ingeniería de requisitos tenga aspectos sociales y culturales y no sólo técnicos [Goguen 1994].

1.2.1 Las dimensiones de la ingeniería de requisitos

Una posible visión de la ingeniería de requisitos es considerarla como un proceso de construcción de una especificación de requisitos en el que se avanza desde especificaciones iniciales, que no poseen las propiedades oportunas, hasta especificaciones finales completas, formales y acordadas entre todos los participantes [Pohl 1994, Pohl 1997].

Por un lado están los factores psicológicos y cognitivos que afectan al grado de *compleción* del conocimiento sobre el sistema que se desea desarrollar, es decir, el llegar a conocer la totalidad de los requisitos que debe satisfacer el sistema.

Por otro lado, está el grado de *formalismo* de la representación del conocimiento sobre dichos requisitos, teniendo en cuenta que un mayor grado de formalismo no implica necesariamente un mayor conocimiento [Pohl 1997, pág. 24].

Por último, como ya se comentó anteriormente, están los aspectos sociales, ya que al ser un proceso en el que participan personas con diferentes puntos de vista, es necesario llegar a un punto de *acuerdo*, normalmente mediante algún tipo de *negociación* [Boehm *et al.* 1994].

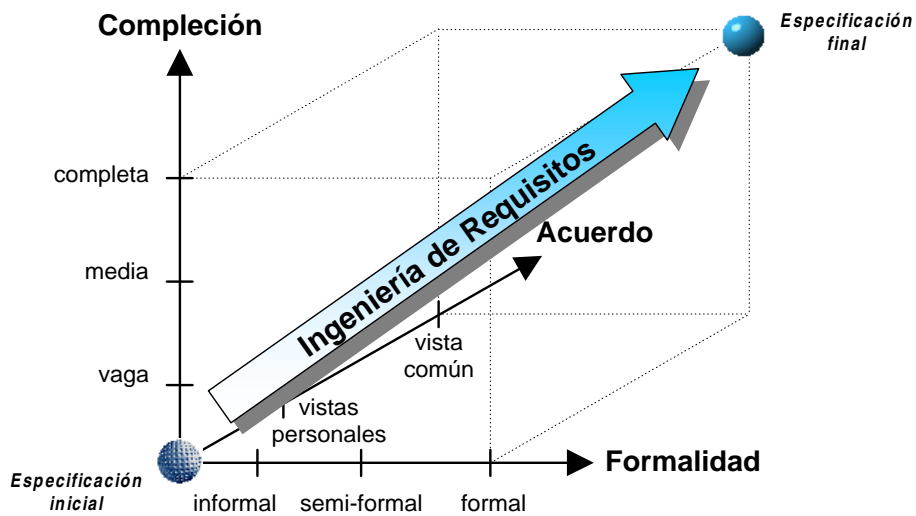


Figura 1.4: Dimensiones de la ingeniería de requisitos

Estos factores pueden representarse como tres dimensiones [Pohl 1994, Pohl 1997]⁷, de forma que durante el proceso de ingeniería de requisitos se avanza desde especificaciones incompletas, informales e individuales hacia la especificación ideal que sería completa, formal y acordada entre todos los participantes (ver figura 1.4).

Como complemento a este modelo, podríamos decir que las actividades del proceso de ingeniería de requisitos descrito en el capítulo 2 serían los *vectores* que hacen avanzar las especificaciones hacia su situación ideal.

Así, las actividades de elicitación harían avanzar el proceso en las dimensiones de completión y acuerdo, las actividades de análisis lo harían en completión y formalidad y las actividades de validación en acuerdo y formalidad.

1.3 El concepto de requisito

Una de las características de la ingeniería de requisitos es la falta de uniformidad en la terminología empleada, tanto para los conceptos básicos como para los procesos y los productos [Davis 1993, Brackett 1990, Pohl

⁷Los nombres originales de las dimensiones descritas en [Pohl 1994, Pohl 1997] son *specification*, *representation* y *agreement*, que se han traducido libremente como *compleción*, *formalidad* y *acuerdo* respectivamente.

1997]. Uno de los conceptos afectados por dicha falta de uniformidad es el de *requisito*.

La definición que aparece en [IEEE 1990] es la siguiente:

requisito (1): *(a) una condición o capacidad que un usuario necesita para resolver un problema o lograr un objetivo. (b) una condición o capacidad que debe tener un sistema o un componente de un sistema para satisfacer un contrato, una norma, una especificación u otro documento formal. (c) una representación en forma de documento de una condición o capacidad como las expresadas en (a) o en (b).*

Mientras que la que aparece en [DoD 1994] es más concisa:

requisito (2): *característica del sistema que es una condición para su aceptación.*

Otra posible definición es la siguiente [Goguen 1994]:

requisito (3): *propiedad que un sistema debería tener para tener éxito en el entorno en el que se usará.*

Sin embargo, a pesar de esta aparente simplicidad del concepto, es frecuente encontrar el término requisito calificado con adjetivos que pueden resultar confusos en un primer momento: *de sistema, hardware, software, de usuario, de cliente, funcional, no funcional*, etc.

1.3.1 Las dimensiones de los requisitos

En nuestra opinión, la gran cantidad de calificativos que se aplican al término *requisito* muestran distintos aspectos ortogonales que ha menudo se consideran aisladamente.

Para intentar clarificar la situación, hemos identificado tres dimensiones en las que se pueden clasificar los requisitos (ver figura 1.5). Estas tres dimensiones son:

- **Ámbito:** esta dimensión indica en qué ámbito se debe entender el requisito. En general, y siguiendo entre otras las propuestas de [IEEE 1997], [DoD 1994] y [Davis 1993], un ámbito de *sistema* indica que el

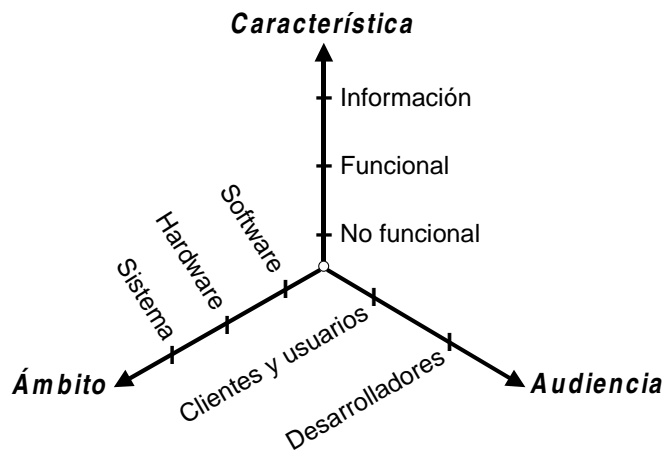


Figura 1.5: Dimensiones de los requisitos

requisito debe cumplirse a nivel de sistema, entendiendo por sistema un conjunto de hardware y software. Si el ámbito es de *software* quiere decir que el requisito sólo afecta a la parte software de un sistema, mientras que si es el ámbito es de *hardware* sólo afecta a la parte hardware.

Para entender esta clasificación conviene recordar que [DoD 1994] es una norma militar y que las normas [IEEE 1997] están fuertemente influidas por dichas normas militares. En el contexto de los desarrollos para fines militares es frecuente tener que desarrollar sistemas en los que el hardware juega un papel tan importante como el software.

En el caso de sistemas sin una parte hardware significativa, es decir sin hardware *a medida*, los requisitos de sistema y los requisitos software coinciden.

La concepción de sistema como conjunto hardware–software no es la única. Por ejemplo, en Métrica V2.1 [MAP 1995] se denominan requisitos del sistema a los requisitos que ha de cumplir el sistema a desarrollar, entendiendo por sistema el conjunto de procesos tanto automáticos como manuales. En esta situación se pueden encontrar matices que indiquen si un requisito se refiere al sistema en su conjunto o sólo al software, aunque en general dichas diferencias se obvian y no se diferencia entre los distintos ámbitos.

Como se comentó en la sección 1.1.3, este trabajo se ocupa de la ingeniería de requisitos para sistemas de información, por lo que nos centraremos principalmente en los requisitos software y cuando se

utilice el término *sistema* se entenderá como *sistema software* a menos que se indique lo contrario.

- **Característica que define:** esta dimensión clasifica los requisitos en función de la naturaleza de la característica del sistema deseada que se especifica. La clasificación más habitual suele ser la de requisitos *funcionales* (qué funciones debe realizar el sistema) y *no funcionales* (otras características del sistema).

En [Pohl 1997] aparece una completa clasificación denominada RSM (*Requirements Specification Model*, Modelo de Especificación de Requisitos), cuyas principales clases son: requisitos funcionales, requisitos de datos y requisitos no funcionales.

En nuestra opinión, y siguiendo la clasificación RSM, es conveniente separar de los requisitos funcionales a los requisitos de datos o de almacenamiento de información, que establecen qué información debe almacenar el sistema por ser relevante para las necesidades y objetivos de clientes y usuarios. En el capítulo 3 realizaremos una justificación más amplia de esta decisión.

Es conveniente destacar que al grupo de requisitos no funcionales no se le ha prestado la atención suficiente y que ya hay opiniones que lo consideran como un *cajón de sastre* excesivamente heterogéneo donde se han clasificado aquellos requisitos que resultan *incómodos* [Bass *et al.* 1998]. Un ejemplo de esta situación es la escasa importancia que se les ha dado en las técnicas de modelado de sistemas, tanto estructuradas como orientadas a objetos o formales.

- **Audiencia:** esta dimensión, fundamental desde nuestro punto de vista, indica la audiencia a la que está dirigido el requisito, es decir, las personas que deben ser capaces de entenderlo. En general, se pueden distinguir dos tipos de audiencia, los *clientes y usuarios*, que no tienen por qué tener formación en ingeniería del software, y los *desarrolladores* de software.

Cuando la audiencia está formada por clientes y usuarios, la forma más habitual de definir los requisitos es mediante lenguaje natural. En el caso de que la audiencia prevista esté formada por desarrolladores de software, los requisitos suelen expresarse mediante un modelo⁸, normalmente utilizando técnicas estructuradas, orientadas a objetos o formales.

⁸Principalmente los requisitos de información y los funcionales. Como ya se comentó, los no funcionales no suelen contemplarse en las técnicas de modelado.

En este trabajo seguiremos la nomenclatura propuesta en [Rombach 1990] y en [Brackett 1990] en la que se denominan requisitos orientados al cliente, abreviadamente *requisitos-C*, a los requisitos desde el punto de vista de los clientes y usuarios, y requisitos orientados al desarrollador, abreviadamente *requisitos-D*, a los requisitos desde el punto de vista de los desarrolladores.

No obstante, es relativamente frecuente encontrar el término *requisito de usuario* o *requisito de cliente* para designar requisitos-C de sistema o de software, y el término *requisito software* para designar requisitos-D de software. Un ejemplo de este uso puede verse en las normas de desarrollo de software de la Agencia Espacial Europea [Mazza *et al.* 1994].

1.3.2 Los requisitos como restricciones

Los requisitos también pueden interpretarse como restricciones que determinan el espacio de sistemas válidos [Gause y Weinberg 1989, Davis 1990] (ver figura 1.6).

Si S_p es el espacio de todos los posibles sistemas, cada requisito R_i establece una partición formada por S_i y \bar{S}_i , donde S_i representa al conjunto de sistemas que satisfacen el requisito R_i y \bar{S}_i representa al conjunto de sistemas que no lo satisfacen.

En esta situación, S_v , el conjunto de los sistemas válidos, es decir aquellos que satisfacen todos los requisitos, se correspondería con la intersección de todos los conjuntos S_i . En otras palabras: $S_v = S_1 \cap S_2 \cap \dots \cap S_n$.

Desde este punto de vista, la ingeniería de requisitos puede considerarse como el proceso iterativo de *exploración* [Gause y Weinberg 1989] para determinar las restricciones correctas, es decir los requisitos, que determinen adecuadamente el espacio de sistemas válidos.

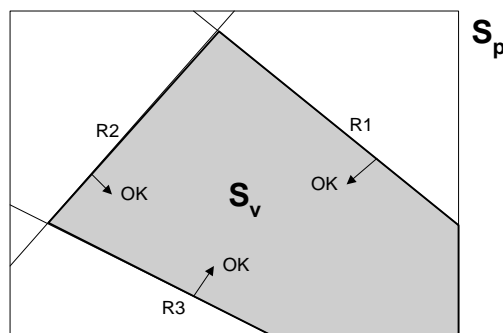


Figura 1.6: Los requisitos como restricciones



Figura 1.7: La ingeniería de requisitos como un proceso de comunicación

En la siguiente sección se utilizará esta interpretación de los requisitos para ilustrar algunas de sus propiedades deseables.

1.3.3 Propiedades deseables de los requisitos

Sobre las propiedades que deben tener los requisitos, o más concretamente las especificaciones de requisitos, hay una amplia bibliografía. A continuación enumeramos las propiedades que consideramos más importantes después de analizar las consideradas, entre otros, en [IEEE 1996], [IEEE 1993], [Davis 1993] y [Wieringa 1996].

- **Comprensible por clientes y usuarios:** como ya se comentó en la sección 1.2, en nuestra opinión la ingeniería de requisitos es básicamente un problema de descubrimiento y comunicación de las necesidades de clientes y usuarios, por lo que la propiedad más importante de una especificación es que sirva como canal de comunicación entre los participantes⁹ en el proceso de ingeniería de requisitos (ver figura 1.7).

Evidentemente, si una especificación de requisitos debe servir de canal de comunicación, es imprescindible que sea comprensible por parte de los clientes y usuarios. En nuestra opinión, la mejor forma

⁹El término inglés habitual para referirse a los participantes en la fase de ingeniería de requisitos es *stakeholder*. Literalmente, *stakeholder* significa *el que tiene una estaca*, aunque el significado real se deduce de la expresión *to have a stake in something* (*tener una estaca en algo*) que significa tener intereses en algo. Por lo tanto, un *stakeholder* es cualquier persona que tenga intereses en juego durante la fase de ingeniería de requisitos: clientes, usuarios, ingenieros de requisitos, desarrolladores, etc. En este trabajo utilizaremos el término castellano *participante* para designar a estas personas.

de lograr esta comunicación es pensar en la audiencia a la que van dirigidos los requisitos, de forma que para comunicarse con los clientes y usuarios lo mejor es utilizar requisitos-C y para hacerlo con los desarrolladores lo mejor es usar requisitos-D (ver figura 1.7).

En [Wieringa 1996] se considera esta propiedad como una característica, junto con la no ambigüedad, de la propiedad más importante que debe tener una especificación de requisitos: la *comunicabilidad*.

En [Davis 1993] y [Davis 1995] se advierte de un uso indiscriminado de notaciones formales que hagan imposible entender la especificación a los clientes y usuarios, que normalmente no tienen la formación necesaria para entender este tipo de notaciones.

En [IEEE 1996] se hace especial hincapié en que la especificación de requisitos debe hacer de puente entre los clientes y los desarrolladores, y que por lo tanto debe ser comprensible para ambos grupos.

En [IEEE 1993, pág. 9] se advierte de la necesidad de que la especificación de requisitos sea preparada conjuntamente por clientes, usuarios y desarrolladores, aunque no especifica ningún estilo para su redacción.

- **Correcta:** una especificación de requisitos es correcta si y sólo si todo requisito contenido en ella representa alguna propiedad requerida por el sistema a desarrollar [IEEE 1993, Davis 1993].

En otras palabras, ningún requisito debe ser innecesario. Deben formar un *conjunto único*, siguiendo la nomenclatura de [IEEE 1996].

Interpretando los requisitos como restricciones, añadir requisitos innecesarios implicaría reducir innecesariamente el espacio de sistemas válidos (ver figura 1.8, en la que el requisito R_4 es un requisito innecesario). Este espacio reducido de sistemas válidos S'_v , sería un subconjunto del verdadero espacio de sistemas válidos S_v .

- **No ambigua:** una especificación de requisitos no es ambigua si y sólo si todo requisito contenido en ella tiene una sola interpretación [IEEE 1993, Davis 1993].

Para evitar interpretaciones erróneas conviene que aquellos términos con más de una posible interpretación aparezcan en un glosario de términos en el que se especifique claramente su significado en el documento. En el caso de que se utilice lenguaje natural para especificar los requisitos, que es lo habitual al especificar los requisitos-C, se ha de tener especial cuidado con esta propiedad.

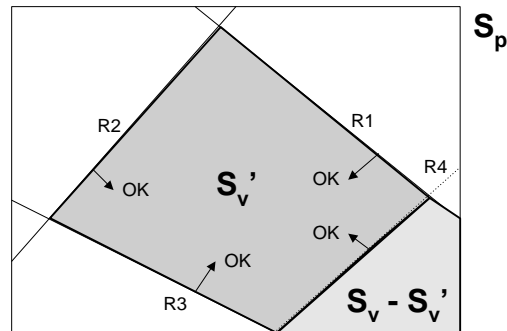


Figura 1.8: Requisitos innecesarios ($S_v' \subset S_v$)

En [Wieringa 1996], la no ambigüedad se considera como una característica de la comunicabilidad que debe tener la especificación de requisitos.

- **Completa:** una especificación de requisitos es completa si cumple las siguientes propiedades [IEEE 1993, Davis 1993]:

- Todo lo que se suponga que deba hacer el sistema a desarrollar está incluido en la especificación, es decir, no faltan requisitos.
- Todas las respuestas del sistema a entradas tanto válidas como inválidas están especificadas, es decir, está especificada toda posible conducta del sistema desde un punto de vista externo o de *caja negra*.
- Todas las páginas, figuras y tablas están numeradas, todas las unidades de medida están definidas, todas las referencias externas son comprobables y no hay elementos *por determinar* (TBD, *to be determined*). En otras palabras, la especificación está organizada, los requisitos son fáciles de localizar y el documento es *sintácticamente* correcto.

En el caso de que haya elementos por determinar, es necesario conocer la causa de su no determinación y quién es responsable de su solución.

En [IEEE 1996], se considera que una especificación es completa si incluye todos los requisitos identificados por el cliente y todos los requisitos necesarios para la definición del sistema.

En [Wieringa 1996], se considera una especificación como completa si no se omite ningún requisito. Desde un punto de vista práctico, esto significa que el cliente ha validado los requisitos, es decir, que está

de acuerdo con ellos y que confirma que no conoce más requisitos que sean necesarios. En [Wieringa 1996] también se considera que los requisitos deben estar priorizados por el cliente para considerar una especificación como completa.

Considerando los requisitos como restricciones, si no se especifican todos los requisitos se está definiendo un espacio incorrecto de sistemas válidos S_v' que incluiría sistemas no válidos. Esta situación incrementa el riesgo de acabar desarrollando sistemas que no satisfagan las necesidades reales de clientes y usuarios, es decir, sistemas que pertenezcan a $S_v' - S_v$ (ver figura 1.9, en la que se supone que no se ha identificado el requisito R_3).

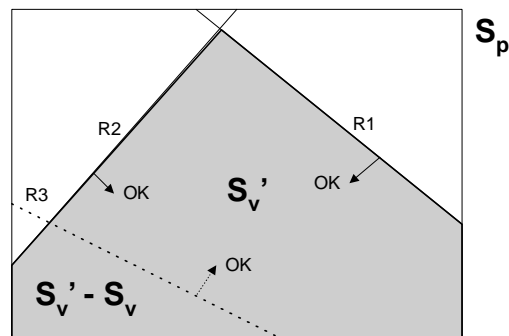


Figura 1.9: Requisitos incompletos ($S_v \subset S_v'$)

- **Consistente:** una especificación de requisitos es consistente externamente si y sólo si todo requisito contenido en ella no está en conflicto con otros documentos de nivel superior [Davis 1993]. Es consistente internamente si y sólo si no existen conflictos entre los requisitos que contiene.

Según [IEEE 1993] y [Davis 1993], los conflictos entre requisitos pueden ser de los siguientes tipos:

- Conflictos de conducta: dos o más requisitos especifican conductas distintas del sistema para las mismas condiciones y el mismo estímulo externo.
- Conflictos de términos: se utilizan términos distintos para referirse al mismo concepto.
- Conflictos de característica: dos o más requisitos especifican aspectos contradictorios para la misma característica del sistema.

- Conflictos temporales: dos o más requisitos exigen características temporales contradictorias al sistema.

En [IEEE 1996] también se comenta la necesidad de un mismo nivel de detalle y de un mismo estilo de redacción y de presentación de los requisitos para considerar una especificación como internamente consistente.

Otra propiedad relacionada con la consistencia interna que se comenta en [IEEE 1996] es la necesidad de que la especificación esté *limitada* (*bounded*), es decir, que el ámbito y el contexto en el que se definen los requisitos esté claramente identificado.

En [Wieringa 1996] se interpreta el término consistente como la propiedad de que pueda existir un sistema que satisfaga todos los requisitos. Evidentemente, si los requisitos son contradictorios es imposible que pueda existir un sistema que los satisfaga todos a la vez.

Considerando los requisitos como restricciones, la consistencia se interpretaría como la existencia de un espacio no vacío de sistemas válidos, es decir $S_v \neq \emptyset$ (ver figura 1.10).

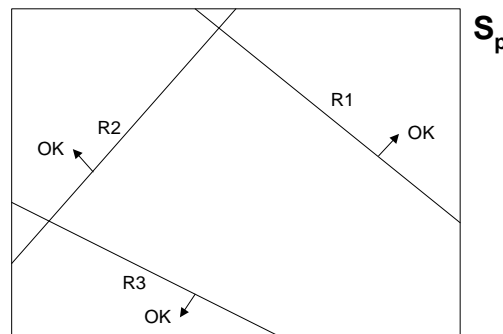


Figura 1.10: Requisitos inconsistentes ($S_v = \emptyset$)

- **Verificable:** una especificación de requisitos es verificable si y sólo si todo requisito contenido en ella es verificable, es decir, existe un proceso finito y de coste razonable por el que una persona o una máquina pueda comprobar que el sistema final cumple el requisito [Davis 1993, IEEE 1993].

Una condición absolutamente necesaria para que un requisito sea verificable es que no sea ambiguo y que se defina de forma medible, ya que no se puede comprobar algo que no se puede medir ni definir de forma precisa. Otra condición es que el proceso de verificación

sea computable en el sentido de Turing. Por ejemplo, un requisito que indique que el sistema nunca podrá entrar en un bucle infinito no puede verificarse porque no existe ningún proceso que sea capaz de hacerlo [Davis 1993].

Los procedimientos de observación para comprobar que el sistema cumple los requisitos son la base para las pruebas de aceptación del sistema por parte del cliente [Wieringa 1996].

Considerando los requisitos como restricciones, la verificabilidad y la no ambigüedad son necesarias para saber si un posible sistema s perteneciente al espacio de los sistemas posibles S_p cumple o no un determinado requisito R_i , es decir, si s pertenece a S_i o a \bar{S}_i .

- **Modificable:** una especificación de requisitos es modificable si y sólo si su estructura y estilo de redacción permite que los cambios se puedan realizar fácil, completa y consistentemente [Davis 1993, IEEE 1993].

Para conseguir esta propiedad, la especificación debe estar organizada coherentemente y debe contar con los índices y las tablas de referencias cruzadas oportunas, no debe ser redundante (debe estar *normalizada* según la terminología utilizada en [IEEE 1996]) y los requisitos deben expresarse individualmente y no de forma conjunta.

En el caso de que se opte por introducir redundancias para facilitar la legibilidad, dichas redundancias deben gestionarse con sumo cuidado, utilizando si es posible herramientas automatizadas [Davis 1993].

En [Wieringa 1996], la modificabilidad se considera como una propiedad, junto con la rastreabilidad, de la facilidad de mantenimiento de una especificación.

En [IEEE 1996] también se considera la necesidad de mantener las distintas versiones de la especificación que se vayan produciendo debido a los cambios, es decir, que la especificación sea *configurable*.

- **Rastreable:** una especificación de requisitos es rastreable si y sólo si para cada requisito contenido en ella se conoce su origen y puede referenciarse como origen en posteriores documentos durante el desarrollo, es decir, cada requisito puede rastrearse hacia atrás y hacia delante [IEEE 1993, Davis 1993].

Una condición necesaria para que un requisito pueda rastrearse hacia delante es que pueda referenciarse de manera única, normalmen-

te mediante algún tipo de código. La forma habitual de registrar la rastreabilidad¹⁰ son las matrices de rastreabilidad [DoD 1994, Davis 1993].

Como ya se comentó en el punto anterior, en [Wieringa 1996] se considera la rastreabilidad como una característica necesaria para la mantenibilidad de la especificación, ya que permite conocer el impacto de un cambio al poder saber a qué elementos afecta.

En [IEEE 1996] también se considera la rastreabilidad dentro de la propia especificación para poder conocer las relaciones entre los requisitos, de forma que éstos formen un *conjunto enlazado*.

- **Anotada con importancia y estabilidad:** una especificación de requisitos está anotada con importancia y estabilidad si y sólo si cada requisito contenido en ella está anotado con la importancia que tiene su cumplimiento para clientes y usuarios y la estabilidad que se espera del requisito, es decir, la probabilidad de que cambie durante el desarrollo [Davis 1993, IEEE 1993].

El conocer la importancia de un requisito permite decidir en qué orden implementarlos en el caso de que se hayan acordado entregar versiones sucesivas del producto. Por otro lado, la estabilidad permite a los diseñadores conocer qué grado de flexibilidad deben introducir en el sistema para soportar futuros cambios.

Como ya se comentó anteriormente, en [Wieringa 1996] la anotación de los requisitos con las preferencias de los clientes se considera como una característica de la compleción de la especificación.

- **Independiente del diseño y la implementación:** una especificación de requisitos es independiente del diseño y de la implementación si y sólo si no especifica una determinada descomposición del sistema (arquitectura) ni ningún aspecto de su posible implementación [Davis 1993, Wieringa 1996].

Sólo deben admitirse requisitos que limiten la libertad de los diseñadores y programadores en el caso de que el cliente lo solicite explícitamente (véase la figura 1.8 para una interpretación de requisitos innecesarios desde el punto de vista de las restricciones).

¹⁰En esta memoria se utilizará el término *rastreabilidad*, y no *trazabilidad* para referirse al termino inglés *traceability*, tal como se propone en [Fernández y Rueda 1998].

1.4 Clasificación de la investigación en ingeniería de requisitos

En [Zave 1997], se establece una clasificación de los esfuerzos de investigación en ingeniería de requisitos. Según dicho trabajo, los problemas en ingeniería de requisitos pueden encuadrarse dentro de los siguientes grupos:

- 1 *Problemas de investigación de los objetivos, funciones y restricciones de un sistema software*: recolección de información, análisis de la información y generación de estrategias alternativas.
 - 1.1 *Superar barreras de comunicación*: los ingenieros de requisitos deben tratar con un amplio abanico de personas con distintos conocimientos, intereses y objetivos personales. ¿Cómo comunicarse con personas con conocimientos, intereses y objetivos diferentes al nuestro?
 - 1.2 *Generar estrategias para convertir requisitos imprecisos en propiedades o conductas específicas*: un ejemplo de este tipo de requisitos imprecisos suelen ser la *amigabilidad con el usuario*, *seguridad*, *precisión*, *fiabilidad*, etc.
 - 1.3 *Entender prioridades y rangos de satisfacción*: muchos requisitos no son absolutos, sino que pueden satisfacerse parcialmente o si los recursos lo permiten.
 - 1.4 *Generar estrategias para asignar requisitos al sistema o a los distintos agentes de su entorno*: dado que los requisitos reales siempre se refieren al mundo real en el que se explotará el sistema software a desarrollar, antes de especificarlo se deben asignar objetivos, funciones y restricciones a los componentes y agentes que contribuirán a satisfacerlos. En otras palabras, decidir el ámbito del sistema: qué aspectos quedan dentro y qué aspectos se asignan al entorno.
 - 1.5 *Estimar costes, riesgos y duración*: es muy importante para decidir sobre los requisitos opcionales, que se satisfacen según los recursos de desarrollo.
 - 1.6 *Asegurar la compleción*: cómo asegurarse de que no se han dejado de lado personas, puntos de vista, hechos, etc. en la investigación del sistema.

- 2 *Problemas de especificar la conducta de un sistema software*: sintetizar información y elegir entre alternativas para crear una especificación software mínima y precisa.
 - 2.1 *Integrar múltiples vistas y representaciones*: los resultados de la investigación suelen ser diversos y tener conflictos, por lo que suele ser necesario algún tipo de negociación.
 - 2.2 *Evaluar estrategias alternativas para satisfacer requisitos*: la especificación del sistema supone tomar decisiones sobre la solución del problema. Normalmente, la solución no es única.
 - 2.3 *Obtener especificaciones completas, consistentes y no ambiguas*: es la única forma de asegurarse de que el sistema final cumplirá los requisitos iniciales.
 - 2.4 *Comprobar que el sistema especificado satisfará los requisitos*: diseñar estrategias de validación que aseguren que la especificación es acorde a los requisitos iniciales.
 - 2.5 *Obtener especificaciones útiles para las actividades de implementación y diseño*: es la mejor estrategia para introducir procesos de ingeniería de requisitos en las organizaciones.
- 3 *Problemas de gestionar la evolución de sistemas y familias de sistemas*.
 - 3.1 *Reutilizar ingeniería de requisitos durante fases evolutivas*: es decir, asegurarse de que los productos de la ingeniería de requisitos son mantenibles. Las técnicas más usadas son la rastreabilidad y la modularidad de la especificación.
 - 3.2 *Reutilizar ingeniería de requisitos para desarrollar sistemas similares*: es decir, asegurarse de que los productos de la ingeniería de requisitos se pueden aplicar a familias de sistemas.
 - 3.3 *Reconstruir requisitos*: ingeniería inversa de requisitos.

1.4.1 Clasificación de la tesis

De acuerdo a esta clasificación, en este trabajo se abordan algunos de estos problemas encuadrados dentro del modelo de procesos propuesto en el capítulo 2.

En la tabla 1.1 se resumen esquemáticamente los problemas que se abordan en este trabajo y en qué actividad del modelo de procesos propuesto serán tratados.

	Problemas Ingeniería de Requisitos	Actividad	¿Abordado?
1	1.1 Comunicación con clientes y usuarios	Elicitación	✓
	1.2 Eliminación de ambigüedades	Análisis	✓
	1.3 Priorización de requisitos	Elicitación	✓
	1.4 Determinar el ámbito del sistema	Elicitación	✓
	1.5 Estimación y planificación	<i>no contemplado</i>	–
	1.6 Asegurar la compleción	Validación	✓
2	2.1 Integración de múltiples vistas	Análisis	✓
	2.2 Evaluación de alternativas	<i>no contemplado</i>	–
	2.3 Obtener especificaciones de calidad	Análisis	✓
	2.4 Validación	Validación	✓
	2.5 Proceso "sin costuras"	Análisis	✓
3	3.1 Mantenibilidad de requisitos	Gestión de requisitos*	✓
	3.2 Reutilización de requisitos	Elicitación	✓
	3.3 Ingeniería inversa de requisitos	<i>no contemplado</i>	–

(*) No contemplada explícitamente en este trabajo

Tabla 1.1: Clasificación de la tesis

1.5 Bibliografía

[Bass *et al.* 1998] L. Bass, P. Clements, y R. Kazman. Nonfunctional Requirements Is a Dysfunctional Term. En *Software Architecture in Practice*, páginas 76–77. Addison–Wesley, 1998.

[Boehm *et al.* 1994] B. W. Boehm, P. Bose, E. Horowitz, y M.-J. Lee. Software Requirements as Negotiated Win Conditions. En *Proceedings of the First International Conference on Requirements Engineering*, 1994. Disponible en <http://sunset.usc.edu/TechRpts/Papers/NGPM-Requirements93.ps>.

[Booch *et al.* 1999] G. Booch, J. Rumbaugh, y I. Jacobson. *The Unified Modeling Language User Guide*. Addison–Wesley, 1999.

[Brackett 1990] J. W. Brackett. Software Requirements. Curriculum Module SEI–CM–19–1.2, Software Engineering Institute, Carnegie Mellon University, 1990. Disponible en <http://www.sei.cmu.edu>.

[Brooks 1995] F. P. Brooks, Jr. *The Mythical Man–Month: Essays on Software Engineering Anniversary Edition*. Addison–Wesley, 1995.

[Christel y Kang 1992] M. G. Christel y K. C. Kang. Issues in Requirements Elicitation. Technical Report CMU/SEI–92–TR–12, Software Engineering Institute, Carnegie Mellon University, 1992. Disponible en <http://www.sei.cmu.edu>.

- [Davis 1990] A. M. Davis. The Analysis and Specification of Systems and Software Requirements. En Thayer y Dorfman [1990], páginas 119–144.
- [Davis 1993] A. M. Davis. *Software Requirements: Objects, Functions and States*. Prentice–Hall, 2ª edición, 1993.
- [Davis 1995] A. M. Davis. *201 Principles of Software Development*. McGraw–Hill, 1995.
- [DoD 1994] DoD. *Military Standard 498: Software Development and Documentation*. Department of Defense of the United States of America, 1994. Disponible en http://www-library.itsi.disa.mil/mil_std/498_win3.exe.
- [Dorfman y Thayer 1990] M. Dorfman y R. H. Thayer, editores. *Standards, Guidelines, and Examples on System and Software Requirements Engineering*. IEEE Computer Society Press, 1990.
- [D’Souza y Wills 1999] D. F. D’Souza y A. C. Wills. *Objects, Components, and Frameworks with UML: The Catalysis Approach*. Addison–Wesley, 1999.
- [Durán *et al.* 1998] A. Durán, B. Bernárdez, M. Toro, y A. Ruiz. Una Propuesta Metodológica para la Elicitación de Requisitos de un Sistema Software. En *Actas de las III Jornadas de Trabajo Menhir*, Murcia, 1998. Universidad de Murcia.
- [ESP 1996] ESPITI – European User Survey Results. Informe Técnico ESI–1996–TR95104, European Software Institute, 1996. Disponible en <http://www.esi.es>.
- [Fernández y Rueda 1998] L. Fernández y M. J. Rueda. Terminología en Ingeniería del Software. *Novática*, (134), Julio–Agosto 1998.
- [GAO 1979] Contracting for Computer Software Development: Serious Problems Require Management Attention to Avoid Wasting Additional Millions. Report FGMSD–80–4, U. S. Government Account Office, Noviembre 1979. Este documento es difícil de encontrar, pero está comentado y referenciado, entre otros, en [Davis 1993], [Christel y Kang 1992] y [Goguen 1994].
- [Gause y Weinberg 1989] D. C. Gause y G. M. Weinberg. *Exploring Requirements: Quality Before Design*. Dorset House, 1989.

- [Goguen 1994] J. A. Goguen. Requirements Engineering as the Reconciliation of Social and Technical Issues. En *Requirements Engineering: Social and Technical Issues*, páginas 165–199. Academic Press, 1994. Disponible en <http://www.cse.ucsd.edu/~goguen>.
- [Hsia et al. 1993] P. Hsia, A. Davis, y D. Kung. Status Report: Requirements Engineering. *IEEE Software*, 10(6), Noviembre 1993.
- [IEEE 1990] IEEE. IEEE Standard Glossary of Software Engineering Terminology. IEEE Standard 610.12–1990, Institute of Electrical and Electronics Engineers, 1990.
- [IEEE 1993] IEEE. IEEE Recommended Practice for Software Requirements Specifications. IEEE/ANSI Standard 830–1993, Institute of Electrical and Electronics Engineers, 1993.
- [IEEE 1996] IEEE. IEEE Guide for Developing System Requirements Specifications. IEEE/ANSI Standard 1233–1996, Institute of Electrical and Electronics Engineers, 1996.
- [IEEE 1997] IEEE. *IEEE Software Engineering Standards Collection*. Institute of Electrical and Electronics Engineers, 1997.
- [MAP 1995] MAP. *Metodología de Planificación y Desarrollo de Sistemas de Información. MÉTRICA Versión 2.1*. Tecnos/Ministerio para las Administraciones Públicas, 1995.
- [Mazza et al. 1994] C. Mazza, J. Fairclough, B. Melton, D. de Pablo, A. Scheffer, y R. Stevens. *Software Engineering Standards*. Prentice–Hall, 1994. Disponible en <http://dxsting.cern.ch/sting/ESA.txt>.
- [Paulk et al. 1993] M. C. Paulk, B. Curtis, M. B. Chrissis, y C. V. Weber. Capability Maturity Model for Software, Version 1.1. Technical Report CMU/SEI–93–TR–024, Software Engineering Institute, Carnegie Mellon University, 1993. Disponible en <http://www.sei.cmu.edu>.
- [Pohl 1994] K. Pohl. The Three Dimensions of Requirements Engineering: A Framework and its Application. *Information Systems*, 3(19), Junio 1994.
- [Pohl 1997] K. Pohl. Requirements Engineering: An Overview. *Encyclopedia of Computer Science and Technology*, 36, 1997. Disponible en <http://sunsite.informatik.rwth-aachen.de/CREWS/reports96.htm>.

- [Pressman 1997] R. S. Pressman. *Ingeniería del Software. Un Enfoque Práctico*. McGraw-Hill, 4ª edición, 1997.
- [Rombach 1990] H. D. Rombach. Software Specifications: A Framework. Curriculum Module SEI-CM-11-2.1, Software Engineering Institute, Carnegie Mellon University, 1990. No está disponible en <http://www.sei.cmu.edu>, aunque aparece en [Dorfman y Thayer 1990].
- [Sawyer *et al.* 1997] P. Sawyer, I. Sommerville, y S. Viller. Requirements Process Improvement through The Phased Introduction of Good Practice. *Software Process – Improvement and Practice*, 3(1), 1997. Disponible en <http://www.comp.lancs.ac.uk/computing/research/cseg/reaims/publications.html>.
- [Sawyer y Kontoya 1999] P. Sawyer y G. Kontoya. SWEBOK: Software Requirements Engineering Knowledge Area Description. Informe Técnico Versión 0.5, SWEBOK Project, 1999. Disponible en <http://www.swebok.org>.
- [Sommerville y Sawyer 1997] I. Sommerville y P. Sawyer. *Requirements Engineering: A Good Practice Guide*. Wiley, 1997.
- [Thayer y Dorfman 1990] R. H. Thayer y M. Dorfman, editores. *System and Software Requirements Engineering*. IEEE Computer Society Press, 1990.
- [TSG 1995] TSG. *The CHAOS Report*. The Standish Group, 1995. Disponible en <http://www.standishgroup.com/chaos.html>.
- [Wieringa 1996] R. J. Wieringa. *Requirements Engineering: Frameworks for Understanding*. John Wiley & Sons, 1996.
- [Zave 1997] P. Zave. Classification of Research Efforts in Requirements Engineering. *ACM Computing Surveys*, 29(4), 1997.

Capítulo 2

Modelos de procesos de ingeniería de requisitos

En este capítulo se expone el modelo de procesos de ingeniería de requisitos que servirá de base para el resto de la tesis, se comparará con algunas propuestas similares y se comentará su relación con el modelo de madurez de procesos de ingeniería de requisitos REAIMS [Sawyer *et al.* 1997]. Para más información sobre la presencia de la ingeniería de requisitos en las principales normas de desarrollo de software puede consultarse el apéndice A.

2.1 Propuesta de modelo de procesos de ingeniería de requisitos

Como ya se comentó en el capítulo anterior, la ingeniería de requisitos es todavía una disciplina joven en la que aún no se han definido claramente la terminología, los procesos ni los productos. En nuestra opinión, es fundamental establecer un modelo de procesos para llegar a comprender los diversos problemas de la ingeniería de requisitos así como para poder desarrollar metodologías y herramientas de desarrollo.

El modelo de procesos de ingeniería de requisitos que se propone en esta tesis es la evolución de los presentados previamente, entre otros trabajos, en [Durán *et al.* 1999a], [Durán *et al.* 1999b] y [Durán *et al.* 2000]. Consta de tres actividades principales, *elicitación*, *análisis* y *validación*, y su principal característica es la *iteratividad*, tal como puede verse en la figu-

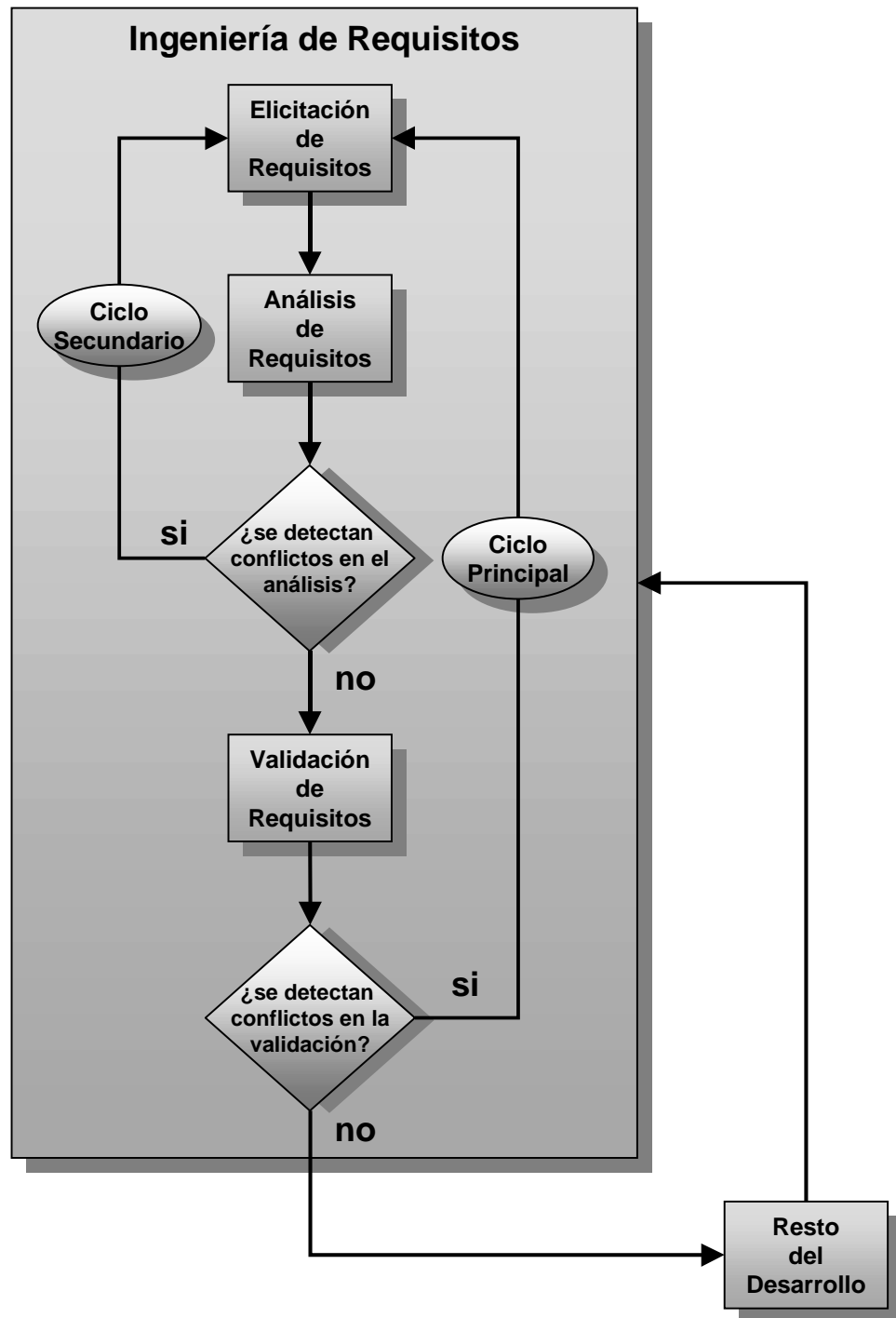


Figura 2.1: Modelo de procesos de ingeniería de requisitos: iteración de actividades

ra 2.1, en la que se ha expandido la fase de ingeniería de requisitos que aparecía como una caja negra en la figura 1.3 de la página 10.

Al igual que la mayor parte de las normas y autores consultados, asumimos que el proceso de elicitar requisitos, analizarlos y validarlos es iterativo por naturaleza, ya que es prácticamente imposible obtener todos los requisitos y que éstos tengan las propiedades que se comentaron en la sección 1.3.3 (pág. 16) sin tener que volver atrás en algún momento del proceso. Sin embargo, no por ello se debe dejar de intentar elicitar, y posteriormente analizar y validar, la mayor cantidad posible de requisitos en cada iteración, intentando alcanzar la situación ideal en la que el proceso sería lineal.

Al igual que en cualquier otra fase de la ingeniería del software, en la ingeniería de requisitos se incluyen también aspectos de gestión de recursos, verificación, control de calidad, control de configuración, etc. En este trabajo nos centraremos únicamente en los aspectos de la ingeniería de requisitos relacionados con el desarrollo, aunque las aportaciones que se realizan tienen repercusiones fuera del desarrollo, por ejemplo la rastreabilidad en los aspectos de gestión de requisitos [Sawyer y Kontoya 1999].

2.1.1 Ciclos de iteración

En el modelo propuesto se definen tres posibles ciclos de iteración, dos internos al proceso y uno externo.

2.1.1.1 Ciclo elicitación–análisis–validación

El ciclo principal *elicitación–análisis–validación* es un ciclo interno que indica la posibilidad de que durante el proceso de validación de los requisitos por parte de los clientes y usuarios aparezcan conflictos o nuevos requisitos que hasta entonces estaban ocultos. En esas circunstancias, es necesario resolver dichos conflictos y consensuar los nuevos requisitos mediante nuevas reuniones de elicitación/negociación, repitiendo a continuación las actividades de análisis y validación.

2.1.1.2 Ciclo elicitación–análisis

El segundo ciclo interno, *elicitación–análisis*, se trata de un ciclo que indica la posibilidad de que durante la realización del análisis de los requi-

sitos elicitados se descubran conflictos o deficiencias en dichos requisitos, lo que puede provocar la necesidad de nuevas reuniones de elicitación/negociación y el posterior análisis de sus resultados.

2.1.1.3 Ciclo ingeniería de requisitos–resto del desarrollo

El tercer ciclo, entre la ingeniería de requisitos y el resto del desarrollo, muestra la posibilidad de que durante el resto del desarrollo sea necesario volver a alguna de las actividades de ingeniería de requisitos, posiblemente porque se detecte la necesidad de renegociar algunos requisitos de difícil implementación, porque aparezcan nuevos requisitos durante el desarrollo, etc.

2.1.2 Actividades del proceso

Las tres actividades que componen el modelo de procesos de ingeniería de requisitos propuesto son las siguientes (ver figura 2.2):

2.1.2.1 Elicitación de requisitos

Desde nuestro punto de vista, esta actividad es la más importante de la ingeniería de requisitos. En el modelo propuesto, esta actividad es en la que se mantiene la interacción entre clientes y usuarios e ingenieros de requisitos más importante. Los principales objetivos que se deben intentar alcanzar en esta actividad y sus entradas y salidas son los siguientes:

• Objetivos

- **Conocer el dominio del problema:** es fundamental que los ingenieros de requisitos conozcan el dominio del problema, de forma que puedan entenderse con los clientes y usuarios y que sean capaces de transmitir dicho conocimiento al resto del equipo de desarrollo [Kovitz 1998].
- **Descubrir las necesidades reales de clientes y usuarios:** además de aquellas necesidades explícitamente manifestadas por los clientes y usuarios, es muy importante llegar a descubrir el *conocimiento tácito*, es decir, aquellas necesidades que la mayor parte de las veces se asumen y toman por implícitas [Goguen y Linde 1993, Kovitz 1998].

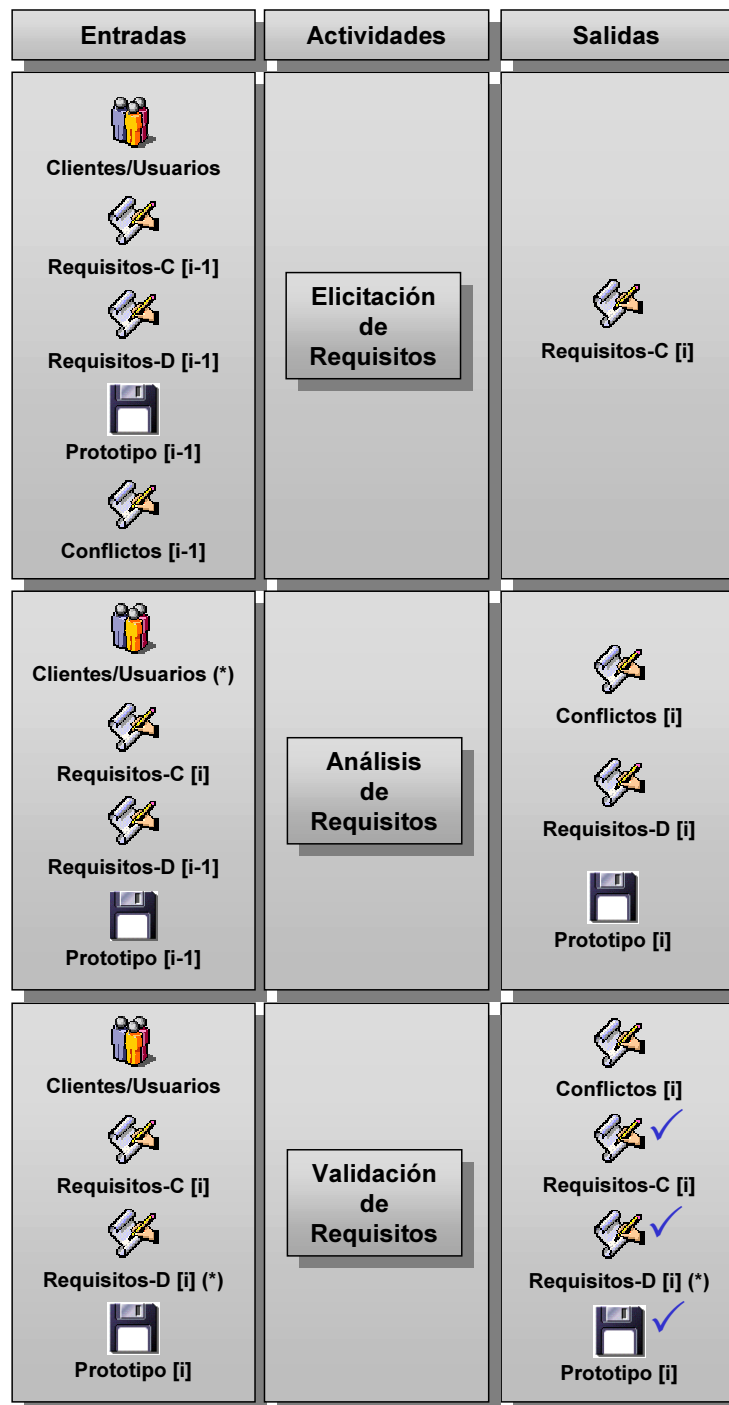


Figura 2.2: Modelo de procesos de ingeniería de requisitos: entradas–actividades–salidas

- **Consensuar los requisitos entre los propios clientes y usuarios:** puede que distintos grupos de clientes y usuarios presenten distintas necesidades que sean contradictorias entre sí. Durante la elicitación, normalmente después de la primera iteración, es necesario *negociar* entre los distintos participantes hasta obtener una visión común de los requisitos [Boehm *et al.* 1994, Parets-Llorca y Grünbacher 1999].

Otros autores [Pohl 1997, Sawyer y Kontoya 1999] separan las actividades de negociación de las de elicitación. En nuestra opinión, preferimos un modelo de procesos más sencillo en el que la elicitación englobe también los aspectos de negociación, dado que la negociación puede entenderse como la elicitación de la información necesaria para llegar a un acuerdo y además suelen aparecer nuevos requisitos cuando es necesario negociar. En cualquier caso, ambas actividades requieren reuniones entre los participantes involucrados en las que predominan los aspectos sociales sobre los técnicos.

• Entradas y salidas

- **Entradas:** en la primera iteración, la entrada principal de esta actividad es la información proveniente de clientes y usuarios. En iteraciones posteriores también se toman como entradas los requisitos–C, los requisitos–D, el prototipo y los conflictos aparecidos en la iteración anterior del proceso¹.
- **Salidas:** la salida de esta actividad es una nueva versión (o la primera si se trata de la primera iteración) de los requisitos–C, que serán analizados en la siguiente actividad.

Otros autores [Raghavan *et al.* 1994, Sawyer y Kontoya 1999, Pohl 1997] añaden una actividad diferenciada a la que denominan *especificación* o *documentación*, en la que se registran los requisitos en uno o más documentos. En el modelo propuesto se ha optado por asumir que el proceso de registrar los requisitos es implícito, por un lado a la elicitación en el caso de los requisitos–C, y por otro lado al análisis en el caso de los requisitos–D. De esta forma, se usará el término *especificación* con el sentido que se le da en [IEEE 1990]:

Especificación de requisitos: *un documento que especifica los requisitos para un sistema o componente. Se incluyen típicamen-*

¹En la figura 2.2 se asume que la iteración actual es la *i*-ésima, de forma que los productos de la iteración anterior tienen el subíndice *i-1* y los de la iteración actual *i*.

te requisitos funcionales, de rendimiento, de interfaz, de diseño y normas de desarrollo.

2.1.2.2 Análisis de requisitos

Esta es la actividad a la que más importancia se la ha otorgado tradicionalmente, aunque desde nuestro punto de vista la elicitación, e incluso la validación podrían considerarse como más importantes de cara a obtener un producto final que satisfaga las necesidades de clientes y usuarios.

Los objetivos principales de esta actividad y sus entradas y salidas son los siguientes:

• Objetivos

- **Detectar conflictos en los requisitos–C:** los requisitos–C suelen ser información proveniente de distintas fuentes y normalmente presentan contradicciones o ambigüedades debido a su naturaleza informal. Durante el análisis de los requisitos–C, normalmente mediante la construcción de un modelo, es habitual que al incrementar la precisión con la que es necesario expresar los requisitos aparezcan dichas contradicciones o ambigüedades que deberán resolverse en nuevas reuniones de elicitación mediante algún mecanismo de negociación.
- **Profundizar en el conocimiento del dominio del problema:** por mucho que el ingeniero de requisitos aprenda sobre el dominio del problema, siempre quedarán aspectos desconocidos. Construir un modelo suele conllevar un incremento en el grado de conocimiento del problema que puede facilitar el proceso de construir un producto útil para clientes y usuarios. En la construcción del modelo podrían participar aquellos clientes y usuarios con los conocimientos apropiados².
- **Establecer las bases para el diseño:** normalmente, los modelos obtenidos durante la realización del análisis de los requisitos–C suelen constituir las bases para las actividades de diseño. En el caso de que se utilicen notaciones similares en análisis y diseño (por ejemplo orientadas a objetos), se consigue lo que se suele denominar *proceso*

²Esta participación opcional está indicada en la figura 2.2 por un asterisco en el icono de clientes y usuarios. No tiene sentido que aquellos clientes y usuarios que no conozcan las técnicas de modelado participen en esta actividad.

sin costuras (*seamless process* en inglés), es decir, una evolución gradual de los modelos abstractos de análisis hacia modelos más cercanos a la implementación.

- **Entradas y salidas**

- **Entradas:** las entradas de esta actividad son los requisitos–C elicitados en la actividad anterior, los requisitos–D y el prototipo de la iteración previa e información proveniente de clientes y usuarios en el caso de que tengan conocimientos de la técnicas de análisis apropiadas.
- **Salidas:** la salida de esta actividad es una nueva versión (o la primera si se trata de la primera iteración) de los requisitos–D y del prototipo, que en el caso de que se utilicen especificaciones formales ejecutables para expresar los requisitos–D puede generarse automáticamente, así como posibles conflictos encontrados en los requisitos–C analizados, lo que provocaría que se repitiese el ciclo elicitación–análisis.

2.1.2.3 Validación de Requisitos

En esta actividad se debe confirmar que los requisitos–C, una vez analizados y resueltos los posibles conflictos, corresponden realmente a las necesidades de clientes y usuarios. Se trata de una actividad que debe ser realizada principalmente por clientes y usuarios, aunque los ingenieros de requisitos pueden ayudar en el proceso mediante las explicaciones oportunas o mediante el uso de prototipos [Pohl 1997, Sawyer y Kontoya 1999].

Los objetivos principales de esta actividad y sus entradas y salidas son los siguientes:

- **Objetivos**

- **Asegurarse de que los requisitos describen el producto deseado:** aunque las actividades de elicitación y análisis se hayan realizado correctamente, siempre es necesario confirmar que los requisitos obtenidos se corresponden realmente con los que los clientes y usuarios

desean, de forma que se evite la situación en la que el producto final, que puede ser técnicamente correcto, no es satisfactorio.

Las actividades de validación conllevan generalmente la elicitación de nuevos requisitos debido a que, a medida que el nuevo sistema se va perfilando, suelen ir apareciendo nuevas necesidades que hasta entonces estaban ocultas, sobre todo mediante la utilización de prototipos [Davis 1995a].

• Entradas y salidas

- **Entradas:** las entradas de esta actividad son los requisitos–C, los requisitos–D, en el caso de que los clientes y usuarios tengan conocimientos suficientes para comprender las notaciones utilizadas³, el prototipo y la información de validación proveniente de clientes y usuarios.
- **Salidas:** la salida de esta actividad son las versiones validadas, total o parcialmente, de los requisitos–C, requisitos–D, en el caso de que los clientes y usuarios tengan conocimientos suficientes para comprender las notaciones utilizadas, y del prototipo⁴. En el caso de que se detecten conflictos o se eliciten nuevos requisitos se repite el ciclo completo de actividades.

2.1.3 Productos del proceso

Los tres productos principales que se incluyen en el modelo de procesos propuesto son los siguientes:

- **Requisitos–C:** los requisitos–C, es decir, los requisitos desde el punto de vista del cliente, son el resultado principal de la actividad de elicitación. Estos requisitos deben expresarse de forma que todos los participantes en el proceso de ingeniería de requisitos sean capaces de entenderlos, especialmente los clientes y usuarios.

Para conseguir ese objetivo, en este trabajo se propone utilizar principalmente casos de uso [Jacobson *et al.* 1993] expresados en lenguaje

³Expresado en la figura 2.2 por un asterisco en el icono de requisitos–D.

⁴Expresadas en la figura 2.2 por una marca de validación en los iconos de requisitos–C, requisitos–D y prototipo.

natural, que a priori es el único lenguaje común entre todos los participantes. Para evitar los problemas inherentes al uso del lenguaje natural se propone el uso de plantillas y de patrones que faciliten su uso. En el capítulo 3 se describen con detalle todos los aspectos relacionados los requisitos–C.

- **Requisitos–D:** los requisitos–D, es decir, los requisitos desde el punto de vista del desarrollador, junto con el prototipo y los posibles conflictos, son el resultado principal de la actividad de análisis. La forma de expresar estos requisitos suele consistir en la elaboración de un modelo del sistema a desarrollar basado en los requisitos–C. Los modelos pueden realizarse mediante técnicas estructuradas [Yourdon Inc. 1993], técnicas orientadas a objetos [Rumbaugh *et al.* 1991, Booch *et al.* 1999], lenguajes formales [Ratcliff 1994] o mezclas de varios de ellos.
- **Prototipo:** la construcción de un prototipo del sistema a desarrollar puede facilitar enormemente tanto la validación de los requisitos por parte de los clientes y usuarios como la elicitación de nuevos requisitos. Los prototipos suelen ser de dos tipos, *de usar y tirar* o *evolutivos* [Gomaa 1990]. Los prototipos de usar y tirar suelen utilizarse principalmente para elicitar y validar requisitos relacionados con la interfaz de usuario, mientras que los evolutivos suelen centrarse más en los requisitos funcionales.
- **Conflictos:** es importante registrar los conflictos que vayan surgiendo para poder acometer su resolución de forma organizada, involucrando a los participantes en el proceso de negociación. En el modelo propuesto, se asume que los conflictos aparecerán principalmente durante la actividad de análisis y se resolverán mediante negociación en las actividades de elicitación.

2.2 Otros modelos de procesos

En esta sección se comentan brevemente algunos de los modelos de procesos para ingeniería de requisitos existentes y que han servido de base para el modelo descrito en este capítulo.

2.2.1 El modelo de Pohl

El modelo de Pohl [Pohl 1997] es un modelo iterativo en el que se definen las cuatro actividades que pueden verse en la figura 2.3, en la que aunque el orden de realización de las actividades puede ser cualquiera, en [Pohl 1997] se asume una secuencia en la que los requisitos son elicitados, a continuación son negociados entre los participantes, se integran con el resto de la documentación y finalmente se validan y verifican para asegurar que corresponden con las necesidades reales de los clientes y usuarios y que no presentan conflictos con los demás requisitos.

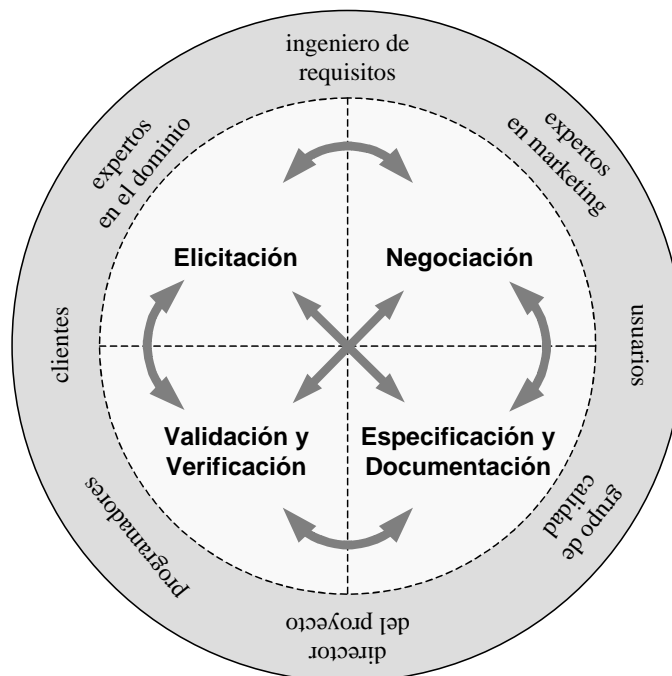


Figura 2.3: Modelo de procesos de ingeniería de requisitos de Pohl

Las características principales de las cuatro actividades de este modelo son las siguientes:

2.2.1.1 Elicitación de requisitos

Para Pohl, el objetivo de la elicitación es hacer explícito el conocimiento oculto sobre las necesidades de clientes y usuarios y el sistema a desarrollar de forma que todos los participantes en el problema sean capaces de entenderlo. Asumiendo las dimensiones de la ingeniería de requisitos pro-

puestas también por Pohl (ver figura 1.4 en la pág. 11), esta actividad haría avanzar el proceso en la dimensión de compleción, ya que incrementa el conocimiento sobre el sistema a desarrollar.

Aunque implícitamente, en este modelo se asume que durante la realización de las actividades de elicitación es necesario identificar a las fuentes de información, conocer lo mejor posible el dominio del problema, reutilizar especificaciones de requisitos similares en la medida de lo posible y utilizar las técnicas habituales de elicitación como son las entrevistas, casos de uso, cuestionarios, prototipos, etc.

2.2.1.2 Negociación de requisitos

El objetivo de esta actividad es alcanzar acuerdos entre todos los participantes sobre los requisitos elicitados en la actividad anterior, avanzando en la dimensión de acuerdo del proceso. Para ello, Pohl propone tener en cuenta cuatro factores:

- Hacer explícitos los conflictos y evitar los conflictos emocionales entre los participantes, de forma que quede claro *qué* es lo que se negocia y que dicha negociación no se vea afectada por motivos personales.
- Hacer explícitos para cada conflicto las alternativas, las argumentaciones y las razones subyacentes que los provocan, de forma que la negociación pueda basarse en las raíces del conflicto.
- Asegurarse de que se toman las decisiones correctas, de forma que la mayoría de los participantes estén de acuerdo en los resultados de la negociación y no se sientan desplazados del proceso.
- Asegurarse de involucrar a las personas adecuadas en el momento adecuado, para evitar tener que volver a replantear las negociaciones porque alguno de los participantes afectados no participó en las negociaciones oportunas.

2.2.1.3 Especificación/Documentación de requisitos

En esta actividad deben documentarse los requisitos elicitados y negociados, para lo que Pohl propone que no se utilice una sola notación sino tantas como sea necesario para que todos los participantes los entiendan,

avanzando en la dimensión de formalidad del proceso. En esto coincide con otros autores como [Davis 1995a] o [Sawyer y Kontoya 1999].

2.2.1.4 Validación/Verificación de requisitos

El propósito de esta actividad es comprobar que los requisitos documentados corresponden con las necesidades de los clientes y usuarios (*validación*) y comprobar que la especificación cumple los criterios de calidad oportunos (*verificación*), haciendo avanzar el proceso en las tres dimensiones descritas en [Pohl 1994] y en la sección 1.2.1 en la página 10.

Para los aspectos de validación puede recurrirse a prototipos, mientras que para la verificación pueden utilizarse verificaciones formales u otro tipo de técnicas.

2.2.2 El modelo espiral

El modelo espiral [Sawyer *et al.* 1997, Sommerville y Sawyer 1997], representado gráficamente en la figura 2.4, está basado en el modelo espiral de Boehm para la ingeniería de requisitos [Boehm *et al.* 1994] y el *Inquiry-Based Model* de Potts [Potts *et al.* 1994b, Potts *et al.* 1994a]. En este modelo se asume una naturaleza iterativa del proceso y la dificultad de establecer un punto de terminación del mismo, dado que los requisitos nunca llegarán a ser perfectos.

A parte de las tres actividades que se describen a continuación, el modelo asume que existe una cuarta, la *gestión de requisitos*, que se realiza durante todo el proceso y que se encarga de gestionar la obtención incremental de los requisitos y los inevitables cambios a los que están sujetos.

2.2.2.1 Elicitación de requisitos

En esta actividad, distintas fuentes de información como clientes, usuarios, expertos en el dominio, etc. son consultadas para entender el dominio del problema y establecer los requisitos del sistema a desarrollar. Los requisitos elicitados puede que no sean completos y pueden estar expresados de forma vaga o no estructurada.

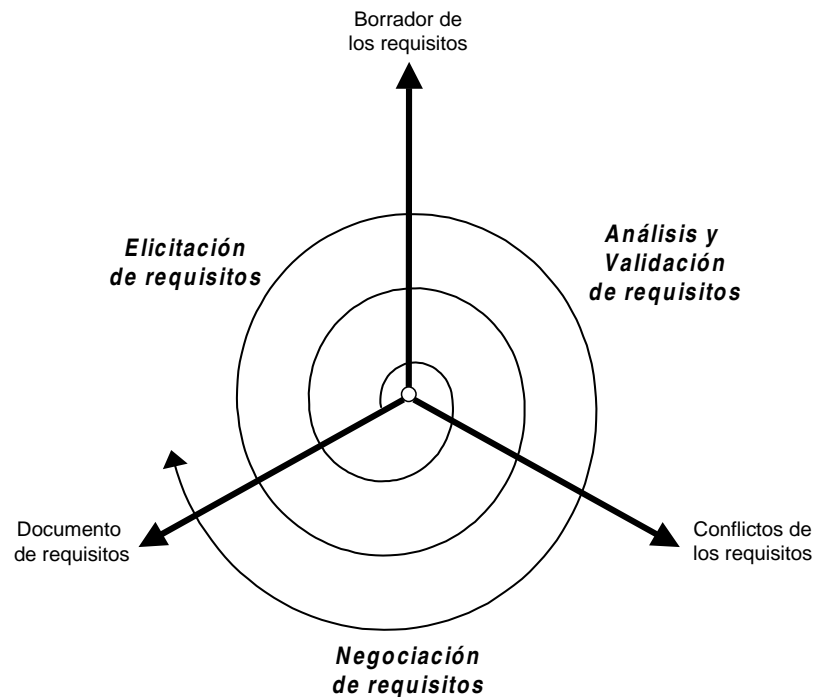


Figura 2.4: Modelo espiral de procesos de ingeniería de requisitos

2.2.2.2 Análisis y validación de requisitos

Los requisitos elicitados se integran y analizan, lo que suele provocar la identificación de requisitos que faltan, inconsistencias y conflictos entre los requisitos.

Aunque en [Sawyer *et al.* 1997] se denomina a esta actividad *análisis y validación de requisitos*, sin embargo no se hace referencia a ningún tipo de actividad de validación tal como se entiende el término en esta tesis, es decir, la confirmación por parte de clientes y usuarios de que los requisitos reflejan realmente sus necesidades y especifican el sistema que ellos desean.

Aplicando la terminología adoptada, el autor se refiere realmente al proceso de verificación (ver capítulo 5, pág. 189).

2.2.2.3 Negociación de requisitos

En esta actividad, los conflictos identificados durante el análisis deben resolverse llegando a acuerdos entre los participantes en el proceso, para lo que suele ser necesario elicitar nueva información.

2.2.3 El modelo SWEBOK

El proyecto SWEBOK (*Software Engineering Body of Knowledge*) es un proyecto conjunto del IEEE y de la ACM para producir un *cuerpo de conocimiento* sobre ingeniería de software que sienta las bases de dicha ingeniería como una profesión [Bourque *et al.* 1999]. Dentro de las 10 áreas de conocimiento que han establecido, la novena corresponde a la ingeniería de requisitos [Sawyer y Kontoya 1999], dentro de cuya descripción se propone el modelo de procesos que puede verse en la figura 2.5, al que hay que añadir las actividades de gestión de requisitos que no se recogen en el gráfico.

2.2.3.1 Elicitación de requisitos

Esta actividad se considera como la primera que es necesario realizar para lograr entender los problemas que hay que resolver, y es fundamentalmente una actividad *humana*. Para acometerla con ciertas garantías es necesario tener en cuenta los siguientes puntos:

- **Los objetivos:** pueden considerarse como los requisitos de alto nivel que deberá cumplir el sistema a desarrollar, por lo que es especialmente crítica su identificación en los comienzos del proceso.
- **Conocimiento del dominio:** es fundamental conocer el dominio del problema para poder inferir el *conocimiento tácito* que los participantes no suelen hacer explícito, además de para facilitar la comunicación.

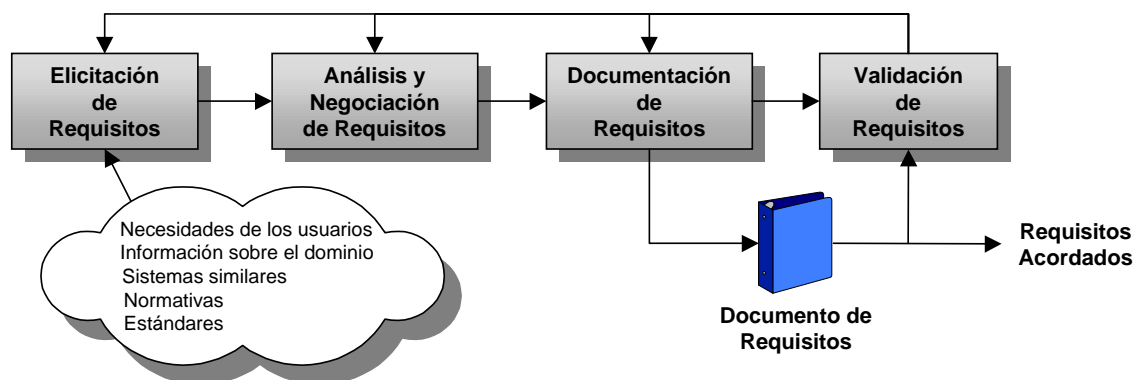


Figura 2.5: Modelo SWEBOK de procesos de ingeniería de requisitos

- **Participantes (*stakeholders*):** es preciso identificar a todos los participantes que tengan algún tipo de interés en el desarrollo y tener en cuenta los puntos de vista de los distintos grupos.
- **Entorno operacional:** el sistema a desarrollar deberá funcionar en un entorno que es necesario conocer para poder identificar las necesidades de interoperabilidad con otros sistemas ya existentes.
- **Entorno organizacional:** además del entorno operacional, el sistema afectará al entorno organizacional, que es necesario conocer para evitar que surjan problemas con los procesos de negocio.

Para la realización de esta actividad se puede recurrir a técnicas como las entrevistas, la observación mediante inmersión en el negocio del cliente [Goguen y Linde 1993], el uso de escenarios o casos de uso y la utilización de prototipos, que también suelen utilizarse para las actividades de validación.

2.2.3.2 Análisis y negociación de requisitos

En esta actividad se pretende detectar y resolver los conflictos entre los requisitos, determinar los límites del sistema y cómo interactuará con su entorno y transformar los requisitos de usuario⁵ en requisitos software⁶.

En este modelo se propone que en esta actividad se clasifiquen los requisitos, se realicen modelos conceptuales y se negocien los conflictos detectados, tareas que se describen a continuación.

- **Clasificación de requisitos**
Los autores consideran importante clasificar los requisitos para ayudar en las labores de negociación. Los criterios de clasificación son: capacidad/restricción, prioridad, coste/impacto, volatilidad/estabilidad y requisito de producto/requisito de proceso.
- **Modelado conceptual**
El objetivo del modelado conceptual es ayudar a la comprensión del problema, aunque normalmente no puede evitarse iniciar el diseño de la solución. Existen múltiples técnicas de modelado conceptual,

⁵Requisitos-C en la terminología adoptada en esta tesis.

⁶Requisitos-D en la terminología adoptada en esta tesis.

por lo que los autores proponen considerar factores como la naturaleza del problema, la experiencia del ingeniero de requisitos en el uso de una determinada técnica, si el cliente impone como requisito la utilización de una determinada metodología o la disponibilidad de metodologías y herramientas que soporten una determinada notación.

Independientemente de la notación de modelado utilizada, los autores proponen realizar siempre un modelo de los límites del sistema para entender mejor el entorno y las interfaces necesarias.

- **Negociación de requisitos**

La negociación de requisitos, también denominada *resolución de conflictos*, se ocupa de resolver los problemas que puedan surgir en los requisitos, bien porque haya peticiones por parte de clientes y usuarios que sean incompatibles, bien porque no se disponga de los recursos necesarios para la realización de ciertos aspectos del sistema, etc.

Para resolver estos conflictos es necesario consultar con todos los participantes afectados y registrar las decisiones tomadas y quién las tomó. Es necesario añadir que los autores asumen que los conflictos pueden aparecer no sólo durante el análisis, sino también durante la validación.

2.2.3.3 Documentación de requisitos

Los documentos de requisitos son el medio habitual para el registro y la comunicación de los requisitos. Los autores consideran deseable que los requisitos-C y los requisitos-D vayan en documentos separados.

Dentro de las actividades relacionadas con la documentación, los autores se remiten a la gestión de requisitos para aspectos como el control de versiones debido a los cambios en los requisitos.

2.2.3.4 Validación de requisitos

En esta actividad se debe comprobar los documentos de requisitos para detectar omisiones, conflictos y ambigüedades no detectadas en el análisis y también se debe comprobar que los requisitos siguen las normas de calidad establecidas. En otras palabras, los autores combinan validación y verificación en una sola actividad.

Para realizar esta actividad se proponen revisiones técnicas de los requisitos basadas en listas de comprobación, el uso de prototipos, verificación formal de especificaciones, etc.

2.2.3.5 Gestión de requisitos

La gestión de requisitos, aunque no está reflejada en la figura 2.5, se realiza durante todas las actividades de ingeniería de requisitos. Su objetivo es gestionar los cambios y el mantenimiento de los requisitos para que representen el sistema que se va a desarrollar o que se ha desarrollado.

Para conseguir estos objetivos, además de definir procedimientos para controlar los cambios y utilizar técnicas de gestión de configuración, los autores apuntan como factores importantes los atributos de los requisitos (identificadores, justificación, las fuentes del requisito, etc.) y la rastreabilidad para poder realizar análisis de impacto cuando se produce un cambio.

Coinciden con Davis [Davis 1995b], en que a pesar de la sencillez teórica del concepto de rastreabilidad (construir un grafo dirigido acíclico entre los requisitos), a menos que se automatice el proceso no suele llevarse a la práctica.

2.3 Comparación de modelos de procesos de ingeniería de requisitos

Comparando el modelo de procesos propuesto en este trabajo, los otros tres modelos estudiados y otras propuestas consultadas [Christel y Kang 1992, Raghavan *et al.* 1994] pueden realizarse el análisis que se recoge esquemáticamente en los siguientes puntos:

- **La ingeniería de requisitos es un proceso iterativo:** todos los modelos consultados y el propuesto en esta tesis incorporan el hecho de que la ingeniería de requisitos, en nuestra opinión por ser básicamente un proceso de descubrimiento y comunicación, no puede realizarse de forma lineal.

En el modelo propuesto se han identificado al menos tres posibles ciclos (ver sección 2.1.1). En los modelos de Pohl y SWEBOK se contempla cualquier tipo de ciclo, mientras que en el modelo espiral

los ciclos han de tener siempre la forma *elicitación–análisis/validación–negociación*. En ninguno de estos tres modelos se contempla el posible ciclo con el resto del desarrollo, que si es contemplado en el modelo propuesto en este capítulo.

- **Los límites de las actividades son difíciles de establecer:** aunque disponer de un modelo de procesos es siempre beneficioso, hay que tener en cuenta que los límites de las actividades no son claros en ingeniería de requisitos.

Por ejemplo, es posible comenzar a construir modelos (actividad de análisis) durante las sesiones de elicitación, e incluso si la experiencia en el dominio del problema así lo aconseja, podrían validarse requisitos en dichas sesiones. También es habitual que durante las sesiones de validación, sobre todo usando prototipos, se eliciten nuevos requisitos que hasta entonces estaban ocultos.

- **Los productos del proceso no están claramente definidos:** la mayor parte de los modelos de procesos propuestos no definen claramente los productos que se supone que se debe obtener como realización de dichos procesos. En el modelo propuesto se definen claramente cuatro productos: requisitos–C, requisitos–D, prototipo y conflictos. Sólo el modelo SWEBOK propone la separación explícita de los requisitos–C y requisitos–D en documentos distintos, aunque sólo como recomendación.

2.4 El modelo de madurez de proceso REAIMS

El modelo de madurez de proceso de ingeniería de requisitos REAIMS es el principal resultado del proyecto europeo REAIMS [Sawyer *et al.* 1997, Sommerville y Sawyer 1997]. Este modelo define tres niveles de madurez que se corresponden con los tres primeros niveles del CMM [Paulk *et al.* 1993] (ver figura 2.6).

Las principales características de los tres niveles definidos son las siguientes:

- **Nivel 1: Inicial**

Las organizaciones que se encuentran en este nivel de madurez no tienen definido un proceso de ingeniería de requisitos y suelen tener los problemas habituales ya comentados en el capítulo 1. Normalmente no producen los documentos de requisitos dentro de los

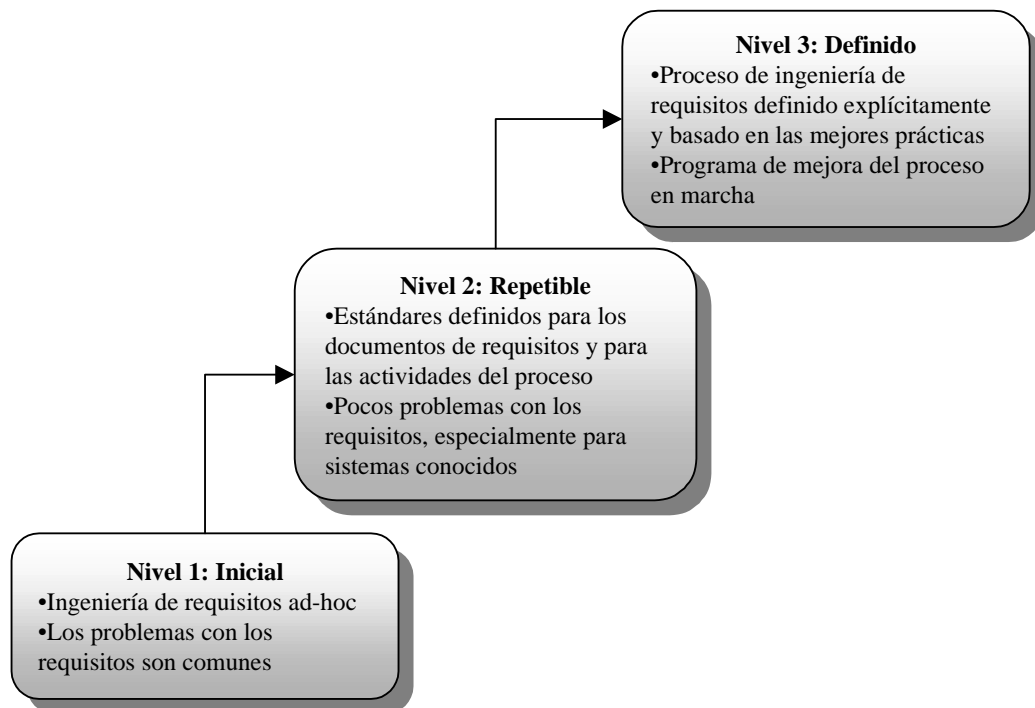


Figura 2.6: Niveles de madurez de proceso de ingeniería de requisitos

plazos y el presupuesto previstos y dependen de las habilidades y experiencia de sus ingenieros de requisitos.

- **Nivel 2: Repetible**

En este nivel se encuentran las organizaciones que tienen definidas normas para los documentos y las descripciones de los requisitos y han introducido políticas y procedimientos para su gestión, probablemente utilizando algún tipo de herramienta automatizada. Sus documentos de requisitos suelen ser más consistentes y suelen realizarse dentro de los plazos.

- **Nivel 3: Definido**

En este nivel, las organizaciones tienen un proceso de ingeniería de requisitos definido basado en las prácticas y técnicas que se han identificado como buenas en experiencias previas. Tienen un programa de mejora de proceso en marcha y pueden evaluar objetivamente la adopción de nuevos métodos y técnicas.

Para ir alcanzando estos niveles, el modelo propone la adopción de determinadas prácticas que se clasifican en tres grupos: *básicas*, *intermedias*

y *avanzadas*. Para clasificar a una organización en uno de los tres niveles de madurez, se comprueba para cada una de las prácticas si dicha práctica está *normalizada*, si es de *uso normal*, si se usa *a discreción del director del proyecto* o si no se usa *nunca*.

Una vez comprobada el estado de la implantación de las prácticas en la organización se identifican las áreas donde residen las debilidades del proceso y se clasifica a la organización en un nivel de madurez de proceso de acuerdo a un baremo establecido.

Durante la realización del proyecto REAIMS se constató que la mayor parte de las empresas europeas estudiadas estaban en un nivel de madurez inicial.

El modelo también define un proceso para la mejora de procesos basado en cuatro cuestiones: *¿Cuáles son los problemas con los procesos actuales?*, *¿Cuáles son los objetivos de mejora?*, *¿Cómo pueden introducirse las mejoras de procesos para alcanzar dichos objetivos?* y *¿Cómo deberían controlarse y gestionarse las mejoras?*.

Para organizaciones sin ningún tipo de proceso de ingeniería de requisitos se proponen las siguientes diez guías básicas:

- 1 Definir una estructura normalizada del documento de requisitos.
- 2 Hacer el documento fácil de cambiar.
- 3 Identificar de manera única cada requisito.
- 4 Definir políticas para la gestión de requisitos.
- 5 Definir plantillas normalizadas para la descripción de requisitos.
- 6 Usar el lenguaje de forma simple, consistente y concisa.
- 7 Organizar revisiones formales de los requisitos.
- 8 Definir listas de comprobación para la validación.
- 9 Usar listas de comprobación para el análisis de los requisitos.
- 10 Planificar los conflictos y su resolución.

2.5 Conclusiones

En este capítulo se ha definido un modelo de procesos para ingeniería de requisitos, se han comentado brevemente otros modelos de procesos similares, se han comparado los modelos entre si y se ha introducido el modelo de madurez de proceso de ingeniería de requisitos REAIMS.

En el resto de la tesis se estudiará cada actividad del modelo propuesto en profundidad proponiendo metodologías que cubran la mayor parte de las guías del modelo de madurez de proceso REAIMS. En las conclusiones del capítulo 6 se comentan los resultados alcanzados.

2.6 Bibliografía

- [Boehm *et al.* 1994] B. W. Boehm, P. Bose, E. Horowitz, y M.-J. Lee. Software Requirements as Negotiated Win Conditions. En *Proceedings of the First International Conference on Requirements Engineering*, 1994. Disponible en <http://sunset.usc.edu/TechRpts/Papers/NGPM-Requirements93.ps>.
- [Booch *et al.* 1999] G. Booch, J. Rumbaugh, y I. Jacobson. *The Unified Modeling Language User Guide*. Addison-Wesley, 1999.
- [Bourque *et al.* 1999] P. Bourque, R. Dupuis, y A. Abran. The Guide to the Software Engineering Body of Knowledge. *IEEE Software*, 16(6), Septiembre/Diciembre 1999.
- [Christel y Kang 1992] M. G. Christel y K. C. Kang. Issues in Requirements Elicitation. Technical Report CMU/SEI-92-TR-12, Software Engineering Institute, Carnegie Mellon University, 1992. Disponible en <http://www.sei.cmu.edu>.
- [Davis 1995a] A. M. Davis. *201 Principles of Software Development*. McGraw-Hill, 1995.
- [Davis 1995b] A. M. Davis. Tracing: A Simple Necessity Neglected. *IEEE Software*, 12(5), Septiembre 1995.
- [Durán *et al.* 1999a] A. Durán, B. Bernárdez, A. Ruiz, y M. Toro. A Requirements Elicitation Approach Based in Templates and Patterns. En *WER'99 Proceedings*, Buenos Aires, 1999.

- [Durán *et al.* 1999b] A. Durán, B. Bernárdez, M. Toro, y A. Ruiz. An Object–Oriented Model and a CASE Tool for Software Requirements Management and Documentation. En *Actas de las IV Jornadas de Trabajo Menhir*, Sedano (Burgos), 1999. Universidad de Valladolid.
- [Durán *et al.* 2000] A. Durán, B. Bernárdez, A. Ruiz Cortés, R. Corchuelo, y M. Toro. Ingeniería de Requisitos y Tecnología de Objetos. *Novática*, (143), Enero–Febrero 2000.
- [Goguen y Linde 1993] J. A. Goguen y C. Linde. Techniques for Requirements Elicitation. En *Proceedings of the First International Symposium on Requirements Engineering*, 1993. También aparece en [Thayer y Dorfman 1997]. Disponible en <http://www.cse.ucsd.edu/~goguen>.
- [Gomaa 1990] H. Gomaa. The Impact of Prototyping on Software System Engineering. En Thayer y Dorfman [1990], páginas 543–552. Aparece también en [Thayer y Dorfman 1997].
- [IEEE 1990] IEEE. IEEE Standard Glossary of Software Engineering Terminology. IEEE Standard 610.12–1990, Institute of Electrical and Electronics Engineers, 1990.
- [Jacobson *et al.* 1993] I. Jacobson, M. Christerson, P. Jonsson, y G. Övergaard. *Object–Oriented Software Engineering: A Use Case Driven Approach*. Addison–Wesley, 4ª edición, 1993.
- [Kovitz 1998] B. L. Kovitz. *Practical Software Requirements: A Manual of Content & Style*. Manning, 1998.
- [Parets-Llorca y Grünbacher 1999] J. Parets-Llorca y P. Grünbacher. Capturing, Negotiating, and Evolving System Requirements: Bridging Win Win and the UML. En *Proceedings of 25th EUROMICRO Conference*, Milan, 1999.
- [Paulk *et al.* 1993] M. C. Paulk, B. Curtis, M. B. Chrissis, y C. V. Weber. Capability Maturity Model for Software, Version 1.1. Technical Report CMU/SEI-93-TR-024, Software Engineering Institute, Carnegie Mellon University, 1993. Disponible en <http://www.sei.cmu.edu>.
- [Pohl 1994] K. Pohl. The Three Dimensions of Requirements Engineering: A Framework and its Application. *Information Systems*, 3(19), Junio 1994.

- [Pohl 1997] K. Pohl. Requirements Engineering: An Overview. *Encyclopedia of Computer Science and Technology*, 36, 1997. Disponible en <http://sunsite.informatik.rwth-aachen.de/CREWS/reports96.htm>.
- [Potts *et al.* 1994a] C. Potts, K. Takahashi, y A. Anton. Inquiry-Based Requirements Analysis. *IEEE Software*, 11(2), 1994.
- [Potts *et al.* 1994b] C. Potts, K. Takahashi, y A. Anton. Inquiry-Based Scenario Analysis of System Requirements. Informe Técnico GIT-CC-94/14, Georgia Institute of Technology, 1994. Disponible en http://www.cc.gatech.edu/computing/SW_Eng/reqts.html.
- [Raghavan *et al.* 1994] S. Raghavan, G. Zelesnik, y G. Ford. Lecture Notes on Requirements Elicitation. Educational Materials CMU/SEI-94-EM-10, Software Engineering Institute, Carnegie Mellon University, 1994. Disponible en <http://www.sei.cmu.edu>.
- [Ratcliff 1994] B. Ratcliff. *Introducing Specification Using Z: A Practical Case Study Approach*. International Series in Software Engineering. McGraw-Hill, 1994.
- [Rumbaugh *et al.* 1991] J. Rumbaugh, M. Blaha, W. Premerlani, F. Eddy, y W. Lorensen. *Object-Oriented Modeling and Design*. Prentice-Hall, 1991.
- [Sawyer *et al.* 1997] P. Sawyer, I. Sommerville, y S. Viller. Requirements Process Improvement through The Phased Introduction of Good Practice. *Software Process – Improvement and Practice*, 3(1), 1997. Disponible en <http://www.comp.lancs.ac.uk/computing/research/cseg/reaims/publications.html>.
- [Sawyer y Kontoya 1999] P. Sawyer y G. Kontoya. SWEBOK: Software Requirements Engineering Knowledge Area Description. Informe Técnico Versión 0.5, SWEBOK Project, 1999. Disponible en <http://www.swebok.org>.
- [Sommerville y Sawyer 1997] I. Sommerville y P. Sawyer. *Requirements Engineering: A Good Practice Guide*. Wiley, 1997.
- [Thayer y Dorfman 1990] R. H. Thayer y M. Dorfman, editores. *System and Software Requirements Engineering*. IEEE Computer Society Press, 1990.

[Thayer y Dorfman 1997] R. H. Thayer y M. Dorfman, editores. *Software Requirements Engineering*. IEEE Computer Society Press, 2ª edición, 1997. Es la 2ª edición de [Thayer y Dorfman 1990].

[Yourdon Inc. 1993] Yourdon Inc. *Yourdon Systems Method: Model-Driven Systems Development*. Prentice-Hall, 1993.

Capítulo 3

Elicitación de Requisitos

3.1 Introducción

En este capítulo se estudia la primera actividad del modelo de procesos de ingeniería de requisitos propuesto en el capítulo anterior: la *elicitación de requisitos*.

En este trabajo se considera la elicitación como la actividad de la ingeniería de requisitos en la que se estudia el dominio del problema y se interactúa con los clientes y usuarios para obtener y registrar información sobre sus necesidades. Estas interacciones pueden realizarse para recopilar información o para resolver conflictos que se hayan detectado en la información recopilada, avanzando de esta forma en las dimensiones de compleción y acuerdo del proceso descritas en [Pohl 1994, Pohl 1997] y comentadas en la sección 1.2.1 (ver figura 1.4, pág. 11).

La estructura del capítulo es la siguiente: en las secciones 3.2, 3.3 y 3.4 se ofrece una visión general de la actividad de elicitación con sus principales características, los principales problemas que suelen aparecer durante su realización y se describen algunas de las técnicas empleadas habitualmente.

En las siguientes secciones se exponen las aportaciones que se realizan sobre el tema. En la sección 3.5 se describe la metodología propuesta para la elicitación de requisitos de sistemas de información que se incluye como apéndice de este trabajo.

En la sección 3.6 se proponen una serie de plantillas y patrones lingüísticos pensados para ser utilizados durante las sesiones de elicitación y como forma de registrar los requisitos—C.

En la sección 3.7 se describen patrones de reutilización de requisitos—C identificados al aplicar las plantillas y patrones propuestos en la sección anterior.

En la sección 3.8 se comentan algunos trabajos relacionados y, por último, en la sección 3.9 se exponen las conclusiones del capítulo.

3.2 La elicitación de requisitos

La elicitación de requisitos no se ha considerado parte del ciclo de vida de desarrollo de software hasta hace relativamente pocos años. Se solía asumir que el cliente proporcionaba los requisitos, de forma que el ciclo de vida comenzaba siempre por el *análisis* de unos requisitos ya dados¹, ya que las actividades de elicitación y validación no se consideraban necesarias². Cuando se ha detectado que los problemas en los requisitos son uno de los principales factores de los fracasos de los proyectos software [GAO 1979, TSG 1995, ESP 1996], es cuando se le ha comenzado a dar importancia al proceso de obtención de esos requisitos³.

A nivel de investigación, la elicitación es sin duda la actividad a la que menos atención se le ha prestado en la ingeniería de requisitos. Por ejemplo, en [Christel y Kang 1992] puede leerse la siguiente cita de J. C. Leite:

"[...] Creemos que la mayoría de los investigadores evitan tratar con la elicitación de requisitos porque es un área dónde se tiene que tratar con informalidad, incompleción e inconsistencia. En su lugar, la investigación etiquetada como dedicada a los requisitos normalmente se ocupa de la especificación [...]"

Y en [Goguen y Linde 1993] puede leerse:

"[...] Algunos científicos informáticos (Computing Scientists) podrían pensar que la elicitación de requisitos es donde la ciencia termina y empieza el caos."

¹Esta es la causa por la que tradicionalmente se ha denominado a la primera fase de desarrollo como *análisis de requisitos*.

²Si se asume que el cliente proporciona los requisitos, ¿para qué va elicitar algo que ya se tiene?, ¿para qué va a validarse algo que ha sido proporcionado por la persona que lo tiene que validar?

³En la sección 1.1.3 (pág. 3) de esta memoria se comentan los resultados de [GAO 1979], [TSG 1995] y [ESP 1996].

Nuestra visión de la elicitación de requisitos es considerarla como la actividad de la ingeniería de requisitos en la que los ingenieros de requisitos *interactúan* con el resto de participantes para obtener, registrar, y si es necesario negociar, los requisitos que deberá satisfacer el sistema a desarrollar desde el punto de vista de clientes y usuarios, es decir, los requisitos-C.

Precisamente por la necesidad de esta interacción es por lo que los aspectos *sociales* son fundamentales durante esta actividad, por encima de los puramente técnicos. En palabras de J. Goguen [Goguen y Linde 1993]:

"Los problemas de la elicitación de requisitos no pueden resolverse de una forma puramente tecnológica porque el contexto social es mucho más crucial que en las fases de programación, especificación o diseño."

Dentro del modelo de procesos iterativo presentado en el capítulo 2, las actividades de elicitación pueden realizarse varias veces. En la primera iteración, la elicitación consistirá básicamente en obtener la mayor cantidad posible de información, asumiendo que lo más probable es que dicha información sea incompleta, ambigua y contenga contradicciones.

En las siguientes iteraciones, la elicitación consistirá principalmente en la resolución de conflictos encontrados en la información elicitada durante las actividades de análisis o de validación. La resolución de estos conflictos se llevará a cabo, normalmente, mediante algún tipo de *negociación* entre los participantes [Pohl 1997, Boehm *et al.* 1994, Parets-Llorca y Grünbacher 1999, Damian *et al.* 2000].

3.3 Problemas de la elicitación de requisitos

Como se justificó en el primer capítulo, la mayor parte de los problemas del desarrollo de software están relacionados con la ingeniería de requisitos, y dentro de ésta, con la elicitación de dichos requisitos.

Aunque se disponga de excelentes lenguajes de especificación de requisitos e incluso aunque se consiga que los clientes y usuarios validen una determinada especificación, si no se han elicitado los requisitos *correctos*, todo el trabajo de desarrollo terminará con un producto técnicamente correcto pero inútil, ya que no satisfará las necesidades que dieron origen a su desarrollo.

Los problemas a los que se enfrenta la elicitación de requisitos son múltiples. En [Raghavan *et al.* 1994] se establecen cinco grandes categorías de

problemas dentro de la elicitación: problemas de *articulación*, de *comunicación*, de *limitaciones cognitivas*, de *conducta humana* y *técnicos*. Nosotros adoptaremos también esta clasificación como guía para exponer las principales características de los problemas en la elicitación de requisitos.

3.3.1 Problemas de articulación

Los problemas de articulación están relacionados con la expresión de sus necesidades por parte de clientes y usuarios y la comprensión de dichas necesidades por parte de los desarrolladores. Algunos de estos problemas son los siguientes:

- Los clientes y usuarios pueden ser conscientes de sus necesidades pero no ser capaces de expresarlas apropiadamente. Es lo que en sociología se denomina el *problema de decir-hacer* y en filosofía se denomina *conocimiento tácito*: las personas saben cómo hacer muchas cosas que no saben describir [Goguen 1994].

En [Raghavan *et al.* 1994] se comenta un claro ejemplo: si una persona con hambre llega a un restaurante pero no puede decidir qué quiere comer o no puede entender el menú, decirle al camarero que tiene hambre no es una articulación apropiada de unos requisitos a los que el camarero pueda responder.

- Los clientes y usuarios pueden no ser conscientes de sus necesidades y puede que no entiendan cómo la tecnología puede ayudarles. Por ejemplo, puede que los directivos de ventas de una determinada empresa no sepan que podrían acelerar el proceso de envío de mercancías a sus clientes si sus representantes comerciales utilizarán ordenadores portátiles con teléfonos móviles para poder enviar los pedidos inmediatamente en lugar de esperar a que terminen sus desplazamientos.
- Algunos usuarios pueden no expresar sus necesidades por miedo a parecer incompetentes ante los demás o porque los desarrolladores juegan un papel excesivamente dominante en el proceso, provocando que la falta de conocimiento tecnológico de los usuarios les haga sentir en inferioridad de condiciones [Macaulay 1999].
- Los clientes pueden no llegar a tomar decisiones porque no pueden prever las consecuencias de su decisión o porque no entienden las

alternativas que se les plantea. En otras ocasiones no se toman decisiones porque no hay una sola persona que tenga una visión global, por lo que puede haber varios puntos de vista que tengan que integrarse [Kontoya y Sommerville 1996, Nuseibeh *et al.* 1994].

- Algunos desarrolladores no escuchan apropiadamente a los clientes y usuarios, bien porque creen haber entendido sus necesidades rápidamente, bien porque se dedican a pensar inmediatamente sobre aspectos de implementación y no se ponen en el lugar de clientes y usuarios.

3.3.2 Problemas de comunicación

Algunas de las dificultades en la comunicación entre clientes, usuarios y desarrolladores son las siguientes:

- Los clientes y usuarios y los desarrolladores tienen culturas y vocabularios diferentes, con la posibilidad de que los mismos términos tengan significados distintos en los distintos vocabularios⁴, o que su significado se vea enormemente afectado por el contexto, ya que en estos momentos del desarrollo la información está fuertemente *contextualizada* [Goguen 1994].
- No sólo la cultura y el vocabulario son distintos, las preocupaciones sobre el sistema a desarrollar también suelen serlo. Mientras los clientes y usuarios suelen preocuparse por aspectos de alto nivel como facilidad de uso o fiabilidad, los desarrolladores suelen preocuparse por aspectos de bajo nivel como utilización de recursos, algoritmos, etc.

Es importante no olvidar que el principal interés de los clientes no es un sistema software en sí mismo, sino los efectos positivos resultantes de la introducción del sistema en su organización. En palabras de A. M. Davis [Davis 1995, pág. 15]:

"Nunca debe perderse de vista porqué se desarrolla el software: para satisfacer necesidades reales, para resolver problemas reales. La única forma de resolver las necesidades reales es comunicarse con aquellos que tienen dichas necesidades. El cliente o usuario es la persona más importante involucrada en el proyecto."

⁴Por ejemplo, se le puede preguntar a un abogado qué significan para él o ella los términos *instancia* o *requerimiento*.

- El medio de comunicación que se utilice debe ser entendible por todos los participantes. Se suele utilizar lenguaje natural porque es el único medio de comunicación común a todos los participantes, a pesar de su inherente ambigüedad. La utilización de otro tipo de técnicas como diagramas o lenguajes artificiales puede presentar problemas de comprensión⁵.

En este trabajo seguiremos las propuestas actuales, por ejemplo las de [Sawyer y Kontoya 1999] o [Davis 1995], en las que se recomienda usar distintos lenguajes según la audiencia. En concreto, lenguaje natural para la audiencia de clientes y usuarios (requisitos-C) y modelos conceptuales para la audiencia de los desarrolladores (requisitos-D).

- La comunicación puede verse afectada también por sus aspectos puramente sociales. El ingeniero de requisitos debe ser capaz de comunicarse y tratar con todo tipo de personas y ser capaz de manejar conflictos personales y políticos [Goguen 1994, Macaulay 1998, Macaulay 1999]. En [Damian *et al.* 2000] se exponen los resultados de utilizar reuniones mediante vídeo conferencia o mediante discusiones por ordenador para evitar los problemas de posibles enfrentamientos cara a cara entre los participantes.

3.3.3 Problemas de limitaciones cognitivas

Los problemas de las limitaciones cognitivas del ser humano aparecen, como no podría ser de otra manera, en las actividades de elicitación de requisitos. Algunos de estos problemas son los siguientes:

- El ingeniero de requisitos debe tener un conocimiento adecuado del dominio del problema y no hacer suposiciones sobre ello, al igual que los clientes y usuarios no deben hacer suposiciones sobre aspectos tecnológicos. Es lo que Goguen denomina el abuso de la *introspección* [Goguen 1994]. Muchas veces las ideas preconcebidas sobre una posible solución afectan a la forma en que se establece la definición del problema.

⁵Los principios 54 a 56 de [Davis 1995] ofrecen una interesante discusión sobre la necesidad de la utilización del lenguaje natural en las especificaciones de requisitos para mantener la comunicabilidad entre los participantes en el proceso.

En el caso del ingeniero de requisitos, es fundamental que conozca el dominio del problema y el entorno tanto organizacional como operacional en el que deberá funcionar el sistema a desarrollar [Sawyer y Kontoya 1999].

Esta necesidad de conocer el entorno es la que ha provocado que en la mayor parte de las metodologías y normas de desarrollo ([DoD 1994, MAP 1995, Mazza *et al.* 1994] por ejemplo) se haya prestado gran atención al estudio del sistema actual como paso previo a la definición de los requisitos del sistema a desarrollar.

- Cuando los problemas son grandes y complejos, algunas personas tienden a hacer simplificaciones no válidas, a ignorar las partes más complejas o a centrarse únicamente en las aspectos que más conocen o que más les afectan. El ingeniero de requisitos debe ser capaz de manejar problemas complejos y asegurarse de que todos los temas importantes son tratados.

3.3.4 Problemas de conducta humana

La naturaleza social de la elicitación de requisitos provoca que surjan problemas de conducta humana, algunos de los cuales son los siguientes:

- Puede haber conflictos y ambigüedades en los roles que cada persona debe jugar en el proceso de elicitación. Dentro del grupo de clientes y usuarios, algunos pueden pensar que, aunque conozcan ciertas necesidades o ciertos aspectos importantes, es responsabilidad de otros participantes más afectados el hacerlas explícitas, con lo que el resultado final es que nadie dice nada.

Otra situación similar puede producirse si algunos clientes y usuarios piensan que los desarrolladores les harán todas las preguntas necesarias sobre el dominio del problema y los desarrolladores piensan que los clientes y usuarios les proporcionarán toda la información necesaria sin necesidad de preguntar por su parte, con lo que pueden quedar aspectos sin tratar.

- La suposición o el temor a que el sistema a desarrollar cambie su forma de trabajar o incluso ponga en peligro su puesto de trabajo, puede provocar que algunos usuarios retengan información o incluso saboteen el desarrollo, por ejemplo proporcionando información falsa.

Como ya se comentó en la sección 2.2.3 (pág. 43), es fundamental que el ingeniero de requisitos conozca el entorno organizacional para poder detectar y tratar este tipo de problemas [Sawyer y Kontoya 1999].

3.3.5 Problemas técnicos

Otros problemas de la elicitación pueden considerarse como *técnicos*. Son los siguientes:

- El software tiene que resolver problemas cada vez más complejos, por lo que sus requisitos son también cada vez más complejos y contemplan detalles cada vez más específicos del dominio del problema.
- Los requisitos cambian en el tiempo, ya que ha medida que los clientes y usuarios van conociendo sus propias necesidades y las posibilidades que les ofrece la tecnología puede surgir la necesidad de reconsiderar decisiones anteriores o descubrir nuevas necesidades⁶. Esta es una de las principales causas de la naturaleza iterativa de la ingeniería de requisitos.
- Otra fuente de cambios es el hecho de que el hardware y el software cambian rápidamente, haciendo asequibles requisitos que antes eran inabordables por su complejidad o por su coste.
- En algunos sistemas puede haber muchas fuentes de requisitos, por lo que a veces no basta con consultar con los clientes y usuarios, sino que también es necesario hacerlo con técnicos, personal de mantenimiento, consultar normativas, estándares, etc.
- Es necesario tener también en cuenta que los sistemas novedosos requieren un esfuerzo mucho mayor de elicitación, y que las fuentes de información pueden ser muy distintas dependiendo de la naturaleza del sistema a desarrollar.

⁶El principio 15 de [Davis 1995] (*Cuanto más se ve, más se necesita*) ilustra los aspectos de descubrimiento de nuevas necesidades.

3.4 Técnicas de elicitación de requisitos

Las técnicas más habituales en la elicitación de requisitos son las entrevistas, el *Joint Application Development* (JAD) o Desarrollo Conjunto de Aplicaciones, el *brainstorming* o tormenta de ideas y la utilización de *escenarios* [Weidenhaput *et al.* 1998, Rolland *et al.* 1998], más conocidos como *casos de uso* [Jacobson *et al.* 1993, Booch *et al.* 1999].

A estas técnicas, que se describen en los siguientes apartados, se las suele apoyar con otras técnicas complementarias como la observación *in situ*, el estudio de documentación, los cuestionarios, la *inmersión* en el negocio del cliente [Goguen y Linde 1993] o haciendo que los ingenieros de requisitos sean *aprendices* del cliente [Beyer y Holtzblatt 1995].

Aunque la construcción de prototipos puede considerarse también como una técnica de elicitación, en esta tesis se le prestará especial atención como herramienta de validación de requisitos, sin dejar de reconocer que durante la validación mediante casos de uso y prototipos pueden elicitar nuevos requisitos (ver capítulo 5).

3.4.1 Entrevistas

Las entrevistas son la técnica de elicitación más utilizada, y de hecho son prácticamente inevitables en cualquier desarrollo ya que son una de las formas de comunicación más *naturales* entre personas.

En las entrevistas se pueden identificar tres fases: *preparación*, *realización* y *análisis* [Piattini *et al.* 1996].

3.4.1.1 Preparación de entrevistas

Las entrevistas no deben improvisarse, por lo que conviene realizar las siguiente tareas previas:

- **Estudiar el dominio del problema:** conocer las categorías y conceptos de la comunidad de clientes y usuarios es fundamental para poder entender las necesidades de dicha comunidad y su forma de expresarlas [Goguen y Linde 1993], y para generar en los clientes y usuarios la confianza de que el ingeniero de requisitos *entiende* sus problemas.

Para conocer el dominio del problema se puede recurrir a técnicas de estudio de documentación, a bibliografía sobre el tema, documentación de proyectos similares realizados anteriormente, la *inmersión* dentro de la organización para la que se va a desarrollar [Goguen y Linde 1993] o a periodos de *aprendizaje* por partes de los ingenieros de requisitos [Beyer y Holtzblatt 1995].

- **Seleccionar a las personas a las que se va a entrevistar:** se debe minimizar el número de entrevistas a realizar, por lo que es fundamental seleccionar a las personas a entrevistar. Normalmente se comienza por los directivos, que pueden ofrecer una visión global, y se continúa con los futuros usuarios, que pueden aportar información más detallada, y con el personal técnico, que aporta detalles sobre el entorno operacional de la organización.

Tal como se recomienda en [Piattini *et al.* 1996], conviene también estudiar el perfil de los entrevistados, buscando puntos en común con el entrevistador que ayuden a *romper el hielo*.

- **Determinar el objetivo y contenido de las entrevistas:** para minimizar el tiempo de la entrevista es fundamental fijar el objetivo que se pretende alcanzar y determinar previamente su contenido.

Previamente a su realización, se pueden enviar cuestionarios que los futuros entrevistados deben rellenar y devolver, y un pequeño documento de introducción al proyecto de desarrollo, de forma que el entrevistado conozca los temas que se van a tratar y el entrevistador recoja información para preparar la entrevista.

Es importante que los cuestionarios, si se usan, se preparen cuidadosamente teniendo en cuenta quién los va a responder y no incluir conceptos que se asuman conocidos cuando puedan no serlo⁷.

- **Planificar las entrevistas:** la fecha, hora, lugar y duración de las entrevistas deben fijarse teniendo en cuenta siempre la agenda del entrevistado.

En general, se deben buscar sitios agradables donde no se produzcan interrupciones y que resulten *naturales* a los entrevistados, tal como se describe en [Goguen y Linde 1993].

⁷En [Goguen y Linde 1993] aparecen dos ejemplos de preguntas de cuestionario cuya respuesta no se previó de antemano por parte del autor del cuestionario dando lugar a respuestas, cuanto menos, curiosas.

3.4.1.2 Realización de entrevistas

Dentro de la realización de las entrevistas se distinguen tres etapas, tal como se expone en [Piattini *et al.* 1996]:

- 1 **Apertura:** el entrevistador debe presentarse e informar al entrevistado sobre la razón de la entrevista, qué se espera conseguir, cómo se utilizará la información, la mecánica de las preguntas, etc.

Si se va a utilizar algún tipo de notación gráfica o matemática que el entrevistado no conozca debe explicarse antes de utilizarse. Es fundamental causar buena impresión en los primeros minutos.

- 2 **Desarrollo:** la entrevista en si no debería durar más de dos horas, distribuyendo el tiempo en un 20% para el entrevistador y un 80% para el entrevistado.

Se deben evitar los monólogos y mantener el control por parte del entrevistador, contemplando la posibilidad de que una tercera persona tome notas durante la entrevista o grabar la entrevista en cinta de vídeo o audio, siempre que el entrevistado esté de acuerdo [Robertson y Robertson 1999].

Durante esta fase se pueden emplear distintas técnicas:

- **Preguntas abiertas:** también denominadas *de libre contexto* [Gause y Weinberg 1989], estas preguntas no pueden responderse con un "sí" o un "no", permiten una mayor comunicación y evitan la sensación de interrogatorio. Por ejemplo, "¿Qué se hace para registrar un pedido?", "Dígame qué se debe hacer cuando un cliente pide una factura" o "¿Cómo se rellena un albarán?".

Estas preguntas se suelen utilizar al comienzo de la entrevista, pasando posteriormente a preguntas más concretas.

En general, se debe evitar la tendencia a anticipar una respuesta a las preguntas que se formulan [Raghavan *et al.* 1994]. En [Gause y Weinberg 1989, cap. 6] se exponen interesantes ejemplos de este tipo de preguntas y consejos para su utilización.

Una posibilidad es utilizar las plantillas descritas en la sección 3.6 como mecanismos tanto de obtención de información, ya que su estructura indica la información a buscar, como de registro de las respuestas a este tipo de preguntas.

- **Utilizar palabras apropiadas:** se deben evitar tecnicismos que no conozca el entrevistado y palabras o frases que puedan perturbar emocionalmente la comunicación [Goleman 1996, Goleman 1999].

- **Mostrar interés en todo momento:** es fundamental cuidar la *comunicación no verbal* [Davis 1985] durante la entrevista: tono de voz, movimiento, expresión facial, etc.

Por ejemplo, para animar a alguien a hablar puede asentirse con la cabeza, decir "*ya entiendo*", "*sí*", repetir algunas respuestas dadas, hacer pausas, poner una postura de atención, etc. Debe evitarse bostezar, reclinarse en el sillón, mirar hacia otro lado, etc.

- 3 Terminación:** al terminar la entrevista se debe recapitular para confirmar que no ha habido confusiones en la información recogida, agradecer al entrevistado su colaboración y citarle para una nueva entrevista si fuera necesario, dejando siempre abierta la posibilidad de volver a contactar para aclarar dudas que surjan al estudiar la información o al contrastarla con otros entrevistados.

3.4.1.3 Análisis de las entrevistas

Una vez realizada la entrevista es necesario leer las notas tomadas, pasarlas a limpio, reorganizar la información, contrastarla con otras entrevistas o fuentes de información, etc.

Una vez elaborada la información, se puede enviar al entrevistado para confirmar los contenidos. También es importante evaluar la propia entrevista para determinar los aspectos mejorables.

3.4.2 Joint Application Development

La técnica denominada *JAD (Joint Application Development, Desarrollo Conjunto de Aplicaciones)*, desarrollada por IBM en 1977, es una alternativa a las entrevistas individuales que se desarrolla a lo largo de un conjunto de reuniones en grupo durante un periodo de 2 a 4 días. En estas reuniones se ayuda a los clientes y usuarios a formular problemas y explorar posibles soluciones, involucrándolos y haciéndolos sentirse partícipes del desarrollo.

Esta técnica se base en cuatro principios [Raghavan *et al.* 1994]: dinámica de grupo, el uso de ayudas visuales para mejorar la comunicación (diagramas, transparencias, multimedia, herramientas CASE, etc.), mantener un proceso organizado y racional y una filosofía de documentación *WYSIWYG (What You See Is What You Get, lo que se ve es lo que se obtiene)*, por

la que durante las reuniones se trabaja directamente sobre los documentos a generar.

El JAD tiene dos grandes pasos, el *JAD/Plan* cuyo objetivo es elicitar y especificar requisitos, y el *JAD/Design*, en el que se aborda el diseño del software. En este documento sólo se verá con detalle el primero de ellos.

Debido a las necesidades de organización que requiere y a que no suele adaptarse bien a los horarios de trabajo de los clientes y usuarios, esta técnica no suele emplearse con frecuencia, aunque cuando se aplica suele tener buenos resultados, especialmente para elicitar requisitos en el campo de los sistemas de información [Raghavan *et al.* 1994].

En comparación con las entrevistas individuales, presenta las siguientes ventajas:

- Ahorra tiempo al evitar que las opiniones de los clientes se contrasten por separado.
- Todo el grupo, incluyendo los clientes y los futuros usuarios, revisa la documentación generada, no sólo los ingenieros de requisitos.
- Implica más a los clientes y usuarios en el desarrollo.

3.4.2.1 Participantes del JAD

Tal como se expone en [Raghavan *et al.* 1994], se pueden distinguir seis clases de participantes o roles en el JAD:

- **Jefe del JAD:** es el responsable de todo el proceso y asume el control durante las reuniones. Debe tener dotes de comunicación y liderazgo. Algunas habilidades importantes que debe tener son: entender y promover la dinámica de grupo, iniciar y centrar discusiones, reconocer cuándo la reunión se está desviando del tema y reconducirla, manejar las distintas personalidades y formas de ser de los participantes, evitar que decaiga la reunión aunque sea larga y difícil, etc.
- **Analista:** es el responsable de la producción de los documentos que se deben generar durante las sesiones JAD. Debe tener la habilidad de organizar bien las ideas y expresarlas claramente por escrito. En el caso de que se utilizan herramientas software durante las sesiones, debe ser capaz de manejarlas eficientemente.

- **Patrocinador ejecutivo:** es el que tiene la decisión final de que se lleve a cabo el desarrollo. Debe proporcionar a los demás participantes información sobre la necesidad del nuevo sistema y los beneficios que se espera obtener de él.
- **Representantes de los usuarios:** durante el *JAD/Plan*, suelen ser directivos con una visión global del sistema. Durante el *JAD/Design* suelen incorporarse futuros usuarios finales.
- **Representantes de sistemas de información:** son personas expertos en sistemas de información que deben ayudar a los usuarios a comprender qué es o no factible con la tecnología actual y el esfuerzo que implica.
- **Especialistas:** son personas que pueden proporcionar información detallada sobre aspectos muy concretos, tanto del punto de vista de los usuarios porque conocen muy bien el funcionamiento de una parte de la organización, como desde el punto de vista de los desarrolladores porque conocen perfectamente ciertos aspectos técnicos de la instalación hardware de la organización.

3.4.2.2 Fases del JAD

Dentro de la técnica del JAD se distinguen tres fases [Raghavan *et al.* 1994]:

- 1 **Adaptación:** es responsabilidad del jefe del JAD, ayudado por uno o dos analistas, adaptar la técnica del JAD para cada proyecto. La adaptación debe comenzar por definir el proyecto a alto nivel, para lo cual pueden ser necesarias entrevistas previas con algunos clientes y usuarios. También suele ser necesario recabar información sobre la organización para familiarizarse con el dominio del problema, por ejemplo utilizando técnicas complementarias como el estudio de documentación o la observación *in situ*.

Una vez obtenida una primera idea de los objetivos del proyecto, es necesario seleccionar a los participantes, citarles para las reuniones y proporcionarles una lista con los temas que se van a tratar en las reuniones para que las puedan preparar.

El jefe del JAD debe decidir la duración y el número de sesiones a celebrar, definir el formato de la documentación sobre la que se trabajará y preparar transparencias introductorias y todo el material audiovisual que considere oportuno.

2 Celebración de las sesiones JAD: durante las sesiones, los participantes exponen sus ideas y se discuten, analizan y refinan hasta alcanzar un acuerdo. Los pasos que se recomienda seguir para este proceso son los siguientes:

2.1 Presentación: se presenta y se da la bienvenida a todos los participantes por parte del patrocinador ejecutivo y del jefe del JAD. El patrocinador ejecutivo expone brevemente las necesidades que han llevado al desarrollo y los beneficios que se esperan obtener. El jefe del JAD explica la mecánica de las sesiones y la planificación prevista.

2.2 Definir objetivos y requisitos: el jefe del JAD promueve la discusión para elicitar los objetivos o requisitos de alto nivel mediante preguntas como: "*¿Por qué se construye el sistema?*", "*¿Qué beneficios se esperan del nuevo sistema?*", "*¿Cómo puede beneficiar a la organización en el futuro?*", "*¿Qué restricciones de recursos disponibles, normas o leyes afectan al proyecto?*", "*¿Es importante la seguridad de los datos?*", ...

A medida que se van elicitando requisitos, el analista los escribe en transparencias o en algún otro medio que permita que permanezcan visibles durante la discusión. Una posibilidad es utilizar para ello las plantillas descritas en la sección 3.6.

2.3 Delimitar el ámbito del sistema: una vez obtenido un número importante de requisitos, es necesario organizarlos y llegar a un acuerdo sobre el ámbito del nuevo sistema.

En el caso de los sistemas de información, es útil identificar a los usuarios potenciales (*actores*) y determinar qué tareas les ayudará a realizar (*casos de uso*).

2.4 Documentar temas abiertos: aquellas cuestiones que hayan surgido durante la sesión que no se han podido resolver, deben documentarse para las siguientes sesiones y ser asignadas a una persona responsable de su solución para una fecha determinada, para lo cual puede utilizarse la plantilla de conflictos descrita en la sección 3.6.6.

2.5 Concluir la sesión: el jefe del JAD concluye la sesión revisando con los demás participantes la información elicitada y las decisiones tomadas. Se da la oportunidad a todos los participantes de expresar cualquier consideración adicional, fomentando por parte del jefe del JAD el sentimiento de propiedad y compromiso de todos los participantes sobre los requisitos elicitados.

3 Conclusión: una vez terminadas las sesiones es necesario transformar las transparencias, notas y demás documentación generada en documentos formales. Se distinguen tres pasos:

3.1 Completar la documentación: los analistas recopilan la documentación generada durante las sesiones en documentos conformes a las normas o estándares vigentes en la organización para la que se desarrolla el proyecto.

3.2 Revisar la documentación: la documentación generada se envía a todos los participantes para que la comenten. Si los comentarios son lo suficientemente importantes, se convoca otra reunión para discutirlos.

3.3 Validar la documentación: una vez revisados todos los comentarios, el jefe del JAD envía el documento al patrocinador ejecutivo para su aprobación. Una vez aprobado el documento se envían copias definitivas a cada uno de los participantes.

3.4.3 Brainstorming

El *brainstorming* o tormenta de ideas es una técnica de reuniones en grupo cuyo objetivo es la generación de ideas en un ambiente libre de críticas o juicios [Gause y Weinberg 1989, Raghavan *et al.* 1994]. Las sesiones de brainstorming suelen estar formadas por un número de cuatro a diez participantes, uno de los cuales es el *jefe* de la sesión, encargado más de comenzar la sesión que de controlarla.

Como técnica de elicitación de requisitos, el brainstorming puede ayudar a generar una gran variedad de vistas del problema y a formularlo de diferentes formas, sobre todo al comienzo del proceso de elicitación, cuando los requisitos son todavía muy difusos.

Frente al JAD, el brainstorming tiene la ventaja de que es muy fácil de aprender y requiere poca organización, de hecho, hay propuestas de realización de brainstorming por vídeo-conferencia a través de Internet [Raghavan *et al.* 1994]. Por otro lado, al ser un proceso poco estructurado, puede no producir resultados con la misma calidad o nivel de detalle que otras técnicas.

3.4.3.1 Fases del brainstorming

En el brainstorming se distinguen las siguientes fases [Raghavan *et al.* 1994]:

- 1 **Preparación:** la preparación para una sesión de brainstorming requiere que se seleccione a los participantes y al jefe de la sesión, citarlos y preparar la sala donde se llevará a cabo la sesión. Los participantes en una sesión de brainstorming para elicitación de requisitos son normalmente clientes, usuarios, ingenieros de requisitos, desarrolladores y, si es necesario, algún experto en temas relevantes para el proyecto.
- 2 **Generación:** el jefe abre la sesión exponiendo un enunciado general del problema a tratar, que hace de *semilla* para que se vayan generando ideas. Los participantes aportan libremente nuevas ideas sobre el problema *semilla*, bien por un orden establecido por el jefe de la sesión, bien aleatoriamente. El jefe es siempre el responsable de dar la palabra a un participante. Este proceso continúa hasta que el jefe decide parar, bien porque no se están generando suficientes ideas, en cuyo caso la reunión se pospone, bien porque el número de ideas sea suficiente para pasar a la siguiente fase. Durante esta fase se deben observar las siguientes reglas:
 - Se prohíbe la crítica de ideas, de forma que los participantes se sientan libres de formular cualquier idea.
 - Se fomentan las ideas más *avanzadas*, que aunque no sean factibles, estimulan a los demás participantes a explorar nuevas soluciones más creativas.
 - Se debe generar un gran número de ideas, ya que cuantas más ideas se presenten más probable será que se generen mejores ideas.
 - Se debe alentar a los participantes a combinar o completar las ideas de otros participantes. Para ello, es necesario, al igual que en la técnica del JAD, que todas las ideas generadas estén visibles para todos los participantes en todo momento.

Una posibilidad es utilizar como semilla objetivos del sistema e ir identificando requisitos. Si estos requisitos se recogen en las plantillas propuestas en la sección 3.6, pueden utilizarse dichas plantillas

para que los participantes tengan visibles las ideas que se van generando.

3 Consolidación: en esta fase se deben organizar y evaluar las ideas generadas durante la fase anterior. Se suelen seguir tres pasos:

3.1 Revisar ideas: se revisan las ideas generadas para clarificarlas. Es habitual identificar ideas similares, en cuyo caso se unifican en un solo enunciado.

3.2 Descartar ideas: aquellas ideas que los participantes consideren excesivamente *avanzadas* se descartan.

3.3 Priorizar ideas: se priorizan las ideas restantes, identificando las absolutamente esenciales, las que estarían bien pero que no son esenciales y las que podrían ser apropiadas para una próxima versión del sistema a desarrollar.

4 Documentación: después de la sesión, el jefe produce la documentación oportuna conteniendo las ideas priorizadas y comentarios generados durante la consolidación.

3.4.4 Casos de uso

Los casos de uso son una técnica para la especificación de requisitos funcionales propuesta inicialmente en [Jacobson *et al.* 1993] y que actualmente forma parte de la propuesta de UML [Booch *et al.* 1999].

Un caso de uso es la descripción de una secuencia de interacciones entre el sistema y uno o más *actores* en la que se considera al sistema como una caja negra y en la que la que los actores obtienen resultados observables.

Los actores son personas u otros sistemas que interactúan con el sistema cuyos requisitos se están describiendo [Schneider y Winters 1998].

Los casos de uso presentan ciertas ventajas sobre la descripción meramente textual de los requisitos funcionales [Firesmith 1997], ya que facilitan la elicitación de requisitos y son fácilmente comprensibles por los clientes y usuarios. Además, pueden servir de base a las pruebas del sistema y a la documentación para los usuarios [Weidenhaput *et al.* 1998].

A pesar de ser una técnica ampliamente aceptada, existen múltiples propuestas para su utilización concreta [Cockburn 1997]. En este trabajo

se propone la utilización de los casos de uso como técnica tanto de elicitación como de especificación de los requisitos funcionales del sistema. Para la descripción concreta de los casos de uso se proponen plantillas, en las que las interacciones se numeran siguiendo las propuestas de [Cockburn 1997], [Schneider y Winters 1998] y [Coleman 1998] y se describen usando lenguaje natural en forma de patrones lingüísticos (ver sección 3.6.4, pág. 90).

3.4.4.1 Diagramas de casos de uso

Los casos de uso tienen una representación gráfica en los denominados *diagramas de casos de uso* [Booch *et al.* 1999]. En estos diagramas, los actores se representan en forma de pequeños monigotes y los casos de uso se representan por elipses contenidas dentro de un rectángulo que representa al sistema. La participación de los actores en los casos de uso se indica por una flecha entre el actor y el caso de uso que apunta en la dirección en la que fluye la información. Un ejemplo de este tipo de diagramas puede verse en la figura 3.1.

Los diagramas de casos de uso sirven para proporcionar una visión global del conjunto de casos de uso de un sistema así como de los actores y los casos de uso en los que éstos intervienen. Las interacciones concretas entre los actores y el sistema no se muestran en este tipo de diagramas.

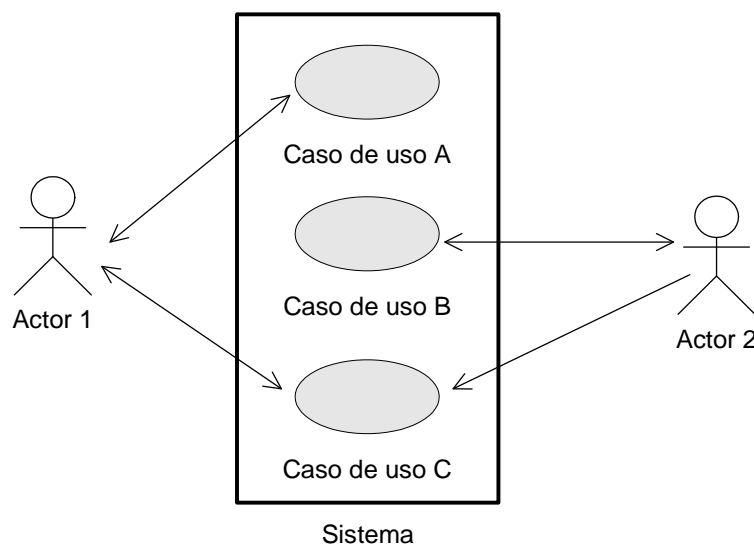


Figura 3.1: Diagrama de casos de uso

3.4.4.2 Relaciones entre casos de uso

A veces conviene establecer relaciones entre distintos casos de uso para simplificar su descripción. Las dos relaciones posibles y sus semánticas según [Booch *et al.* 1999] son las siguientes, cuya representación gráfica puede verse en el ejemplo de la figura 3.2.

- **includes:** se dice que un caso de uso *A incluye* el caso de uso *B*, cuando *B* es una parte del caso de uso *A*, es decir, la secuencia de interacciones de *B* forma parte de la secuencia de interacciones de *A*.

El caso de uso *B* se realiza siempre dentro del caso de uso *A*. Además, siempre que ocurre *A* ocurre también *B*, por lo que se dice que *B* es un caso de uso *abstracto* [Jacobson *et al.* 1997, Firesmith 1997]. Un caso de uso es abstracto si no puede ser realizado por sí mismo, por lo que sólo tiene significado cuando se utiliza para describir alguna funcionalidad que es común a otros casos de uso. Por otra parte, un caso de uso será *concreto* si puede ser iniciado por un actor y realizado por sí mismo.

Se suele utilizar esta relación cuando se detectan subsecuencias de interacciones comunes a varios casos de uso. Dichas subsecuencias comunes se sacan "factor común" de los casos de uso que las contienen y se les da forma de casos de uso que son *incluidos* por los casos de uso de los que se han "extraído". De esta forma se evita repetir las mismas subsecuencias de interacciones una y otra vez en varios casos de uso.

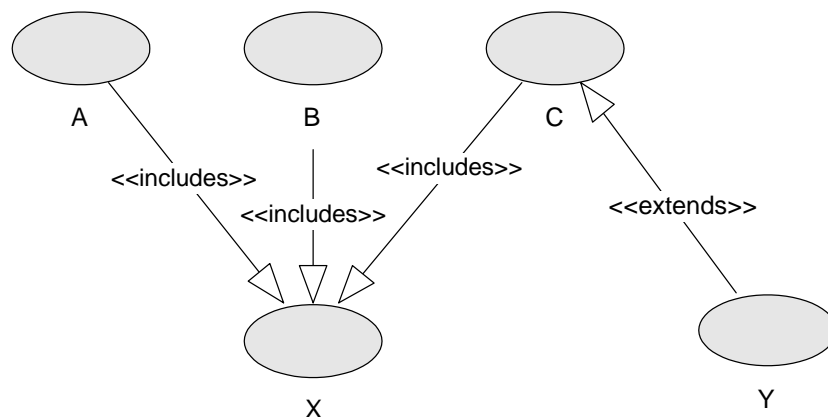


Figura 3.2: Representación gráfica de las relaciones *includes* y *extends*

- **extends:** un caso de uso *A* *extiende* a otro caso de uso *B* cuando *A* es una subsecuencia de interacciones de *B* que ocurre en una determinada circunstancia.

En cierta forma, *A* completa la funcionalidad de *B*. El caso de uso *A* puede realizarse o no cuando se realiza el caso de uso *B*, según se den las circunstancias. Por otro lado, el caso de uso *A* puede ser un caso de uso abstracto o concreto, en cuyo caso puede ocurrir sin necesidad de que ocurra el caso de uso *B*.

3.4.4.3 Organización de casos de uso

En la mayoría de sistemas, el número de casos de uso es lo suficientemente elevado como para que sea oportuno organizarlos de alguna forma en lugar de tener una *lista plana* por la que no es fácil navegar.

Una posible forma de organizar los casos de uso es recurrir a los *paquetes* descritos en la propuesta de UML [Booch *et al.* 1999]. De esta forma, los casos de uso pueden organizarse en niveles, facilitando así su comprensión. Cada paquete contiene a otros paquetes o a varios casos de uso.

En el caso de que los casos de uso se agrupen por criterios funcionales, los paquetes que los agrupan pueden estereotiparse como *subsistemas* [Schneider y Winters 1998], tal como puede verse en el ejemplo de la figura 3.3.

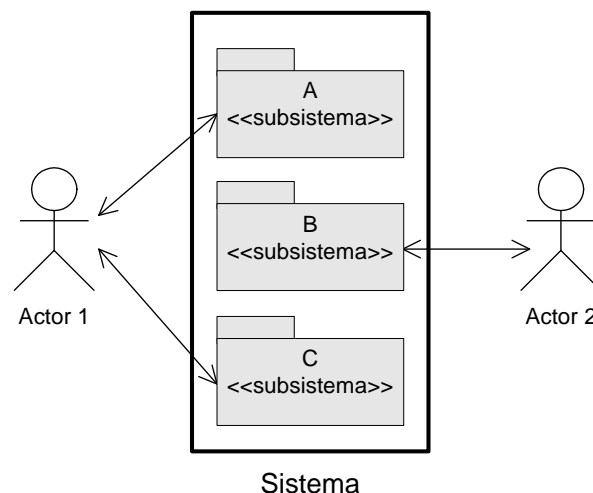


Figura 3.3: Representación gráfica de los paquetes de casos de uso

3.5 Propuesta metodológica para la elicitación de requisitos

La propuesta metodológica para la elicitación de requisitos que se propone en este trabajo, y que se incluye en el apéndice B, propone una serie de tareas a realizar y productos a obtener (tanto internos como externos o *entregables*) durante la realización de la actividad de elicitación de requisitos del modelo de procesos descrito en el capítulo 2. Esta metodología es una evolución de la presentada inicialmente en [Durán *et al.* 1998].

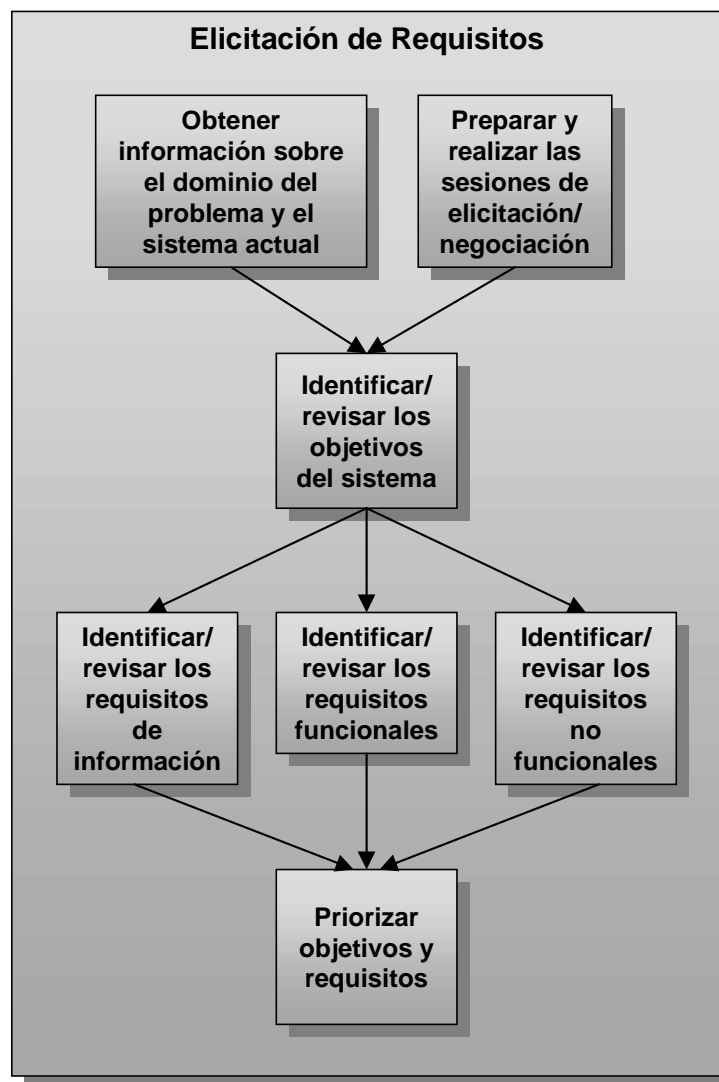


Figura 3.4: Tareas de elicitación de requisitos

En la propuesta se contemplan las siete tareas que pueden verse en la figura 3.4, en la que se propone un posible orden de realización orientativo. En las siguientes secciones se describen cada una de ellas.

3.5.1 Tarea 1: Obtener información sobre el dominio del problema y el sistema actual

El objetivo principal de esta tarea es conocer el dominio del problema lo mejor posible. Las razones para ello son varias. En primer lugar, como ya se ha comentado en la sección 3.3, el ingeniero de requisitos debe conocer el lenguaje de clientes y usuarios para poder comunicarse con ellos. Cada dominio de problemas posee un vocabulario propio que es necesario conocer [Rombach 1990, Brackett 1990].

En segundo lugar, el ingeniero de requisitos debe evitar utilizar sus propios esquemas y categorías mentales a la hora de obtener la información, ya que esto puede dificultar la comunicación [Goguen 1994]. Debe aprender a pensar en los términos en los que lo hacen clientes y usuarios. Una herramienta muy útil para ello es realizar un estudio del sistema actual, en caso de que exista.

En tercer lugar, mantener sesiones de elicitación sin conocer el dominio del problema puede provocar que los malentendidos o las preguntas continuas sobre el significado de los términos empleados por clientes y usuarios hagan que la confianza hacia el equipo de ingeniería de requisitos se vea deteriorada enormemente, provocando cierta reticencia a participar e involucrarse en el proyecto⁸ [Blaha y Premerlani 1998, pág. 118].

Los productos resultantes de la realización de esta tarea son productos internos resultado de la búsqueda de información sobre el dominio del problema y la situación del sistema actual: modelos del sistema actual, documentación de la organización, resultados de entrevistas iniciales, resultados de cuestionarios exploratorios, documentación de desarrollos previos sobre el mismo dominio de problemas, información proveniente de expertos, etc.

Las técnicas que pueden utilizarse para realizar esta tarea son varias: recopilación de documentación, entrevistas, reuniones en grupo, cuestionarios, *inmersión* en el negocio del cliente [Goguen y Linde 1993], *aprendizaje* del negocio [Beyer y Holtzblatt 1995], modelado del sistema actual [Laguna *et al.* 1999, García *et al.* 2000], etc.

⁸Que es la primera causa de fracaso de los proyectos software [TSG 1995].

3.5.2 Tarea 2: Preparar y realizar las sesiones de elicitación/negociación

El objetivo de esta tarea es conocer las necesidades de clientes y usuarios y resolver los conflictos identificados en las actividades de análisis de iteraciones previas. Es la tarea más crítica, ya que en ella es donde existe una mayor interacción personal entre el equipo de ingeniería de requisitos y los clientes y usuarios, por lo que una adecuada selección de los participantes es crucial.

Los productos resultantes de esta tarea, al igual que en la tarea anterior, son también internos y suelen componerse de notas tomadas durante las sesiones, transcripciones o actas de las sesiones, grabaciones de audio o vídeo [Robertson y Robertson 1999], etc. Si se han resuelto conflictos durante las sesiones, se pueden considerar como productos dichos conflictos resueltos junto con la probable necesidad de cambios en los requisitos-C. En el caso de que se usen las plantillas para elicitación de requisitos propuestas en este trabajo, podría obtenerse una primera versión o un borrador de los requisitos-C.

Las técnicas que pueden emplearse para la realización de esta tarea son las técnicas de elicitación que se describieron en la sección 3.4, técnicas de negociación como las propuestas en [Boehm *et al.* 1994], [Potts *et al.* 1994] o [Parets-Llorca y Grünbacher 1999], y las plantillas para elicitación de requisitos que se describen en la sección 3.6.

3.5.3 Tarea 3: Identificar/revisar los objetivos del sistema

El objetivo de esta tarea es conocer *por qué* se acomete el desarrollo, y por lo tanto conocer qué objetivos se esperan alcanzar mediante el sistema software a desarrollar. En la primera iteración se realizará una primera identificación de los objetivos. En las iteraciones posteriores puede que sea necesario revisarlos si se han identificado conflictos que les afecten.

Esta información puede que haya sido dada previamente al comienzo del desarrollo⁹, puede que se haya manifestado de forma explícita durante las sesiones de elicitación o puede que haya que deducirla de la información recopilada durante dichas sesiones.

La idea básica es ir obteniendo los requisitos como un refinamiento de los objetivos, de forma que la existencia de un requisito esté siempre

⁹Por ejemplo, por la existencia de un *Plan de Sistemas de Información* [MAP 1995].

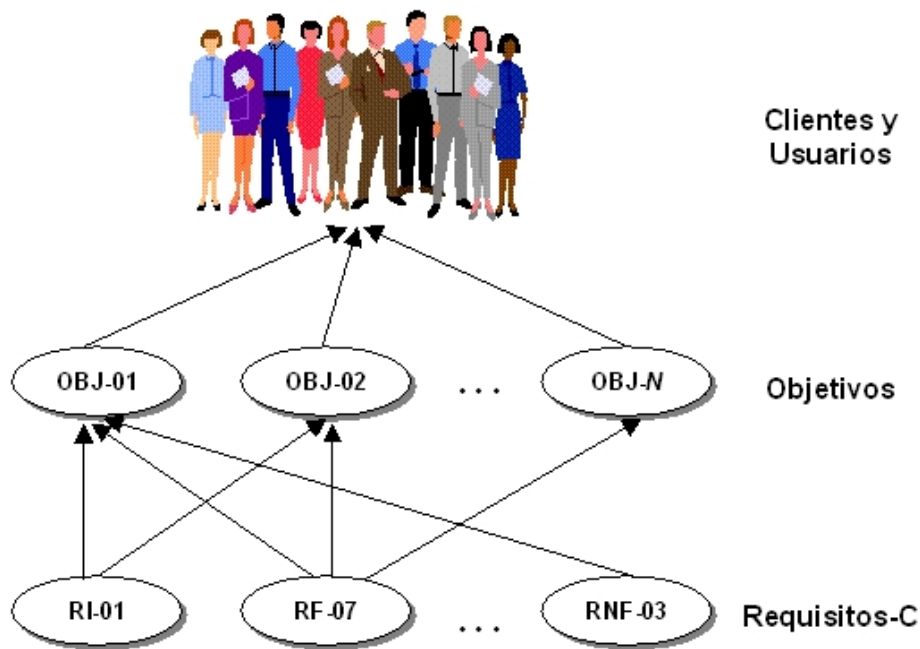


Figura 3.5: Rastreabilidad entre participantes, objetivos y requisitos-C

justificada como una necesidad para alcanzar uno o más objetivos. Esta es una de las relaciones de rastreabilidad, en concreto de *prerrastreabilidad* [Jarke 1998], que se contemplan en la propuesta (ver figuras 3.5 o 3.33).

Los productos resultantes de esta tarea son los objetivos del sistema expresados mediante las plantillas para objetivos descritas en la sección 3.6.1.

Para la determinación de los objetivos se pueden utilizar técnicas como el análisis de factores críticos de éxito [MAP 1995, Raghavan *et al.* 1994] o alguna técnica similar de identificación de objetivos.

3.5.4 Tarea 4: Identificar/revisar los requisitos de información

El objetivo de las tareas 4, 5 y 6 es identificar, o revisar en función de posibles conflictos, los requisitos-C a partir de la información obtenida en las tareas anteriores. La división en tres tareas se ha realizado por simplicidad, no porque se asuma que esa deba ser la secuencia de realización, ya que habitualmente se realizan en paralelo.

En esta tarea en concreto se deben identificar o revisar los requisitos de almacenamiento de información que deberá satisfacer el sistema. Normalmente estos requisitos son la respuesta a la pregunta *¿qué información, relevante para los objetivos de su negocio, deberá almacenar el sistema?*.

Este tipo de requisitos no suele incluirse como un grupo separado en las metodologías actuales, sin embargo en nuestra opinión, y siguiendo las propuestas de [Pohl 1997] y [Robertson y Robertson 1998], es importante identificarlos claramente.

Otros enfoques, por ejemplo [Iturrioz 1998], no incluyen estos requisitos porque pueden deducirse de los requisitos funcionales, especialmente si se utilizan casos de uso. En nuestra opinión, aunque es cierto que la información que debe almacenar el sistema puede deducirse de los casos de uso, creemos conveniente hacerla explícita, teniendo en cuenta que el tipo de sistemas para los que está pensada la propuesta metodológica son los sistemas de información.

Una posible división de esta tarea en subtareas podría recomendar identificar primero los conceptos relevantes sobre los que el sistema debe almacenar información, para posteriormente ir identificando los datos específicos sobre cada concepto relevante.

Por lo tanto, los productos de esta tarea serían los requisitos de almacenamiento de información, expresados usando las plantillas que se describen en la sección 3.6.2.

3.5.5 Tarea 5: Identificar/revisar los requisitos funcionales

En esta tarea se deben identificar o revisar los requisitos funcionales que deberá satisfacer el sistema, para lo que se ha optado por la utilización de los casos de uso [Jacobson *et al.* 1993]. Estos requisitos suelen obtenerse como respuesta a la pregunta *¿qué debe hacer el sistema con la información almacenada para alcanzar los objetivos de su negocio?* o *¿qué debe permitir el sistema hacer a los usuarios con la información almacenada?*.

Además de los casos de uso, en esta tarea es necesario identificar y describir a los *actores* del sistema [Jacobson *et al.* 1993]. Otro aspecto importante es la determinación del ámbito del sistema [Lilly 2000], es decir qué aspectos serán responsabilidad del sistema y qué aspectos se gestionarán manualmente o por otro procedimiento. La utilización de los diagramas de casos de uso [Booch *et al.* 1999], permite de forma sencilla especificar claramente qué queda dentro y qué queda fuera del ámbito del sistema (ver figura 3.6).

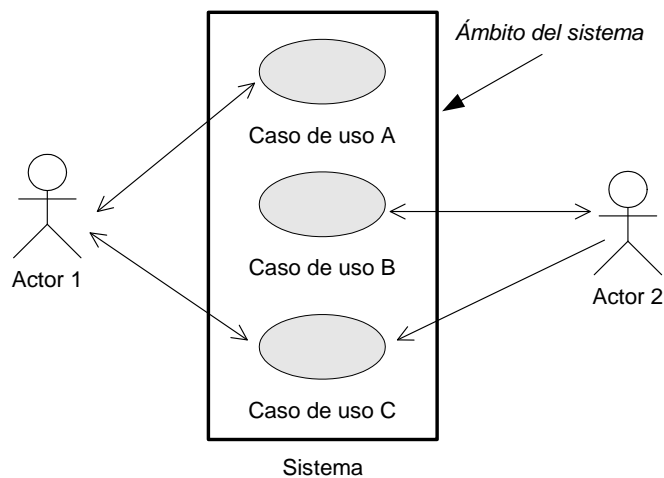


Figura 3.6: Representación del ámbito en los diagramas de casos de uso

Aunque en esta propuesta no se aborda una división en subtareas para dejar una mayor libertad a los desarrolladores, una posible división de esta tarea podría recomendar identificar primero los actores, posteriormente los casos de uso asociados a cada actor, desarrollar la secuencia normal de interacciones para cada caso de uso y, finalmente, contemplar la posibilidad de excepciones. Otra posibilidad es la planteada en [Díaz y Rodríguez 1998], en la que, partiendo de los objetivos, se van identificando los casos de uso asociados a los actores, se estudian las distintas alternativas de cada caso de uso y se determina la responsabilidad de los distintos criterios que dan lugar a las alternativas asignándolas al sistema o a otros actores.

Los productos de esta tarea son los diagramas de casos de uso y las especificaciones tanto de los actores como de los casos de uso expresados usando las plantillas propuestas en las secciones 3.6.3 y 3.6.4 respectivamente.

3.5.6 Tarea 6: Identificar/revisar los requisitos no funcionales

En esta tarea se deben identificar o revisar los requisitos no funcionales del sistema, normalmente relacionados con aspectos técnicos o legales: comunicaciones, interfaces con otros sistemas, fiabilidad, entorno de desarrollo, portabilidad, etc.

Los productos de esta tarea son los requisitos no funcionales expresados mediante la plantilla propuesta en la sección 3.6.5.

3.5.7 Tarea 7: Priorizar objetivos y requisitos

En esta última tarea se deben asignar prioridades a los objetivos y requisitos estableciendo su importancia y su urgencia, de forma que en el caso de que se desarrolle incrementalmente se tengan los criterios suficientes para saber qué requisitos deben implementarse en cada versión que se vaya entregando.

Los productos de esta tarea son los objetivos y requisitos identificados/revisado en las tareas anteriores con su prioridad establecida.

3.5.8 El documento de requisitos del sistema

En la propuesta metodológica, el producto final entregable de las actividades de elicitación de requisitos es el *Documento de Requisitos del Sistema* (DRS), cuya estructura puede verse en la figura 3.7, y que se describe con detalle en el apéndice B.

Este documento coincide básicamente con el *Operational Concept Document* de la norma MIL-STD-498 [DoD 1994, Fairley y Thayer 1997], el *User Requirements Document* de las normas PSS-05 [Mazza *et al.* 1994] y el *Documento de Requisitos del Sistema* de Métrica V2.1 [MAP 1995]. En el apéndice A puede consultarse más información sobre dichos documentos.

Un elemento importante de este documento es la matriz de rastreabilidad entre objetivos y requisitos. Dicha matriz, de la que puede verse un ejemplo en la figura 3.8, expresa de forma tabular la relación entre objetivos y requisitos que puede verse de forma gráfica en las figuras 3.5 y 3.33.

La utilidad de esta matriz radica en que permite por un lado una visión rápida de las relaciones de dependencias entre objetivos y requisitos, y por otro lado permite realizar una rápida comprobación de si todos los objetivos tienen algún requisito asociado y de si todos los requisitos están justificados por un determinado objetivo. Además permite estudiar el impacto de posibles cambios en los requisitos.

Durante la aplicación práctica de las ideas expuestas en esta tesis, hemos podido constatar la dificultad para establecer relaciones de rastreabi-

Portada
Lista de cambios
Índice
Lista de figuras
Lista de tablas
1 Introducción
2 Participantes en el proyecto
3 Descripción del sistema actual <i>[opcional]</i>
4 Objetivos del sistema
5 Catálogo de requisitos del sistema
5.1 Requisitos de almacenamiento de información
5.2 Requisitos funcionales
5.2.1 Diagramas de casos de uso
5.2.2 Definición de actores
5.2.3 Casos de uso del sistema
5.3 Requisitos no funcionales
6 Matriz de rastreabilidad objetivos/requisitos
7 Conflictos pendientes de resolución <i>[opcional, pueden ir en un documento aparte]</i>
8 Glosario de términos <i>[opcional]</i>
Apéndices <i>[opcionales]</i>

Figura 3.7: Estructura del Documento de Requisitos del Sistema

	OBJ-01	OBJ-02	...	OBJ- <i>n</i>
RI-01	•	•		
RI-02		•		•
...				
RF-01	•			•
RF-02	•	•		
...				
...				

Figura 3.8: Matriz de rastreabilidad del DRS

lidad entre objetivos y requisitos no funcionales. Normalmente los objetivos son del tipo "El sistema deberá gestionar un < determinado aspecto de la organización > ", dónde los aspectos de la organización pueden ser la gestión del almacén, gestión de clientes, etc. No parece sencillo asociar a unos objetivos de este tipo requisitos que, por ejemplo, indiquen la necesidad de una determinada fiabilidad o portabilidad.

En la estructura propuesta se han incluido los conflictos en el DRS, como se propone en la plantilla para documentos de requisitos de Volere [Robertson y Robertson 1999], aunque también puede optarse por mantenerlos en un documento aparte, el *Registro de Conflictos* por ejemplo, que se actualizaría durante todo el proceso de ingeniería de requisitos.

3.6 Plantillas y patrones lingüísticos para elicitación de requisitos

Las plantillas y patrones lingüísticos que se presentan en los siguientes apartados están pensados para utilizarse tanto durante las reuniones de elicitación con clientes y usuarios como para registrar y gestionar los requisitos-C.

Su objetivo es doble: por un lado intentar paliar la falta de propuestas concretas sobre la expresión de requisitos-C. Por otro lado, también pueden usarse como elementos de elicitación y negociación durante las reuniones con clientes y usuarios de forma similar a las conocidas tarjetas CRC (*Clase, Responsabilidad, Colaboración*) [Wirfs-Brock *et al.* 1990] (ver figura 3.9).

De esta forma se consigue que durante las sesiones de elicitación se trabaje con una filosofía WYSIWYG, tal como se propone en las técnicas de JAD o brainstorming, ya que los participantes manejan directamente la documentación final, favoreciéndose así su implicación en el proceso.

Como fruto de la experiencia de su utilización, para algunos campos de las plantillas se han identificado frases "estándar" que son habituales en las especificaciones de requisitos y que se han parametrizado. Estas frases, a las que hemos denominado *patrones lingüísticos*, o abreviadamente *patrones-L*, pueden usarse para rellenar los campos de las plantillas dándole valores a los parámetros con la información oportuna.

Ambos aspectos, la estructuración de la información en forma de plantilla y la propuesta de frases "estándar", facilita la redacción de los requi-

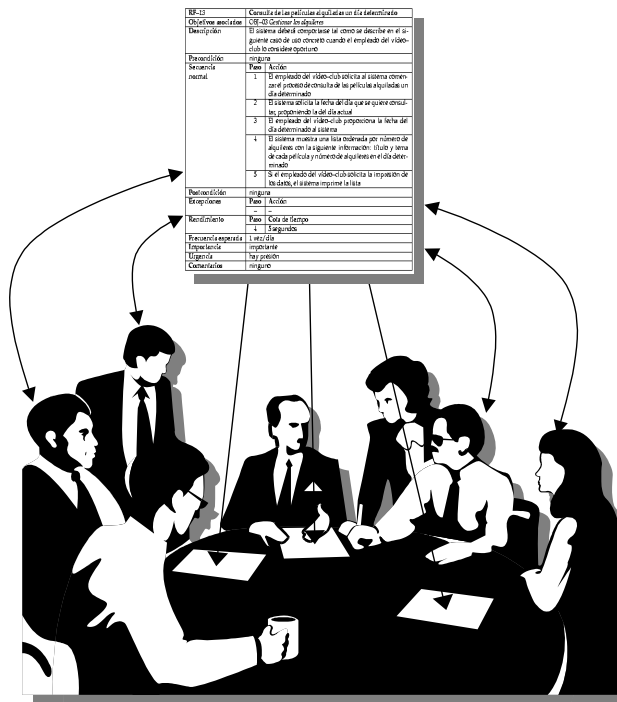


Figura 3.9: La plantilla como elemento de elicitación y negociación

sitos, permitiendo a los participantes en las actividades de elicitación centrarse en expresar sus necesidades y no en cómo expresarlas.

En la notación usada para describir los patrones-L, las palabras o frases entre < y > deben ser convenientemente reemplazadas, mientras que las palabras o frases que se encuentren entre { y } y separadas por comas representan opciones de las que se debe escoger una.

En las siguientes secciones se describen las plantillas propuestas y los patrones-L identificados.

3.6.1 Plantilla para los objetivos del sistema

Los objetivos del sistema pueden considerarse como *requisitos de alto nivel* [Sawyer y Kontoya 1999], de forma que los requisitos propiamente dichos serían la forma de alcanzar los objetivos. La plantilla propuesta para los objetivos puede verse en la figura 3.10.

El significado de los campos que la componen, cuya mayoría está presente también en las plantillas para los requisitos, es el siguiente:

OBJ-<i><id></i>	<i><nombre descriptivo></i>
Versión	<i><nº de la versión actual> (<fecha de la versión actual>)</i>
Autores	<ul style="list-style-type: none"> • <i><autor de la versión actual> (<organización del autor>)</i> ...
Fuentes	<ul style="list-style-type: none"> • <i><fuente de la versión actual> (<organización de la fuente>)</i> ...
Descripción	El sistema deberá <i><objetivo a cumplir por el sistema></i>
Subobjetivos	<ul style="list-style-type: none"> • OBJ-x <i><nombre del subobjetivo></i> • ...
Importancia	<i><importancia del objetivo></i>
Urgencia	<i><urgencia del objetivo></i>
Estado	<i><estado del objetivo></i>
Estabilidad	<i><estabilidad del objetivo></i>
Comentarios	<i><comentarios adicionales sobre el objetivo></i>

Figura 3.10: Plantilla y patrones-L para objetivos

- **Identificador y nombre descriptivo:** siguiendo la propuesta, entre otros, de [Sawyer *et al.* 1997], cada objetivo debe identificarse por un código único y un nombre descriptivo. Con objeto de conseguir una rápida identificación, los identificadores de los objetivos comienzan con *OBJ*.
- **Versión:** para poder gestionar distintas versiones, este campo contiene el número y la fecha de la versión actual del objetivo.
- **Autores, Fuentes:** estos campos contienen el nombre y la organización de los autores (normalmente desarrolladores) y de las fuentes (clientes o usuarios), de la versión actual del objetivo, de forma que la rastreabilidad pueda llegar hasta las personas que propusieron la necesidad del requisito.
- **Descripción:** este campo contiene un patrón-L que se debe completar con la descripción del objetivo.
- **Subobjetivos:** en este campo pueden indicarse los subobjetivos que dependen del objetivo que se está describiendo. En sistemas complejos puede ser necesario establecer una jerarquía de objetivos previa a la identificación de los requisitos. En caso de que ésto no sea necesario, puede ignorarse este campo.
- **Importancia:** este campo indica la importancia del cumplimiento del objetivo para los clientes y usuarios. Se puede asignar un valor numérico o alguna expresión enumerada como *vital*, *importante* o *que-*

daría bien, tal como se propone en [OOTC 1996]. En el caso de que no se haya establecido aún la importancia, se puede indicar que está *por determinar (PD)*, equivalente al TBD (*To Be Determined*) empleado en las especificaciones escritas en inglés.

- **Urgencia:** este campo indica la urgencia del cumplimiento del objetivo para los clientes y usuarios en el supuesto caso de un desarrollo incremental. Como en el caso anterior, se puede asignar un valor numérico o una expresión enumerada como *inmediatamente*, *hay presión* o *puede esperar* [OOTC 1996], o *PD* en el caso de que aún no se haya determinado.
- **Estado:** este campo indica el estado del objetivo desde el punto de vista de su desarrollo. El objetivo puede estar *en construcción* si se está elaborando, *pendiente de negociación* si tiene algún conflicto asociado pendiente de solución, *pendiente de validación* si no tiene ningún conflicto pendiente y está a la espera de validación o, por último, puede estar *validado* si ha sido validado por clientes y usuarios.
- **Estabilidad:** este campo indica la estabilidad del objetivo, es decir una estimación de la probabilidad de que pueda sufrir cambios en el futuro. Esta estabilidad puede indicarse mediante un valor numérico o mediante una expresión enumerada como *alta*, *media* o *baja* o *PD* en el caso de que aún no se haya determinado.

La información sobre la estabilidad, bien a nivel de objetivos como en este caso, bien a nivel de requisitos, ayuda a los diseñadores a diseñar software que prevea de antemano la necesidad de posibles cambios futuros en aquellos aspectos relacionados con los elementos identificados como inestables durante la fase de ingeniería de requisitos, favoreciendo así el mantenimiento y la evolución del software [Brackett 1990].

- **Comentarios:** cualquier otra información sobre el objetivo que no encaje en los campos anteriores puede recogerse en este apartado.

3.6.2 Plantilla para requisitos de almacenamiento de información

Lo más importante en los sistemas de información es precisamente la información que gestionan. La plantilla para requisitos de almacenamiento de información, que puede verse en la figura 3.11, ayuda a los clientes y

RI-<i><id></i>	<i><nombre descriptivo></i>
Versión	<i><nº de la versión actual> (<fecha de la versión actual>)</i>
Autores	<ul style="list-style-type: none"> • <i><autor de la versión actual> (<organización del autor>)</i> ...
Fuentes	<ul style="list-style-type: none"> • <i><fuente de la versión actual> (<organización de la fuente>)</i> ...
Objetivos asociados	<ul style="list-style-type: none"> • <i>OBJ-x <nombre del objetivo></i> ...
Requisitos asociados	<ul style="list-style-type: none"> • <i>Rx-y <nombre del requisito></i> ...
Descripción	El sistema deberá almacenar la información correspondiente a <i><concepto relevante></i> . En concreto:
Datos específicos	<ul style="list-style-type: none"> • <i><datos específicos sobre el concepto relevante></i> • ...
Intervalo temporal	{ pasado y presente, sólo presente }
Importancia	<i><importancia del requisito></i>
Urgencia	<i><urgencia del requisito></i>
Estado	<i><estado del requisito></i>
Estabilidad	<i><estabilidad del requisito></i>
Comentarios	<i><comentarios adicionales sobre el requisito></i>

Figura 3.11: Plantilla y patrones-L para requisitos de almacenamiento de información

usuarios a responder a la pregunta "*¿qué información, relevante para los objetivos de su negocio, debe ser almacenada por el sistema?*".

El significado de los campos de la plantilla es el siguiente:

- **Identificador y nombre descriptivo:** siguiendo las recomendaciones, entre otros, de [IEEE 1993] y [Sawyer *et al.* 1997], cada requisito se debe identificar por un código único y un nombre descriptivo. Con objeto de conseguir una rápida identificación, los identificadores de los requisitos de almacenamiento de información comienzan con *RI*.
- **Versión, Autores, Fuentes:** estos campos tienen el mismo significado que en la plantilla para objetivos aunque referidos al requisito.
- **Objetivos asociados:** este campo debe contener una lista con los objetivos a los que está asociado el requisito. Esto permite conocer qué requisitos harán que el sistema a desarrollar alcance los objetivos propuestos y justifican de esta forma la existencia o propósito del requisito.

- **Descripción:** para los requisitos de almacenamiento de información este campo usa un patrón-L que se debe completar con el concepto relevante sobre el que se debe almacenar información.
- **Datos específicos:** este campo contiene una lista de los datos específicos asociados al concepto relevante, de los que pueden indicarse todos aquellos aspectos que se considere oportunos (descripción, restricciones, ejemplos, etc.).
- **Intervalo temporal:** este campo indica durante cuánto tiempo es relevante la información para el sistema. Puede tomar los valores *pasado y presente*, si la información es siempre relevante, o *sólo presente* si la información tiene un periodo de validez concreto.

Por ejemplo, si el concepto es *empleados*, y el intervalo de tiempo es *pasado y presente*, quiere decir que los ex-empleados son relevantes para el sistema, mientras que un periodo de tiempo de *sólo presente* indicaría que los ex-empleados no se deben considerar.

Un intervalo temporal de pasado y presente suele implicar considerar la necesidad de dispositivos de almacenamiento con grandes capacidades o la necesidad de algún tipo de archivos *históricos*.

- **Requisitos asociados:** en este campo se indican otros requisitos que estén asociados por algún motivo con el requisito que se está describiendo, permitiendo así tener una rastreabilidad *horizontal*, similar a las relaciones entre *assets* del mismo nivel descritas en [García 2000].
- **Importancia, Urgencia, Estado, Estabilidad, Comentarios:** estos campos tienen el mismo significado que en la plantilla para objetivos aunque referidos al requisito.

3.6.3 Plantilla para actores

Aunque, estrictamente hablando, los actores de los casos de uso no son requisitos, por homogeneidad con el estilo de definición del resto de los elementos que componen el catálogo de requisitos se ha descrito la plantilla para definirlos que puede verse en la figura 3.12.

El único campo específico de esta plantilla es la descripción, en la que se usa un patrón-L que debe completarse con la descripción del rol o papel que representa el actor respecto al sistema. El significado del resto de los campos es el mismo que para las plantillas anteriores.

ACT-<i><id></i>	<i><nombre descriptivo></i>
Versión	<i><nº de la versión actual> (<fecha de la versión actual>)</i>
Autores	<ul style="list-style-type: none"> • <i><autor de la versión actual> (<organización del autor>)</i> ...
Fuentes	<ul style="list-style-type: none"> • <i><fuente de la versión actual> (<organización de la fuente>)</i> ...
Descripción	Este actor representa a <i><rol que representa el actor></i>
Comentarios	<i><comentarios adicionales sobre el actor></i>

Figura 3.12: Plantilla y patrones-L para actores

3.6.4 Plantilla para requisitos funcionales

Los sistemas de información no sólo almacenan información, también deben proporcionar servicios usando la información que almacenan. La plantilla de requisitos funcionales, que puede verse en la figura 3.13, describe casos de uso y ayuda a los clientes y usuarios a responder a la pregunta "*¿qué debe hacer el sistema con la información almacenada para alcanzar los objetivos de su negocio?*".

El significado de los campos específicos de esta plantilla es el siguiente (los campos comunes con la plantilla para requisitos de almacenamiento de información tienen el mismo significado):

- **Identificador y nombre descriptivo:** igual que en la plantilla anterior, excepto que los identificadores de los requisitos funcionales empiezan con *RF* y que el nombre descriptivo suele coincidir con el objetivo que los actores esperan alcanzar al realizar el caso de uso.

No se debe confundir este objetivo con los objetivos del sistema. El objetivo que los actores esperan alcanzar al realizar un caso de uso es de más bajo nivel, por ejemplo *registrar un nuevo socio* o *consultar los pedidos pendientes*.

- **Descripción:** para los requisitos funcionales, este campo contiene un patrón-L que debe completarse de forma distinta en función de que el caso de uso sea abstracto o concreto (ver sección 3.4.4.2, pág. 74).

Si el caso de uso es abstracto, deben indicarse los casos de uso en los que se debe realizar, es decir, aquellos desde los que es *incluido* o a los que *extiende*. Si, por el contrario, se trata de un caso de uso concreto, se debe indicar el *evento de activación* que provoca su realización.

En versiones anteriores de este patrón-L, aparecían las expresiones *caso de uso abstracto* y *caso de uso concreto*. La experiencia durante la

RF-<i><id></i>	<i><nombre descriptivo></i>	
Versión	<i><nº de la versión actual> (<fecha de la versión actual>)</i>	
Autores	<ul style="list-style-type: none"> • <i><autor de la versión actual> (<organización del autor>)</i> ... 	
Fuentes	<ul style="list-style-type: none"> • <i><fuente de la versión actual> (<organización de la fuente>)</i> ... 	
Objetivos asociados	<ul style="list-style-type: none"> • OBJ-<i>x</i> <i><nombre del objetivo></i> ... 	
Requisitos asociados	<ul style="list-style-type: none"> • Rx-<i>y</i> <i><nombre del requisito></i> ... 	
Descripción	El sistema deberá comportarse tal como se describe en el siguiente caso de uso { durante la realización de los casos de uso <i><lista de casos de uso></i> , cuando <i><evento de activación></i> }	
Precondición	<i><precondición del caso de uso></i>	
Secuencia normal	Paso	Acción
	<i>p</i> ₁	{El actor <i><actor></i> , El sistema} <i><acción/es realizada/s por actor/sistema></i>
	<i>p</i> ₂	Se realiza el caso de uso <i><caso de uso (RF-<i>x</i>)></i>
	<i>p</i> ₃	Si <i><condición></i> , {el actor <i><actor></i> , el sistema} <i><acción/es realizada/s por actor/sistema></i>
	<i>p</i> ₄	Si <i><condición></i> , se realiza el caso de uso <i><caso de uso (RF-<i>x</i>)></i>

Postcondición	<i><postcondición del caso de uso></i>	
Excepciones	Paso	Acción
	<i>p</i> _{<i>i</i>}	Si <i><condición de excepción></i> , {el actor <i><actor></i> , el sistema} <i><acción/es realizada/s por actor/sistema></i> , a continuación este caso de uso {continúa , termina}
	<i>p</i> _{<i>j</i>}	Si <i><condición de excepción></i> , se realiza el caso de uso <i><caso de uso (RF-<i>x</i>)></i> , a continuación este caso de uso {continúa , termina}

Rendimiento	Paso	Cota de tiempo
	<i>q</i>	<i>m</i> <i><unidad de tiempo></i>

Frecuencia esperada	<i><nº de veces></i> veces / <i><unidad de tiempo></i>	
Importancia	<i><importancia del requisito></i>	
Urgencia	<i><urgencia del requisito></i>	
Estado	<i><estado del requisito></i>	
Estabilidad	<i><estabilidad del requisito></i>	
Comentarios	<i><comentarios adicionales sobre el requisito></i>	

Figura 3.13: Plantilla y patrones-L para requisitos funcionales (casos de uso)

utilización de estas plantillas en los tres proyectos reales en los que se ha utilizado (ver sección 3.9) nos ha llevado a eliminar dichas expresiones, que resultaban difíciles de entender por los participantes en el proceso de elicitación.

- **Precondición:** en este campo se expresan en lenguaje natural las condiciones necesarias para que se pueda realizar el caso de uso.
- **Secuencia normal:** este campo contiene la secuencia normal de interacciones del caso de uso. En cada paso, un actor o el sistema realiza una o más acciones, o se realiza (se *incluye*) otro caso de uso. Un paso puede tener una condición de realización, en cuyo caso si se realizara otro caso de uso se tendría una relación de *extensión*. Se asume que, después de realizar el último paso, el caso de uso termina.

Otras propuestas similares, por ejemplo [Coleman 1998], proponen utilizar estructuras similares al pseudocódigo para expresar las interacciones de los casos de uso. En nuestra opinión, esto puede llevar a que dichas descripciones sean excesivamente complejas de entender para los participantes sin conocimientos de programación y se corre el peligro de especificar los casos de uso con un estilo cercano a la programación.

Para representar estructuras condicionales complejas se puede recurrir a añadir información aparte, por ejemplo una tabla de decisión, y referenciarla desde el paso o los pasos oportunos.

En el caso de estructuras iterativas, su uso puede evitarse con un uso cuidadoso del lenguaje natural. Por ejemplo, para indicar que se procesan todos los artículos de un pedido se puede optar por frases como "*el sistema procesa todos los artículos del pedido introducidos por el usuario*", en lugar de estructuras como:

```
REPETIR
  procesar artículo del pedido introducido por el usuario
HASTA que no haya más artículos
```

Otro ejemplo puede ser especificar que el usuario puede intentar conectarse al sistema un máximo de tres veces. Una posible especificación sería la que puede verse en la figura 3.14, bastante más *natural* y fácil de entender que la que puede verse en la figura 3.15 utilizando la propuesta descrita en [Coleman 1998].

Secuencia normal	Paso	Acción
	1	El sistema solicita al usuario su nombre de usuario y su clave de acceso
	2	El usuario proporciona al sistema su nombre y su clave de acceso
	3	El sistema comprueba si el nombre de usuario y la clave de acceso son correctas
	4	Si el nombre de usuario y la clave no son correctas, el sistema permite al usuario repetir el intento (pasos 1–3) hasta un máximo de tres veces
	5	Si el nombre de usuario y la clave son correctas, el sistema permite el acceso al usuario
Excepciones	Paso	Acción
	4	Si el usuario ha intentado tres veces acceder sin éxito, el sistema rechaza el acceso del usuario, a continuación este caso de uso termina

Figura 3.14: Ejemplo de caso de uso de conexión de usuario (plantilla)

```

1. El sistema inicializa intentos a 0

2. REPETIR
    2.1 El sistema solicita al usuario su nombre de usuario
    2.2 El usuario proporciona al sistema su nombre de usuario
    2.3 El sistema solicita al usuario su clave
    2.4 El usuario proporciona al sistema su clave
    2.5 El sistema comprueba si el nombre de usuario y
        la clave son correctas
    2.6 El sistema incrementa intentos
HASTA QUE el nombre de usuario y la clave sean correctas o
        intentos = 3

3. SI el nombre de usuario y la clave son correctas
    3.1 El sistema permite el acceso al usuario
SINO
    3.2 El sistema rechaza el acceso del usuario
FINSI

```

Figura 3.15: Ejemplo de caso de uso de conexión de usuario (Coleman)

- **Postcondición:** en este campo se expresan en lenguaje natural las condiciones que se deben cumplir después de la terminación normal del caso de uso.
- **Excepciones:** este campo especifica el comportamiento del sistema en el caso de que se produzca alguna situación excepcional durante la realización de un paso determinado.

Después de realizar las acciones o el caso de uso asociados a la excepción (una *extensión*), el caso de uso puede continuar la secuencia normal o terminar, lo que suele ir acompañado por una cancelación de todas las acciones realizadas en el caso de uso dejando al sistema en el mismo estado que antes de comenzar el caso de uso, asumiendo una semántica *transaccional*.

Inicialmente, la expresión utilizada para indicar una terminación anormal del caso de uso como resultado de una excepción era "*este caso de uso aborta*". La experiencia durante su aplicación nos llevó a la conclusión de que el término *abortar* resultaba *emocionalmente* molesto para algunos participantes [Goleman 1996], por lo que se cambió por "*este caso de uso termina*" con el significado comentado anteriormente.

- **Rendimiento:** en este campo puede especificarse el tiempo máximo para cada paso en el que el sistema realice un acción.
- **Frecuencia esperada:** en este campo se indica la frecuencia esperada de realización del caso de uso, que aunque no es realmente un requisito, es una información interesante para los desarrolladores.

3.6.5 Plantilla para requisitos no funcionales

Los requisitos no funcionales del sistema se pueden expresar usando la plantilla que puede verse en la figura 3.16. El único campo específico de esta plantilla es la descripción, en la que se usa un patrón-L que debe completarse con la capacidad que deberá presentar el sistema, el significado del resto de los campos es el mismo que para las plantillas anteriores.

RNF-<i><id></i>	<i><nombre descriptivo></i>
Versión	<i><nº de la versión actual> (<fecha de la versión actual>)</i>
Autores	• <i><autor de la versión actual> (<organización del autor>)</i> ...
Fuentes	• <i><fuente de la versión actual> (<organización de la fuente>)</i> ...
Objetivos asociados	• <i>OBJ-x <nombre del objetivo></i> ...
Requisitos asociados	• <i>Rx-y <nombre del requisito></i> ...
Descripción	El sistema deberá <i><capacidad del sistema></i>
Importancia	<i><importancia del requisito></i>
Urgencia	<i><urgencia del requisito></i>
Estado	<i><estado del requisito></i>
Estabilidad	<i><estabilidad del requisito></i>
Comentarios	<i><comentarios adicionales sobre el requisito></i>

Figura 3.16: Plantilla y patrones-L para requisitos no funcionales

3.6.6 Plantilla para conflictos

Como ya se ha comentado, durante las sesiones de elicitación puede ser necesario resolver mediante algún tipo de negociación posibles conflictos en los requisitos-C elicitados en iteraciones previas del proceso. Para documentar dichos conflictos, y las soluciones adoptadas, se propone la plantilla que puede verse en la figura 3.17.

CFL-<i><id></i>	<i><nombre descriptivo></i>
Versión	<i><nº de la versión actual> (<fecha de la versión actual>)</i>
Autores	• <i><autor de la versión actual> (<organización del autor>)</i> ...
Fuentes	• <i><fuente de la versión actual> (<organización de la fuente>)</i> ...
Objs./Reqs. en conflicto	• <i>OBJ-x/Rx-y <nombre del objetivo o requisito en conflicto></i> ...
Descripción	<i><descripción del conflicto></i>
Alternativas	• <i><descripción alternativa de solución> (<autores alternativa>)</i> ...
Solución	<i><descripción de la solución adoptada (si se ha acordado)></i>
Importancia	<i><importancia de la resolución del conflicto></i>
Urgencia	<i><urgencia de la resolución del conflicto></i>
Estado	<i><estado del resolución del conflicto></i>
Comentarios	<i><comentarios adicionales sobre el conflicto></i>

Figura 3.17: Plantilla para conflictos

El significado de los campos de la plantilla es el siguiente:

- **Identificador y nombre descriptivo:** al igual que el resto de la información correspondiente a los requisitos–C, cada conflicto debe poderse identificar de forma única y tener un nombre descriptivo. El prefijo propuesto para lograr una rápida identificación es *CFL*.
- **Versión, Autores, Fuentes:** estos campos tienen el mismo significado que en las plantillas para objetivos y requisitos, aunque referidos al conflicto. En este caso especial, las fuentes son los participantes que deben participar en las posibles negociaciones necesarias para su resolución.
- **Objetivos y requisitos en conflicto:** este campo debe contener una lista con los objetivos y/o requisitos afectados por el conflicto.
- **Descripción:** este campo debe contener la descripción del conflicto.
- **Alternativas:** este campo debe contener una lista con las posibles alternativas de solución que se hayan identificado para solucionar el conflicto así como los autores de dicha alternativas.
- **Solución:** este campo debe contener la descripción de la solución negociada del conflicto, una vez que se haya acordado.
- **Importancia, Urgencia:** estos campos indican respectivamente la importancia y la urgencia de la resolución del conflicto.
- **Estado:** este campo indica el estado de resolución del conflicto, que podrá estar *en negociación* o bien *resuelto*.
- **Comentarios:** este campo tienen el mismo significado que en las plantillas descritas previamente.

3.7 Patrones de reutilización de requisitos–C

Las plantillas descritas se han utilizado hasta el momento en más de 80 prácticas académicas en asignaturas de ingeniería del software y en tres proyectos reales aún en desarrollo. Como resultado de esta experiencia se han ido identificando requisitos que, con pequeñas variaciones, aparecen

en un elevado número de ocasiones. A dichos requisitos los hemos denominado *patrones de reutilización de requisitos- R_C* , o abreviadamente, *patrones- R_C* .

Los campos de las plantillas para patrones- R_C son los mismos que para las plantillas para requisitos a excepción de los campos cuya información sólo tiene sentido dentro del contexto de un determinado proyecto.

3.7.1 Patrones- R_C de requisitos de almacenamiento de información

Dentro de los requisitos de almacenamiento de información se han identificado varios patrones- R_C que aparecen en un elevado número de los casos estudiados. Por un lado se ha cuantificado el número de veces que cada patrón- R_C ha aparecido, y por otro, para cada patrón- R_C se ha cuantificado el número de veces que cada dato específico ha aparecido.

Siguiendo la clasificación de reutilización descrita en [Keepence *et al.* 1995] y [Mannion *et al.* 1999], en la que los requisitos se clasifican como *no reutilizables*, *directamente reutilizables* por composición y *basados en parámetros* para reutilizar por generación, los patrones- R_C de requisitos de almacenamiento de información serían *directamente reutilizables*, aunque probablemente necesitarían adaptaciones específicas en cada caso.

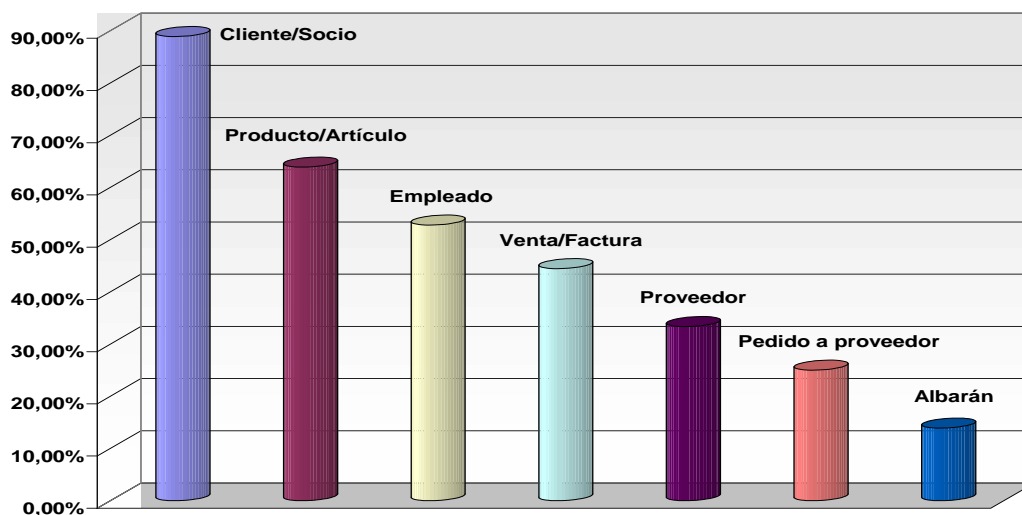


Figura 3.18: Frecuencia de aparición de algunos patrones- R_C de requisitos de almacenamiento de información

En la figura 3.18 puede verse gráficamente el porcentaje de aparición de los patrones- R_C de requisitos de almacenamiento de información más frecuentes.

Como ejemplo más significativo, en la figura 3.19 puede verse el patrón- R_C correspondiente a los requisitos de almacenamiento de información relativos a *clientes/socios*, en el que se han ordenado los distintos datos específicos según su frecuencia de aparición.

Otros ejemplos de este tipo de patrones- R_C pueden verse en las figuras 3.20 y 3.21, que al igual que en el caso del patrón- R_C *Cliente/Socio*, presentan un porcentaje elevado de casos en el que se incluyen otros datos específicos del sistema que se está especificando que no pueden generalizarse, lo que indica claramente la necesidad de adaptación a cada proyecto concreto. En nuestro caso, se han englobado como *otros* a aquellos datos específicos con una incidencia menor al 5%.

Los patrones- R_C de almacenamiento de información presentados responden únicamente a un análisis cuantitativo de su utilización en prácticas académicas. Lo ideal sería combinar estos datos cuantitativos, que pueden ayudar a *descubrir* patrones- R_C que pasan inadvertidos, con información cualitativa proveniente de expertos en dominios de problemas específicos, de forma que se obtengan patrones- R_C con una alta calidad y que puedan estar a disposición de los ingenieros de requisitos en algún tipo de repositorio, por ejemplo el descrito en [Manso *et al.* 1999, García 2000].

RI- x	Información sobre <clientes/socios>
Descripción	El sistema deberá almacenar la información correspondiente a <los clientes/socios del negocio>. En concreto:
Datos Específicos	<ul style="list-style-type: none"> • Nombre y apellidos del <cliente/socio> (100%) • Dirección del <cliente/socio> (93.75%) • Número de identificación fiscal del <cliente/socio> (90.63%) • Números de teléfono del <cliente/socio> (84.38%) • Código del <cliente/socio> (53.13%) • Fecha de alta/antigüedad del <cliente/socio> (34.38%) • Fecha de nacimiento/edad del <cliente/socio> (34.38%) • Sexo <cliente/socio> (25%) • Número de cuenta bancaria del <cliente/socio> (9.38%) • Dirección de correo electrónico del <cliente/socio> (6.25%) • Otros datos específicos del <cliente/socio> (81.25%)

Figura 3.19: Patrón- R_C *Cliente/Socio/Paciente* de requisito de almacenamiento de información

RI- <i>x</i>	Información sobre <productos/artículos>
Descripción	El sistema deberá almacenar la información correspondiente a <los productos/artículos del negocio>. En concreto:
Datos específicos	<ul style="list-style-type: none"> • Nombre del <producto/artículo> (91.30%) • Cantidad disponible del <producto/artículo> (69.57%) • Precio de venta al público del <producto/artículo> (56.52%) • Código del producto/artículo (56.52%) • Proveedores que suministran el <producto/artículo> (47.83%) • Descripción del <producto/artículo> (34.78%) • Otros datos específicos del <producto/artículo> (78.26%)

Figura 3.20: Patrón- R_C *Producto* de requisito de almacenamiento de información

RI- <i>x</i>	Información sobre <empleados>
Descripción	El sistema deberá almacenar la información correspondiente a <los empleados del negocio>. En concreto:
Datos específicos	<ul style="list-style-type: none"> • Nombre y apellidos del <empleado> (100%) • Dirección del <empleado> (100%) • Números de teléfono del <empleado> (89.47%) • Número de identificación fiscal del <empleado> (84.21%) • Fecha de contratación del <empleado> (57.89%) • Número de la Seguridad Social del <empleado> (26.32%) • Sueldo anual del <empleado> (21.05%) • Tipo de contrato del <empleado> (21.05%) • Fecha de nacimiento del <empleado> (10.53%) • Número de cuenta bancaria del <empleado> (5.26%) • Dirección de correo electrónico del <empleado> (5.26%) • Otros datos específicos del <empleado> (68,42%)

Figura 3.21: Patrón- R_C *Empleado* de requisito de almacenamiento de información

3.7.2 Patrones- R_C de requisitos funcionales

En el caso de los requisitos funcionales, dada la dificultad de su análisis cuantitativo debido a su heterogeneidad, se ha optado por una análisis cualitativo aunque reforzado cuantitativamente, ya que los 4 patrones- R_C de requisitos funcionales identificados aparecen en todas las especificaciones estudiadas, siempre con detalles específicos en cada caso.

Aplicando la clasificación de [Keepence *et al.* 1995, Mannion *et al.* 1999] comentada en la sección anterior, estos patrones- R_C serían requisitos *basados en parámetros*, reutilizables por lo tanto mediante generación al darles valores concretos a los parámetros.

Estos cuatro patrones- R_C corresponden a los típicos conceptos de *alta*, *baja*, *modificación* y *consulta* (CRUD, *Create*, *Read*, *Update* y *Delete* en inglés [Booch *et al.* 1999, pág. 401]) y describen los correspondientes cuatro casos de uso genéricos que pueden verse en las figuras 3.22, 3.23, 3.24 y 3.25, y que se describen en las siguientes secciones.

3.7.2.1 Patrón- R_C Crear/Dar de alta/Registrar

El patrón- R_C *Crear/Dar de alta/Registrar* de la figura 3.22 describe el caso de uso típico en que un actor proporciona nueva información al sistema como respuesta a una nueva situación en el entorno del que el sistema debe mantener información.

Siguiendo las recomendaciones para redactar casos de uso de, entre otros [Cockburn 1997], [Creel 2000] o [Rolland y Achour 1998], se han evitado referencias específicas a aspectos de implementación o de interfaz de usuario.

RF- x	{ <i>Crear, Dar de alta, Registrar</i> } <X>	
Descripción	El sistema deberá comportarse tal como se describe en el siguiente caso de uso cuando { <i>un cliente realiza una compra por primera vez, una persona solicita su ingreso como socio, ...</i> }	
Precondición	La información correspondiente al nuevo <X> no está almacenada todavía	
Secuencia normal	Paso	Acción
	1	El <actor> solicita al sistema comenzar el proceso de { <i>crear/dar de alta/registrar</i> } un nuevo <X>
	2	El sistema solicita los siguientes datos correspondientes al nuevo <X>: <datos solicitados por el sistema>
	3	El <actor> proporciona los datos requeridos y solicita al sistema que los almacene
	4	El sistema almacena los datos proporcionados e informa al <actor> de que el proceso ha terminado con éxito
Postcondición	El sistema ha almacenado la información correspondiente al nuevo <X>	
Excepciones	Paso	Acción
	3	Si el sistema detecta que los datos proporcionados ya están almacenados, el sistema informa de la situación al <actor> permitiéndole modificar los datos proporcionados, a continuación este caso de uso continúa
	3	Si el <actor> solicita cancelar la operación, el sistema cancela la operación, a continuación este caso de uso termina

Figura 3.22: Patrón- R_C *Crear/Dar de alta/Registrar* de requisito funcional

De forma más concreta, se ha utilizado el patrón *Presentación* descrito en [Creel 2000], que aconseja indicar qué datos debe mostrar o solicitar el sistema sin especificar cómo lo va a hacer (ver sección 3.8 en este mismo capítulo).

Las situaciones excepcionales habituales en este tipo de caso de uso suelen ser que el actor intente introducir como nueva una información ya existente o que cancele el proceso durante la introducción de los datos que le solicita el sistema.

3.7.2.2 Patrón- R_C *Eliminar/Dar de baja/Cancelar*

En la figura 3.23 puede verse el patrón- R_C *Eliminar/Dar de baja/Cancelar* que describe el caso de uso típico en que un actor informa al sistema que una determinada información relativa al entorno ha dejado de ser relevante y no es necesario que siga almacenándola.

Al igual que en el patrón anterior, se ha evitado hacer referencia a posibles aspectos de implementación o de interfaz de usuario, utilizando la

RF-x	{ <i>Eliminar, Dar de baja, Cancelar</i> } <X>	
Descripción	El sistema deberá comportarse tal como se describe en el siguiente caso de uso cuando { <i>un cliente deja de serlo por algún motivo, una socio solicita su baja, ...</i> }	
Precondición	El sistema tiene almacenada la información correspondiente al <X> a eliminar	
Secuencia normal	Paso	Acción
	1	El <actor> solicita al sistema comenzar el proceso de { <i>eliminar/dar de baja/cancelar</i> } la información correspondiente a un <X>
	2	El sistema solicita que se identifique al <X> a eliminar
	3	El <actor> identifica el <X> a eliminar y solicita al sistema su eliminación
	4	El sistema elimina los datos correspondientes al <X> a eliminar e informa al <actor> de que el proceso ha terminado con éxito
Postcondición	El sistema no tiene almacenada la información correspondiente al <X> eliminado	
Excepciones	Paso	Acción
	3	Si <la eliminación de la información del <X> viola alguna restricción>, el sistema comunica la situación al <actor> y cancela la operación, a continuación este caso de uso termina
	3	Si el <actor> solicita cancelar la operación, el sistema cancela la operación, a continuación este caso de uso termina

Figura 3.23: Patrón- R_C *Eliminar/Dar de baja/Cancelar* de requisito funcional

idea del patrón *Especificar* de [Creel 2000], en el que no se especifica la forma en que el usuario selecciona una determinada información almacenada en el sistema, dejando la puerta abierta a distintas implementaciones.

Aparte de la situación excepcional en la que el usuario cancela la operación, también se ha contemplado el caso en que la información no pueda eliminarse por motivos de integridad, por ejemplo que un cliente tenga pagos pendientes o que un socio de un vídeo-club tenga películas alquiladas.

3.7.2.3 Patrón- R_C *Modificar/Editar*

El patrón- R_C correspondiente a la típica modificación o edición de datos puede verse en la figura 3.24, en el que también se han aplicado los patrones *Especificar* y *Presentación* de [Creel 2000].

RF-x	{ <i>Modificar, Editar</i> } <X>	
Descripción	El sistema deberá comportarse tal como se describe en el siguiente caso de uso cuando { <i>un cliente/socio comunica un cambio de alguno de sus datos, ...</i> }	
Precondición	El sistema tiene almacenada la información correspondiente al <X> a { <i>modificar, editar</i> }	
Secuencia normal	Paso	Acción
	1	El <actor> solicita al sistema comenzar el proceso de { <i>modificar, editar</i> } la información correspondiente a un <X>
	2	El sistema solicita que se identifique al <X> a { <i>modificar, editar</i> }
	3	El <actor> identifica el <X> a { <i>modificar, editar</i> }
	4	El sistema muestra los siguientes datos correspondientes al <X> a modificar: <datos mostrados por el sistema>
	5	El sistema permite al <actor> modificar los siguientes datos: <datos que pueden modificarse>
	6	El <actor> modifica los datos que el sistema le permite y solicita al sistema que los almacene
	7	El sistema modifica los datos correspondientes al <X> a modificar e informa al <actor> de que el proceso ha terminado con éxito
Postcondición	El sistema ha actualizado la información correspondiente al <X> { <i>modificado, modificada</i> }	
Excepciones	Paso	Acción
	6	Si <la modificación de la información del <X> viola alguna restricción>, el sistema comunica la situación al usuario y cancela la operación, a continuación este caso de uso termina
	6	Si el <actor> solicita cancelar la operación, el sistema cancela la operación, a continuación este caso de uso termina

Figura 3.24: Patrón- R_C *Modificar/Editar* de requisito funcional

El patrón- R_C indica claramente en los pasos 4 y 5 cuáles son los datos que muestra al actor y cuáles de los datos mostrados le deja modificar, ya que habitualmente ambos conjuntos de datos no coinciden.

Una posible adaptación del patrón- R_C puede consistir en identificar distintos conjuntos de datos mostrados y datos modificables en función del tipo de actor que esté participando en el caso de uso, permitiendo de esta forma especificar requisitos de seguridad de forma integrada con los requisitos funcionales.

Al igual que en el patrón- R_C *Eliminar/Dar de baja/Cancelar*, una posibilidad de excepción en este caso de uso, aparte de la cancelación por parte del actor, es que la información modificada viole alguna restricción.

3.7.2.4 Patrón- R_C Consultar

El patrón- R_C correspondiente a una consulta sobre la información almacenada por el sistema puede verse en la figura 3.25. Este patrón- R_C es muy similar al patrón- R_C *Modificar/Editar*, con la diferencia de que en este patrón no se modifica la información almacenada por el sistema.

RF-x	Consultar <X>	
Descripción	El sistema deberá comportarse tal como se describe en el siguiente caso de uso cuando { <i>se considere oportuno consultar una determinada información, ...</i> }	
Precondición	Ninguna	
Secuencia normal	Paso	Acción
	1	El <actor> solicita al sistema comenzar el proceso de consultar la información correspondiente a { un, una } <X>
	2	El sistema solicita que se identifique { al, a la, a los } <X> a consultar
	3	El <actor> identifica { el, la, a los } <X> a consultar
	4	El sistema muestra los siguientes datos correspondientes { al, a la, a los } <X> a consultar: <datos mostrados por el sistema> en <un orden determinado> e informa al <actor> de que el proceso ha terminado con éxito
Postcondición	La información correspondiente { al, a la, a los } <X> a consultar no ha cambiado	
Excepciones	Paso	Acción
	3	Si el <actor> solicita cancelar la operación, el sistema cancela la operación, a continuación este caso de uso termina

Figura 3.25: Patrón- R_C Consultar de requisito funcional

Posibles variaciones de este patrón- R_C pueden contemplar la necesidad de que el actor proporcione alguna información al sistema para que éste seleccione la información a mostrar en lugar de identificar una determinada información, contemplar la posibilidad de que el actor solicite la impresión en papel del resultado mostrado o que se indique fuera de la plantilla el formato deseado en el que se desea que se muestre la información.

3.7.3 Patrones- R_C de requisitos no funcionales

Algunos de los aspectos que tradicionalmente se consideraban como *no funcionales*, por ejemplo el rendimiento, se han integrado en las plantillas propuestas para requisitos funcionales. Otros aspectos como seguridad pueden especificarse mediante casos de uso. Si a esto se le añade que la identificación de patrones- R_C se ha realizado principalmente sobre prácticas académicas y que los requisitos no funcionales pueden ser de naturaleza muy heterogénea, puede explicarse el hecho de que no se hayan identificado patrones- R_C de requisitos no funcionales significativos.

En la figura 3.26 puede verse un posible ejemplo de patrón- R_C de requisitos no funcionales para especificar bajo qué sistema operativo debería funcionar el sistema.

RNF- x	Sistema operativo
Descripción	El sistema deberá poder ejecutarse bajo el sistema operativo < <i>sistema operativo</i> >

Figura 3.26: Patrón- R_C *Sistema operativo* de requisito no funcional

3.8 Trabajos relacionados

3.8.1 Metodologías de elicitación

En general, la propuesta de tareas específicas para la elicitación de requisitos es escasa en la bibliografía sobre ingeniería de requisitos. Respecto a la metodología de elicitación propuesta, los trabajos más relacionados son principalmente la estructura de *actividades/tareas*, los productos y algunas de las actividades propuestas en el módulo de *Análisis de Requisitos*

del Sistema (ARS) de Métrica [MAP 1995], cuya relación con la actividades propuestas es la siguiente:

- 1 **Establecer el ámbito y alcance del proyecto:** esta actividad coincidiría con la tarea 1 (*Obtener información sobre el dominio del problema*), aunque en la propuesta se incluye también el estudio del sistema actual, de forma que se cuente con más información antes de mantener las reuniones de elicitación/negociación.
- 2 **Identificar y definir requisitos:** esta actividad coincidiría con las tareas 2 a 6 de la metodología propuesta, ya que abarca tanto las reuniones con los clientes y usuario como la identificación de los requisitos-C y la elaboración del catálogo de dichos requisitos.
- 3 **Diseñar el modelo lógico actual:** esta actividad está incluida en la tarea 1, como ya se ha comentado.
- 4 **Estudiar alternativas de construcción:** esta actividad no tiene ninguna tarea asignada en la metodología propuesta porque se ha considerado que en esta fase de desarrollo es excesivamente prematuro proponer alternativas de posibles implementaciones, sobre todo cuando aún no se ha realizado el análisis de los requisitos propuestos por clientes y usuarios.

Otro modelo de procesos para la elicitación, mucho más detallado que el presentado en este trabajo, es *Volere*, propuesto en [Robertson y Robertson 1999] y cuyo esquema original puede verse en la figura 3.27.

En *Volere* se denomina *proceso de requisitos* a las actividades de elicitación de requisitos, diferenciándolas claramente de las de análisis. Dentro de este proceso de requisitos se definen las siguientes tareas:

- 1 **Lanzamiento del proyecto:** en esta tarea se comienza el proyecto, normalmente con sesiones JAD, para determinar si es o no viable, si merece la pena invertir en su desarrollo y si hay un acuerdo general entre todos los participantes de la necesidad de abordar el desarrollo de un nuevo sistema. También se debe establecer el contexto del problema a tratar, establecer el propósito del producto a desarrollar y realizar unas primeras estimaciones de costes y de tiempo de desarrollo.

En la metodología propuesta en esta tesis no se contempla explícitamente la toma de decisiones sobre la viabilidad del proyecto, que se

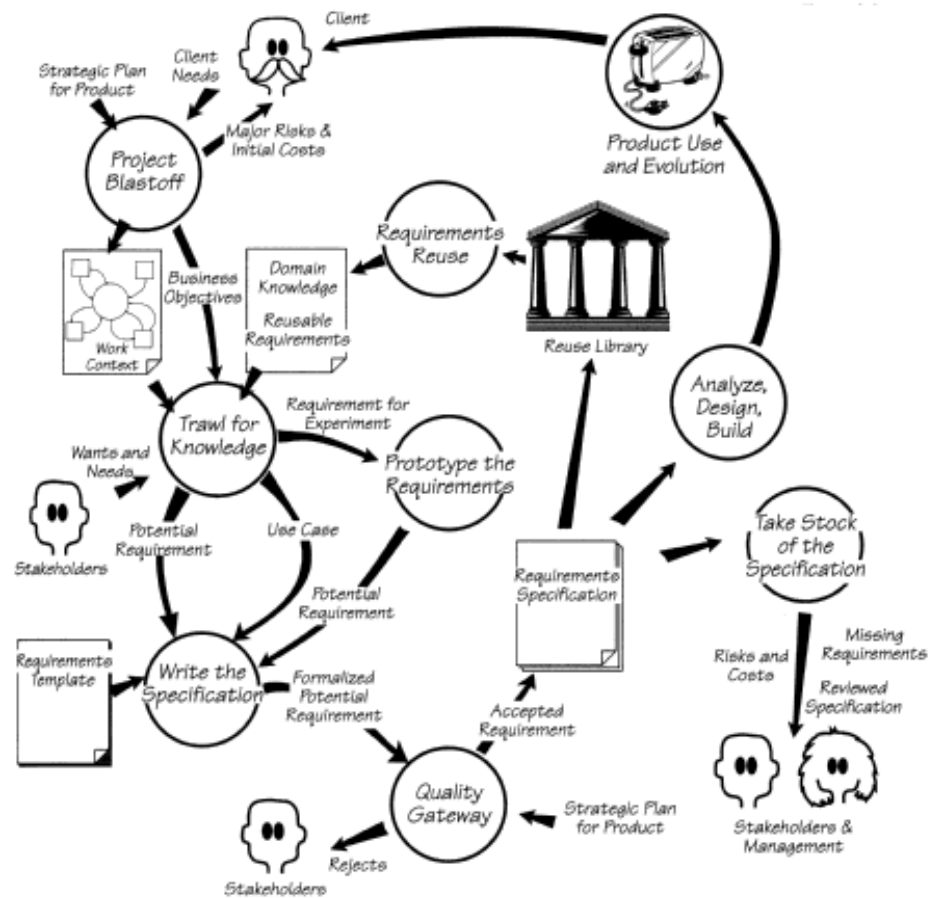


Figura 3.27: Modelo de procesos de elicitación de requisitos de Volere

delega en las actividades propias de gestión del proyecto. El establecimiento del contexto y el propósito se realizan básicamente durante la tarea 3 (*Identificar/revisar los objetivos del sistema*).

- 2 **Búsqueda de conocimiento:** esta tarea, que coincide básicamente con las tareas 1 y 2 de la propuesta que se describe en esta memoria, se centra en estudiar el dominio del problema y el sistema actual mediante reuniones y *aprendizaje*. El objetivo es detectar los eventos a nivel del negocio y los casos de uso de negocio asociados a dichos eventos para, posteriormente, producir los requisitos provisionales en sesiones de *brainstorming*.
- 3 **Prototipar los requisitos:** para presentar los requisitos provisionales a los usuarios se recomienda el uso de prototipos, tanto dibujados a mano como con herramientas específicas, conjuntamente con el uso

de escenarios. En la metodología propuesta, el prototipado se contemplan durante las actividades de análisis.

- 4 **Escribir la especificación:** en esta tarea se deben escribir los requisitos, teniendo especial cuidado en usar el lenguaje del cliente. Para ello, se propone aplicar las plantillas para documentos de requisitos y para requisitos individuales propuesta por los autores (ver figura 3.28). Básicamente, esta tarea coincide con las tareas 3 a 6 de la metodología descrita en este capítulo.
- 5 **La pasarela de la calidad:** en esta tarea, cada requisito individual que se va generando debe ser aprobado por un equipo formado por un representante de los usuarios y un ingeniero de requisitos, validándolo y verificándolo. Tanto esta tarea como la siguiente, incluyen aspectos de validación que son tratados en la metodología para validación descrita en el capítulo 5 de esta memoria.
- 6 **Revisar los requisitos:** en esta tarea, se verifican y validan los requisitos en conjunto, no individualmente, y se revisan las estimaciones de coste y tiempo.
- 7 **Reutilización de requisitos:** en esta tarea, que se realiza en paralelo con las demás, se recopila información sobre proyectos similares que pueda reutilizarse en el proyecto en desarrollo. Aunque en las metodologías propuestas en esta tesis no se ha planteado ninguna tarea específica que contemple la reutilización, los patrones-R, tanto a nivel de requisitos-C como de requisitos-D (ver sección 4.8, pág. 175) permiten abordarla en cualquier momento del proceso que se considere oportuno.

Otro modelo que también se han tenido en cuenta han sido las actividades de elicitación propuestas en [Christel y Kang 1992], en el que se definen cinco actividades principales:

- 1 **Determinación de hechos (*Fact-Finding*):** en esta actividad se deben identificar los participantes en el proceso, determinar el contexto operacional, investigar sistemas similares, etc., que equivaldría a la tarea 1 (*Obtener información sobre el dominio del problema*) de la metodología propuesta.
- 2 **Recolección y clasificación de requisitos:** en esta actividad se identifican los requisitos de los clientes y usuarios y se construyen modelos

de acuerdo a las técnicas a emplear en el desarrollo (estructuradas, orientadas a objetos, etc.). Equivaldría a las tareas de la 2 a la 6 de la metodología propuesta, incluyendo parte de las actividades de análisis de requisitos descritas en el siguiente capítulo.

- 3 Evaluación y justificación:** en esta actividad se deben evaluar la implementabilidad y coste de los requisitos y justificar su necesidad, correspondiendo con aspectos tanto de elicitación (justificación de la necesidad) como de análisis (evaluar la implementabilidad).
- 4 Priorización:** esta actividad, que coincidiría con la tarea 7 (*Priorizar objetivos y requisitos*) de la metodología propuesta, se realiza para priorizar los requisitos en función a su criticidad y su coste, identificando posibles arquitecturas que permitan un desarrollo incremental. Al igual que en la actividad anterior, existen elementos clasificados como de elicitación y de análisis en la metodología propuesta.
- 5 Integración y validación:** en esta actividad se deben validar los requisitos con clientes y usuarios y resolver los posibles conflictos, correspondiendo por tanto con las actividades de validación y de elicitación (negociación) de la propuesta.

3.8.2 Plantillas para requisitos

En relación con las plantillas presentadas, la plantilla para casos de uso de Cockburn [Cockburn 1997] es la que ha ejercido una mayor influencia, aunque también se tuvieron en cuenta las propuestas de plantillas de casos de uso de [Rumbaugh 1994], [Schneider y Winters 1998] y [Coleman 1998].

Dentro de las plantillas específicas para requisitos, se tuvieron en cuenta la del método Volere (ver figura 3.28) [Robertson y Robertson 1998, Robertson y Robertson 1999], las plantillas para requisitos basados en la norma PSS-05 de [Ruggier 1998], y las plantillas generales descritas en [Yourdon Inc. 1993].

En la figura 3.29 puede verse una comparativa entre las propuestas citadas y la realizada en esta tesis.

3.8.3 Patrones lingüísticos

La idea de los patrones-L surgió de nuestra experiencia con prácticas académicas, aunque algunos de los resultados del proyecto CREWS, en con-

The diagram shows a template for a requirement with various fields. Handwritten annotations in blue ink explain each field:

- Requirement #:** Unique Id
- Requirement Type:** The type from the template
- Event/use case #:** List of events / use cases that need this requirement
- Description:** A one sentence statement of the intention of the requirement
- Rationale:** A justification of the requirement
- Source:** Who raised this requirement?
- Fit Criterion:** A measurement of the requirement such that it is possible to test if the solution matches the original requirement
- Customer Satisfaction:** Degree of stakeholder happiness if this requirement is successfully implemented. Scale from 1 = uninterested to 5 = extremely pleased.
- Customer Dissatisfaction:** Measure of stakeholder unhappiness if this requirement is not part of the final product. Scale from 1 = hardly matters to 5 = extremely displeased.
- Dependencies:** A list of other requirements that have some dependency on this one
- Conflicts:** Other requirements that cannot be implemented if this one is
- Supporting Materials:** Pointer to documents that illustrate and explain this requirement
- History:** Creation, changes, deletions, etc.

The template includes the **Volere** logo and the text "Copyright © Atlantic Systems Guild".

Figura 3.28: Plantilla para requisitos de Volere

Campos incluidos	Cockburn	Coleman	Volere	Rumbaugh	Schneider	Propuesta
Identificador	sí	sí	sí	no	no	sí
Nombre	sí	sí	sí(a)	sí	sí	sí
Versión	no	sí(b)	sí(b)	no	no	sí
Autores	no	no	no	no	no	sí
Objetivo	sí	sí(c)	sí(d)	no	sí(e)	sí(e)
Objetivos asociados	sí(f)	no	sí(a)	no	no	sí
Descripción	sí(g)	sí	sí	sí	no	sí
Actores	sí	sí	no	no	sí(g)	sí(g)
Evento activación	sí	no	no	no	sí(g)	sí(c)
Precondición	sí	sí	no	no	sí	sí
Postcondición	sí	no	no	no	sí	sí
Pasos	sí	sí	no	no	sí	sí
Variaciones	sí	sí	no	no	no	no(h)
Excepciones	sí(i)	no	no	no	no	sí
Include/extend	sí	sí	no	sí	sí	sí
Rendimiento	sí	sí	no	no	sí	sí
Frecuencia	no	sí	no	no	no	sí
Importancia	no	no	sí(j)	no	no	sí
Prioridad	sí	sí	no	no	sí	sí(k)
Estado desarrollo	no	no	no	no	no	sí
Estabilidad	no	no	no	no	no	sí
Requisitos asociados	sí(l)	no	sí	no	sí	sí
Modelos asociados	no	no	sí	no	sí	sí(m)
Conflictos pendientes	sí	sí	sí	no	no	sí(n)
Criterio de aceptación	no	no	sí(o)	no	no	no(p)

- (a) En el caso de uso del que depende el requisito
- (b) Como historia del requisito
- (c) En la descripción
- (d) Como propósito
- (e) Como nombre descriptivo
- (f) Como jerarquía de casos de uso de distintos niveles
- (g) Implícitamente
- (h) No se contemplan explícitamente, pero pueden incluirse en los comentarios
- (i) Como relación de extensión (*extends*) entre casos de uso
- (j) Como grado de satisfacción/insatisfacción
- (k) Como urgencia
- (l) Como relación de inclusión/incluido
- (m) Se incluye en la descripción de los elementos del modelo
- (n) En la descripción de los conflictos se indican los requisitos
- (o) Se trata de un criterio muy general, no de una prueba
- (p) Los aspectos relacionados con las pruebas no se abordan en este trabajo

Figura 3.29: Comparativas de plantillas para casos de uso

creto los expuestos, entre otros, en [Rolland y Achour 1998] o en [Maiden *et al.* 1998], presentan también el concepto de *patrón lingüístico* desde un punto de vista diferente. En estos trabajos se pretende analizar los casos de uso (denominados *escenarios* en los trabajos citados) utilizando técnicas del campo del procesamiento de lenguaje natural, para obtener así estructuras semánticas que puedan analizarse para comprobar si cumplen las propiedades descritas en los, denominados por los autores, *patrones de validación* que se comentan en la siguiente sección.

Para alcanzar este objetivo se proponen unas *guías de estilo* (ver figura 3.30) sobre el estilo de redacción de los pasos de los casos de uso, y unas *guías de contenido* (ver figura 3.31) sobre el formato que deben tener las frases que describen los pasos y que son muy similares a los patrones-L descritos en las secciones anteriores.

En nuestra opinión, este enfoque presenta el inconveniente de la necesidad de *reducción* del lenguaje natural utilizado para expresar los casos de uso para que pueda ser procesado, lo que obliga a utilizar un lenguaje natural *reducido* parecido al clásico *lenguaje estructurado* propuesto para la especificación de los procesos en los diagramas de flujos de datos (DFDs) [Pressman 1997], con el consiguiente peligro de que la descripción de los casos de uso acabe siendo algo similar a un pseudocódigo que los clientes y usuarios tengan dificultades para entender, sin obtener unos beneficios significativos a cambio.

SG1	escribir la secuencia normal del caso de uso como una lista de acciones discretas de la forma <i><nº acción><descripción de la acción></i> . Cada descripción de acción debe empezar en una nueva línea. Dado que cada acción es atómica, evitar frases con más de dos cláusulas.
SG2	usar el orden secuencial de las descripciones de acciones (y por lo tanto su identificadores numéricos únicos) para indicar una secuencia estricta entre las acciones. Las variaciones deben escribirse en una sección separada
SG3	las iteraciones y las acciones concurrentes deben expresarse en la misma sección del caso de uso, mientras que las acciones alternativas deben escribirse en una sección diferente
SG4	usar nombres consistentes para los agentes (actores), objetos y acciones en todas las descripciones de casos de uso. Evitar el uso de sinónimos y homónimos así como pronombres como él, ella, ellos, ello, etc. Ser consistente en el uso de la terminología
SG5	usar el presente de indicativo y voz activa en la descripción de las acciones
SG6	evitar el uso de negaciones, adverbios y verbos modales en las descripción de las acciones

Figura 3.30: Guías de estilo CREWS

CG1	<agente><acción "mover"><objeto> de <origen> a <destino>
CG2	<agente origen><acción "poner"><objeto> a <agente destino>
CG3	<agente destino><acción "tomar"><objeto> de <agente origen>
CG4	<agente><acción><agente>
CG5	<agente><acción><objeto>
CG6	Si <condición> entonces <acción>
CG7	Mientras <condición> hacer <acción>
CG8	<acción ₁ > mientras que <acción ₂ >

Figura 3.31: Guías de contenido CREWS

No obstante, no deja de ser una posibilidad interesante que podría llegar a permitir algún día obtener automáticamente modelos, e incluso implementaciones, a partir de descripciones en lenguaje natural.

3.8.4 Patrones de reutilización

Los patrones de reutilización no son más que el fruto de la constatación de la repetición de requisitos similares en las prácticas académicas en las que se han utilizado las ideas propuestas en esta tesis. Dichas prácticas académicas tienen en común su dominio de aplicación, sistemas de información para pequeños negocios (gestión de pequeños comercios, gestión de bibliotecas, etc.), por lo que parece natural la repetición de requisitos similares.

En [Robertson y Robertson 1999, págs. 222-231] se plantea también la idea de patrones de requisitos, aunque con un formato específico similar al que se usa para los patrones de diseño [Gamma *et al.* 1995] e incluyendo los requisitos-D asociados, tal como se plantea también en la sección 4.8, pág. 175 de esta memoria, aunque utilizando técnicas estructuradas basadas en diagramas de flujo de datos.

Sobre la reutilización de requisitos en familias de aplicaciones similares existen trabajos previos como [Finkelstein 1998], que se basa más en modelos (requisitos-D) que en requisitos en lenguaje natural (requisitos-D).

En [Keepence *et al.* 1995] y [Mannion *et al.* 1999] se estudia, dentro del dominio de problemas de los sistemas de planificación de misiones de naves espaciales, la clasificación de requisitos-C en función de su reutilizabilidad como *no reutilizables*, *reutilizables directamente* (componentes) y *parametrizados* (generadores). Para hacer que los requisitos no reutilizables pasen a una de las otras dos categorías se proponen la tres siguientes heurísticas básicas:

- **Eliminar referencias específicas:** esta heurística propone, de forma no muy clara, evitar hacer referencias específicas dentro de un requisito. Por ejemplo, si se está desarrollando el sistema *X*, evitar usar frases como "*El sistema X deberá ...*" y usar en su lugar "*El sistema deberá ...*", con lo cual podrían reutilizarse requisitos similares en varios desarrollos.
- **Usar términos comunes:** esta heurística sugiere que se utilicen nombres lo más comunes posibles dentro de un determinado dominio de problemas, relegando las diferencias específicas de cada proyecto a los respectivos glosarios de términos.
- **Separar lo específico de lo genérico:** la última heurística propone detectar en un requisito qué partes son generales y reutilizables y qué partes son específicas, y separar éstas últimas en uno o más requisitos aparte. Una idea parecida al patrón *Especificar* de [Creel 2000] que se describe a continuación.

En [Lam *et al.* 1997] se exponen diez heurísticas generales sobre la introducción de la reutilización en el proceso de ingeniería de requisitos. Básicamente se propone:

- identificar familias de sistemas en los que los requisitos suelen coincidir
- desarrollar requisitos parametrizables abstractos (una idea similar a la presentada en este artículo)
- separar los aspectos específicos de los generales
- intentar identificar patrones de requisitos al trabajar en dominios específicos (es decir, patrones-R)
- intentar reutilizar también los procesos de obtención de ciertos tipos de requisitos, es decir las preguntas a realizar a los clientes y usuarios, las consideraciones a tener en cuenta a la hora de especificarlos, etc.

En la bibliografía consultada también se han hallado patrones de requisitos a un nivel de abstracción más alto. En [Creel 2000] se proponen los tres siguientes:

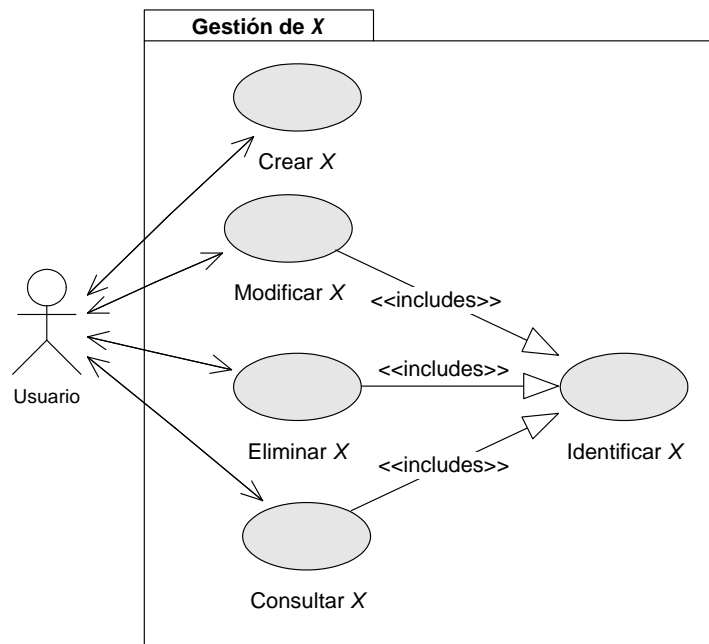


Figura 3.32: Aplicación del patrón *Especificar* en los casos de uso

- Patrón Especificar:** este patrón aconseja describir *cómo* puede el usuario de un sistema seleccionar (*especificar*) una determinada información (para modificarla, eliminarla o consultarla) en un requisito separado y hacer referencia a dicho requisito cuando sea necesario. Por ejemplo, un requisito podría establecer que "el sistema deberá permitir al usuario seleccionar a los clientes por su número de DNI, por sus apellidos o por su número de teléfono" y posteriormente otro que dijera que "el sistema deberá permitir a los usuarios modificar la información correspondientes a los clientes seleccionados mediante el procedimiento descrito en el requisito X (el requisito anterior)".

Abstrayendo y adaptando el patrón a las ideas expuestas en esta tesis, podría reformularse el patrón para que aconsejará no especificar la forma en que un actor selecciona o identifica una determinada información en los casos de uso en que fuera necesario, sino sacar dicho procedimiento *factor común* en un caso de uso abstracto e *incluirlo* en los casos de uso que fuera necesario (ver figura 3.32).

Otra posibilidad, que es la contemplada en los patrones-R de requisitos funcionales descritos en la sección 3.7.2, es no indicar *cómo* puede el usuario seleccionar o identificar una determinada información, dejando este tipo de detalles para fases posteriores del desarrollo.

- **Patrón Presentación:** este patrón recomienda limitarse a indicar qué datos debe solicitar o presentar el sistema sin entrar en detalles concretos de interfaz de usuario. En los patrones-R de casos de uso de la sección 3.7.2 puede verse un ejemplo de su aplicación.
- **Patrón Priorizar:** este otro patrón sugiere que, en el caso de que el usuario desee poder ordenar (*priorizar*) la información presentada por el sistema, se separen las posibles formas de ordenar dicha información en un requisito aparte y se referencie desde los que sea necesario, de forma similar al patrón *Especificar*.

Otros patrones de requisitos de alto nivel son los presentados como parte de los resultados del proyecto CREWS en [Maiden *et al.* 1998]. Son patrones muy generales que se proponen como estrategia de validación de requisitos y cuya descripción es la siguiente:

- **Máquina-Función:** este patrón propone tener un requisito funcional por cada acción en la que esté involucrado el sistema. Este patrón se justifica porque, dentro de los resultados del proyecto CREWS, los casos de uso o escenarios se consideran solamente como una técnica de elicitación para obtener los requisitos funcionales, no como los requisitos en si mismos. De esta forma, este patrón podría entenderse como la necesidad de que para cada caso de uso del sistema debe haber un requisito funcional que lo recoja.

En la propuesta realizada en esta tesis, se utilizan los casos de uso como técnica de elicitación y como forma de expresión de los requisitos funcionales, por lo que este patrón está implícito.

- **Recoger-Primero-Objetivo-Último:** este patrón aconseja que en las interacciones entre personas y el sistema a desarrollar se debe evitar que se dejen efectos personales en los dispositivos de interacción, por ejemplo, no proporcionando el dinero en un cajero automático hasta que el usuario no haya retirado su tarjeta de crédito.
- **Transacción-Insegura-Segura:** este patrón recomienda que el sistema se diseñe de forma que las transacciones no se puedan volver inseguras por negligencias del usuario.

En [Maiden *et al.* 1998] también se propone una clasificación de patrones en las cuatro siguientes categorías:

- **Patrones de diseño de sistemas:** capturan elementos de buenos diseños de sistemas, por ejemplo el patrón *Transacción–Insegura–Segura*.
- **Patrones de diseño de dispositivos:** describen buenos diseños de dispositivos de interacción, por ejemplo el patrón *Recoger–Primero–Objetivo–Último*.
- **Patrones de diseño de software:** son los patrones de diseño recogidos, entre otros, en [Gamma *et al.* 1995].
- **Patrones de diseño de especificaciones:** estos patrones capturan elementos de diseño de los documentos de requisitos que facilitan que los diseñadores realicen diseños satisfactorios, por ejemplo el patrón *Máquina–Función* o los anteriormente comentados de [Creel 2000].

3.9 Conclusiones

En este capítulo se ha presentado una propuesta metodológica para la elicitación de requisitos, que se incluye como apéndice de este trabajo. Se han presentado también plantillas y patrones lingüísticos para la elicitación y especificación de requisitos–C y se han indicado las posibilidades de reutilización que ofrecen dichas plantillas en forma de patrones–R_C.

Las ideas expuestas en este capítulo se ha utilizado en más de 80 prácticas académicas correspondientes a las asignaturas de *Ingeniería del Software I* e *Ingeniería del Software de Gestión I* de las titulaciones de Ingeniero en Informática e Ingeniero Técnico en Informática de Gestión impartidas en la Universidad de Sevilla. Los resultados han sido, aparte de una homogenización de los documentos resultantes, una mayor comprensión por parte de los alumnos del proceso general de la ingeniería de requisitos, una mayor facilidad para describir el sistema a desarrollar y una sustancial mejora del proceso de revisión de sus trabajos por parte de los profesores responsables.

Resultados muy similares se han obtenido en la universidades de Valladolid y Salamanca, donde también han adoptado las plantillas descritas en este trabajo para realizar las prácticas de las asignaturas de *Ingeniería*

del Software II de la Ingeniería Técnica de Informática de Gestión y de *Ingeniería del Software* de la Ingeniería Técnica de Informática de Sistemas respectivamente.

También se han utilizado en tres proyectos de Sadiel S.A.: *Reintegros por pagos indebidos a personal de la Junta de Andalucía*, *Gestión y control de subvenciones de la Junta de Andalucía* y *Gestión y control de pensiones no contributivas de la Junta de Andalucía*, que se encuentran aún en desarrollo. La aplicación de las plantillas descritas en este capítulo ha supuesto una mejora sustancial en la comunicación con los clientes y usuarios, que no han tenido dificultades en comprender su uso y que han participado activamente en la redacción y revisión de los requisitos al encontrar en las plantillas una forma fácil de hacerlo.

Dado que los tres proyectos anteriores tenían como cliente a la Junta de Andalucía, que exige la aplicación de Métrica V2.1 [MAP 1995] en los desarrollos que contrata, se ha realizado también la integración de las ideas expuestas con dicha metodología [Durán *et al.* 1999].

Aún no se han podido obtener resultados empíricos de la aplicación de los patrones de reutilización, aunque la posibilidad de reutilización de requisitos individuales (reutilización *horizontal*) es prometedora. Más prometedora aún si cabe es la reutilización *vertical* (ver figura 3.33), también denominada *downstream reuse* [D'Souza y Wills 1999, pág. 455], en la que al reutilizar un elemento se reutilizan también los elementos de nivel inferior asociados por las relaciones de rastreabilidad en forma de *mecano* [García 2000].

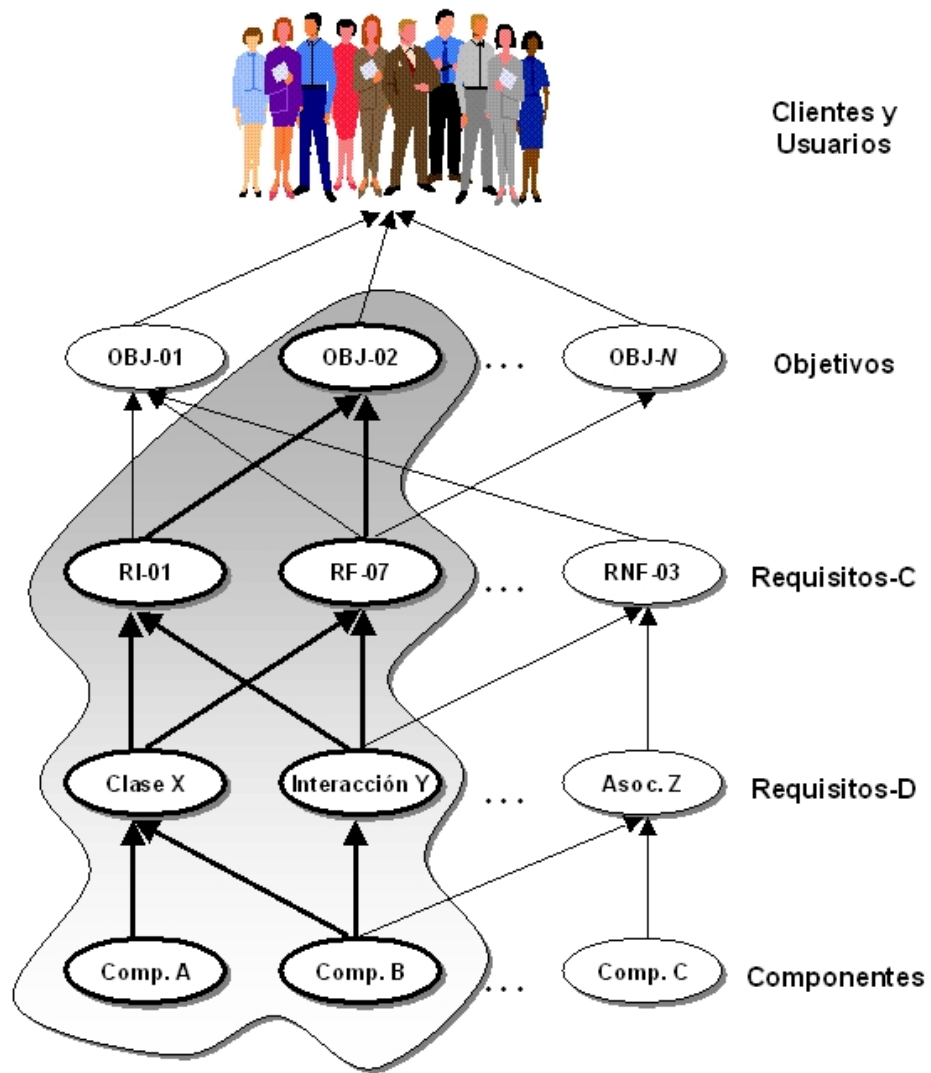


Figura 3.33: La rastreabilidad como clave para la reutilización vertical

3.10 Bibliografía

- [Beyer y Holtzblatt 1995] H. R. Beyer y K. Holtzblatt. Apprenticing with the Customer. *Communications of the ACM*, 38(5), Mayo 1995.
- [Blaha y Premerlani 1998] M. Blaha y W. Premerlani. *Object-Oriented Modeling and Design for Database Applications*. Prentice-Hall, 1998.
- [Boehm et al. 1994] B. W. Boehm, P. Bose, E. Horowitz, y M.-J. Lee. Software Requirements as Negotiated Win Conditions. En *Proceedings of the First International Conference on Requirements Engineering*, 1994. Disponible en <http://sunset.usc.edu/TechRpts/Papers/NGPM-Requirements93.ps>.
- [Booch et al. 1999] G. Booch, J. Rumbaugh, y I. Jacobson. *The Unified Modeling Language User Guide*. Addison-Wesley, 1999.
- [Brackett 1990] J. W. Brackett. Software Requirements. Curriculum Module SEI-CM-19-1.2, Software Engineering Institute, Carnegie Mellon University, 1990. Disponible en <http://www.sei.cmu.edu>.
- [Christel y Kang 1992] M. G. Christel y K. C. Kang. Issues in Requirements Elicitation. Technical Report CMU/SEI-92-TR-12, Software Engineering Institute, Carnegie Mellon University, 1992. Disponible en <http://www.sei.cmu.edu>.
- [Cockburn 1997] A. Cockburn. Structuring Use Cases with Goals. *Journal of Object-Oriented Programming*, Sept. y Nov./Dic. 1997. Disponible en <http://members.aol.com/acockburn/papers/usecases.htm>.
- [Coleman 1998] D. Coleman. A Use Case Template: Draft for Discussion. *Fusion Newsletter*, Abril 1998. Disponible en http://www.hpl.hp.com/fusion/md_newletters.html.
- [Creel 2000] C. Creel. Requirements by Pattern. *Software Development*, Diciembre 2000. Disponible en <http://www.sdmagazine.com/breakrm/features/s9912f4.shtml>.
- [Damian et al. 2000] D. E. H. Damian, A. Eberlein, M. L. G. Shaw, y B. R. Gaines. Using Different Communication Media in Requirements Negotiation. *IEEE Software*, 17(3):28-36, Mayo/Junio 2000.
- [Davis 1985] F. Davis. *La comunicación no verbal*, volumen 616 de *El Libro de Bolsillo*. Alianza Editorial, 1985.

- [Davis 1993] A. M. Davis. *Software Requirements: Objects, Functions and States*. Prentice–Hall, 2ª edición, 1993.
- [Davis 1995] A. M. Davis. *201 Principles of Software Development*. McGraw–Hill, 1995.
- [Díaz y Rodríguez 1998] O. Díaz y J. J. Rodríguez. On Use Case Elicitation. En *Actas de las III Jornadas de Ingeniería del Software*, Murcia, 1998.
- [DoD 1994] DoD. *Military Standard 498: Software Development and Documentation*. Departament of Defense of the United States of America, 1994. Disponible en http://www-library.itsi.disa.mil/mil_std/498_win3.exe.
- [Dorfman y Thayer 1990] M. Dorfman y R. H. Thayer, editores. *Standards, Guidelines, and Examples on System and Software Requirements Engineering*. IEEE Computer Society Press, 1990.
- [D’Souza y Wills 1999] D. F. D’Souza y A. C. Wills. *Objects, Components, and Frameworks with UML: The Catalysis Approach*. Addison–Wesley, 1999.
- [Durán *et al.* 1998] A. Durán, B. Bernárdez, M. Toro, y A. Ruiz. Una Propuesta Metodológica para la Elicitación de Requisitos de un Sistema Software. En *Actas de las III Jornadas de Trabajo Menhir*, Murcia, 1998. Universidad de Murcia.
- [Durán *et al.* 1999] A. Durán, B. Bernárdez, M. Toro, y R. Corchuelo. Una Propuesta para el Catálogo de Requisitos de MÉTRICA V2.1. *Novática*, (142), Noviembre–Diciembre 1999.
- [ESP 1996] ESPITI – European User Survey Results. Informe Técnico ESI–1996–TR95104, European Software Institute, 1996. Disponible en <http://www.esi.es>.
- [Fairley y Thayer 1997] R. E. Fairley y R. H. Thayer. The Concept of Operations: The Bridge from Operational Requirements to Technical Specifications. En Thayer y Dorfman [1997], páginas 73–83.
- [Finkelstein 1998] A. Finkelstein. Re–use of Formatted Requirements Specifications. *Journal of Software Engineering*, Septiembre 1998.
- [Firesmith 1997] D. G. Firesmith. Uses Cases: the Pros and Cons, 1997. Disponible en <http://www.ksscary.com/usecjrnl.html>.

- [Gamma *et al.* 1995] E. Gamma, R. Helm, R. Johnson, y J. Vlissides. *Design Patterns: Elements of Reusable Object-Oriented Software*. Addison-Wesley, 1995.
- [GAO 1979] Contracting for Computer Software Development: Serious Problems Require Management Attention to Avoid Wasting Additional Millions. Report FGMSD-80-4, U. S. Government Account Office, Noviembre 1979. Este documento es difícil de encontrar, pero está comentado y referenciado, entre otros, en [Davis 1993], [Christel y Kang 1992] y [Goguen 1994].
- [García *et al.* 2000] J. García, M. J. Ortín, B. Moros, y J. Nicolás. Modelado de Casos de Uso y Conceptual a partir del Modelado del Negocio. En *Actas de las V Jornadas de Trabajo Menhir*, Granada, 2000.
- [García 2000] F. J. García. *Modelo de Reutilización Soportado por Estructuras Complejas de Reutilización Denominadas Mecanos*. Tesis doctoral, Universidad de Salamanca, 2000.
- [Gause y Weinberg 1989] D. C. Gause y G. M. Weinberg. *Exploring Requirements: Quality Before Design*. Dorset House, 1989.
- [Goguen y Linde 1993] J. A. Goguen y C. Linde. Techniques for Requirements Elicitation. En *Proceedings of the First International Symposium on Requirements Engineering*, 1993. También aparece en [Thayer y Dorfman 1997]. Disponible en <http://www.cse.ucsd.edu/~goguen>.
- [Goguen 1994] J. A. Goguen. Requirements Engineering as the Reconciliation of Social and Technical Issues. En *Requirements Engineering: Social and Technical Issues*, páginas 165–199. Academic Press, 1994. Disponible en <http://www.cse.ucsd.edu/~goguen>.
- [Goleman 1996] D. Goleman. *La Inteligencia Emocional*. Kairós, 1996.
- [Goleman 1999] D. Goleman. *La Práctica de la Inteligencia Emocional*. Kairós, 1999.
- [IEEE 1993] IEEE. IEEE Recommended Practice for Software Requirements Specifications. IEEE/ANSI Standard 830-1993, Institute of Electrical and Electronics Engineers, 1993.
- [Iturrioz 1998] J. I. Iturrioz. *Una Metodología para el Desarrollo de Sistemas de Bases de Datos Objeto-Relacionales*. Tesis doctoral, Universidad del País Vasco, 1998.

- [Jacobson *et al.* 1993] I. Jacobson, M. Christerson, P. Jonsson, y G. Övergaard. *Object-Oriented Software Engineering: A Use Case Driven Approach*. Addison-Wesley, 4ª edición, 1993.
- [Jacobson *et al.* 1997] I. Jacobson, M. Griss, y P. Jonsson. *Software Reuse: Architecture, Process and Organization for Business Success*. Addison-Wesley, 1997.
- [Jarke 1998] M. Jarke. Requirements Tracing. *Communications of the ACM*, 41(12), Diciembre 1998.
- [Keepence *et al.* 1995] B. Keepence, M. Mannion, y S. Smith. SMARTRe Requirements: Writing Reusable Requirements. En *Proceedings of the IEEE Symposium on Engineering of Computer-Based Systems*, 1995.
- [Kontoya y Sommerville 1996] G. Kontoya y I. Sommerville. Requirements Engineering with Viewpoints. *Software Engineering Journal*, 11(1), Enero 1996.
- [Laguna *et al.* 1999] M. A. Laguna, J. M. Marqués, y F. J. García. Una Herramienta para la Captura de Requisitos de Usuario. En *Actas de las JISBD'99*, Cáceres, 1999.
- [Lam *et al.* 1997] W. Lam, J. A. McDermid, y A. J. Vickers. Ten Steps Towards Systematic Requirements Reuse. *Requirements Engineering Journal*, 2:102–113, 1997.
- [Lilly 2000] S. Lilly. How to Avoid Use-Case Pitfalls. *Software Development*, Enero 2000. Disponible en <http://www.sdmagazine.com/breakrm/features/s0001f4.shtml>.
- [Macaulay 1998] L. A. Macaulay. The Role of the Facilitator in Requirements Engineering. En *Fifth International Conference on Requirements Engineering*, 1998.
- [Macaulay 1999] L. A. Macaulay. Seven-Layer Model of the Role of the Facilitator in Requirements Engineering. *Requirements Engineering Journal*, 4(1), 1999.
- [Maiden *et al.* 1998] N. A. Maiden, M. Cisse, H. Perez, y D. Manuel. CREWS Validation Frames: Patterns for Validating Systems Requirements. En *Fourth International Workshop on Requirements Engineering: Foundation for Software Quality (RESFQ)*, 1998. Disponible en <http://sunsite.informatik.rwth-aachen.de/CREWS/reports98.htm>.

- [Mannion *et al.* 1999] M. Mannion, H. Kaindl, y J. Wheadon. Reusing Single System Requirements from Application Family Requirements. En *Proceedings of the International Conference on Software Engineering*, 1999.
- [Manso *et al.* 1999] M. E. Manso, M. P. Romay, y F. J. Peñalvo. Repository Asset Audit. En *Actas de las IV Jornadas de Trabajo Menhir*, Sedano (Burgos), 1999. Universidad de Valladolid.
- [MAP 1995] MAP. *Metodología de Planificación y Desarrollo de Sistemas de Información. MÉTRICA Versión 2.1*. Tecnos/Ministerio para las Administraciones Públicas, 1995.
- [Mazza *et al.* 1994] C. Mazza, J. Fairclough, B. Melton, D. de Pablo, A. Scheffer, y R. Stevens. *Software Engineering Standards*. Prentice-Hall, 1994. Disponible en <http://dxsting.cern.ch/sting/ESA.txt>.
- [Nuseibeh *et al.* 1994] B. Nuseibeh, J. Kramer, y A. Finkelstein. A Framework for Expressing the Relationships between Multiple Views in Requirements Specification. *IEEE Transactions on Software Engineering*, 20(10), Octubre 1994.
- [OOTC 1996] IBM OOTC. *Developing Object-Oriented Software: An Experience-Based Approach*. Prentice-Hall, 1996.
- [Parets-Llorca y Grünbacher 1999] J. Parets-Llorca y P. Grünbacher. Capturing, Negotiating, and Evolving System Requirements: Bridging Win Win and the UML. En *Proceedings of 25th EUROMICRO Conference*, Milan, 1999.
- [Piattini *et al.* 1996] M. G. Piattini, J. A. Calvo-Manzano, J. Cervera, y L. Fernández. *Análisis y Diseño Detallado de Aplicaciones Informáticas de Gestión*. ra-ma, 1996.
- [Pohl 1994] K. Pohl. The Three Dimensions of Requirements Engineering: A Framework and its Application. *Information Systems*, 3(19), Junio 1994.
- [Pohl 1997] K. Pohl. Requirements Engineering: An Overview. *Encyclopedia of Computer Science and Technology*, 36, 1997. Disponible en <http://sunsite.informatik.rwth-aachen.de/CREWS/reports96.htm>.
- [Potts *et al.* 1994] C. Potts, K. Takahashi, y A. Anton. Inquiry-Based Requirements Analysis. *IEEE Software*, 11(2), 1994.

- [Pressman 1997] R. S. Pressman. *Ingeniería del Software. Un Enfoque Práctico*. McGraw-Hill, 4ª edición, 1997.
- [Raghavan *et al.* 1994] S. Raghavan, G. Zelesnik, y G. Ford. Lecture Notes on Requirements Elicitation. Educational Materials CMU/SEI-94-EM-10, Software Engineering Institute, Carnegie Mellon University, 1994. Disponible en <http://www.sei.cmu.edu>.
- [Robertson y Robertson 1998] J. Robertson y S. Robertson. Volere Requirements Specification Template Edition 6.0. Informe técnico, Atlantic Systems Guild, 1998. Disponible en <http://www.atlsysguild.com/GuildSite/Robs/Templsects.html>.
- [Robertson y Robertson 1999] S. Robertson y J. Robertson. *Mastering the Requirement Process*. Addison-Wesley, 1999.
- [Rolland *et al.* 1998] C. Rolland, C. Ben Achour, C. Cauvet, J. Ralyté, A. Sutcliffe, N. A. M. Maiden, M. Jarke, P. Haumer, K. Pohl, E. Dubois, y P. Heymans. A Proposal for a Scenario Classification Framework. *Requirements Engineering Journal*, 3(1), 1998. Disponible en <http://sunsite.informatik.rwth-aachen.de/CREWS/reports98.htm>.
- [Rolland y Achour 1998] C. Rolland y C. Ben Achour. Guiding the Construction of Textual Use Case Specifications. *Data & Knowledge Engineering Journal*, 25(1-2), 1998. Disponible en <http://sunsite.informatik.rwth-aachen.de/CREWS/reports98.htm>.
- [Rombach 1990] H. D. Rombach. Software Specifications: A Framework. Curriculum Module SEI-CM-11-2.1, Software Engineering Institute, Carnegie Mellon University, 1990. No está disponible en <http://www.sei.cmu.edu>, aunque aparece en [Dorfman y Thayer 1990].
- [Ruggier 1998] M. Ruggier. *PSS-05 User Requirements Document Template*, 1998. Disponible en <http://framemaker.cern.ch/pss05>.
- [Rumbaugh 1994] J. Rumbaugh. Getting Started: Using Use Cases to Capture Requirements. *Journal of Object-Oriented Programming*, Septiembre 1994.

- [Sawyer *et al.* 1997] P. Sawyer, I. Sommerville, y S. Viller. Requirements Process Improvement through The Phased Introduction of Good Practice. *Software Process – Improvement and Practice*, 3(1), 1997. Disponible en <http://www.comp.lancs.ac.uk/computing/research/cseg/reaims/publications.html>.
- [Sawyer y Kontoya 1999] P. Sawyer y G. Kontoya. SWEBOK: Software Requirements Engineering Knowledge Area Description. Informe Técnico Versión 0.5, SWEBOK Project, 1999. Disponible en <http://www.swebok.org>.
- [Schneider y Winters 1998] G. Schneider y J. P. Winters. *Applying Use Cases: a Practical Guide*. Addison–Wesley, 1998.
- [Thayer y Dorfman 1990] R. H. Thayer y M. Dorfman, editores. *System and Software Requirements Engineering*. IEEE Computer Society Press, 1990.
- [Thayer y Dorfman 1997] R. H. Thayer y M. Dorfman, editores. *Software Requirements Engineering*. IEEE Computer Society Press, 2ª edición, 1997. Es la 2ª edición de [Thayer y Dorfman 1990].
- [TSG 1995] TSG. *The CHAOS Report*. The Standish Group, 1995. Disponible en <http://www.standishgroup.com/chaos.html>.
- [Weidenhaput *et al.* 1998] K. Weidenhaput, K. Pohl, M. Jarke, y P. Haumer. Scenarios in System Development: Current Practice. *IEEE Software*, 15(2):34–45, Marzo/Abril 1998. Este artículo aparece también en las actas del ICRE'98 y está disponible en <http://sunsite.informatik.rwth-aachen.de/CREWS/reports97.htm>.
- [Wirfs-Brock *et al.* 1990] R. Wirfs-Brock, B. Wilkerson, y L. Wiener. *Designing Object–Oriented Software*. Prentice–Hall, 1990.
- [Yourdon Inc. 1993] Yourdon Inc. *Yourdon Systems Method: Model–Driven Systems Development*. Prentice–Hall, 1993.

Capítulo 4

Análisis de Requisitos

4.1 Introducción

En este capítulo se estudia la segunda actividad del modelo de procesos de ingeniería de requisitos propuesto en el capítulo 2: el *análisis de requisitos*.

Como ya se comentó en la sección 3.2 del capítulo anterior, el término *análisis de requisitos* se ha utilizado durante bastante tiempo para referirse al conjunto global de las actividades relacionadas con los requisitos en el desarrollo de software, al asumirse que eran proporcionados directamente por el cliente, por lo que no era labor del equipo de desarrollo la elicitación de dichos requisitos ni había necesidad de una validación por parte del cliente, ya que era él mismo el que los había producido.

En este trabajo, el análisis de requisitos se considera como una actividad dentro de la fase de ingeniería de requisitos cuyo objetivo principal es descubrir conflictos en los requisitos-C elicitados previamente, profundizando en el conocimiento del problema mediante la construcción de modelos conceptuales (*requisitos-D* en la terminología de [Rombach 1990, Brackett 1990]) que sean más fácilmente entendibles por los desarrolladores y que puedan servir de base en la fase de diseño, avanzando de esta forma en las dimensiones de compleción y formalidad del proceso descritas en [Pohl 1994, Pohl 1997] y comentadas en la sección 1.2.1 (ver figura 1.4, pág. 11).

Las técnicas de modelado conceptual son la principal herramienta del análisis de requisitos. Tradicionalmente se han considerado las técnicas estructuradas y las orientadas a objetos como dos formas incompatibles de modelar sistemas de información. En este capítulo se analizarán ambos

tipos de técnicas y se mostrará que realmente las diferencias no son tantas como habitualmente se supone.

La estructura del capítulo es la siguiente: en la sección 4.2 se exponen los conceptos básicos sobre modelado de sistemas de información. En las secciones 4.3 y 4.4 se comentan las características de las técnicas de modelado estructuradas y orientadas a objetos respectivamente. En la sección 4.6 se describe la propuesta metodológica para el análisis de requisitos y, por último, en la sección 4.9 se exponen las conclusiones.

4.2 Modelado de sistemas de información

Como se ha comentado en la introducción de este capítulo, la técnica principal para el análisis de requisitos es el modelado conceptual. Para modelar sistemas de información es necesario al menos representar tres aspectos¹ de un sistema [Yourdon Inc. 1993, Piattini *et al.* 1996]:

- los aspectos *estáticos*, que corresponden al estado del sistema representado por los *datos* sobre la información que gestiona,
- los aspectos *funcionales*, que corresponden a las *funciones* o transformaciones que realiza sobre dicha información, y
- los aspectos *dinámicos*, correspondientes a los *eventos* o estímulos que provocan que el sistema realice las funciones mencionadas.

En la figura 4.1 pueden verse esquemáticamente las tres dimensiones correspondientes a los aspectos descritos.

Los requisitos-C describen el sistema como una *caja negra*, principalmente mediante los casos de uso [Jacobson *et al.* 1993], tal como se expuso en el capítulo 3. Los requisitos-D, es decir, los modelos conceptuales que se realizan durante las actividades de análisis, no describen el sistema como si fuera una *caja negra*, sino más bien como una *caja gris*.

Esto es así debido a que para realizar una descripción más detallada de la conducta de un sistema normalmente es necesario describir su estado

¹En [Yourdon Inc. 1993] se denominan a los tres aspectos *viewpoints*. En este trabajo no se ha utilizado este término debido a las distintas acepciones que existen de él en la bibliografía consultada, por ejemplo: [Leite y Freeman 1991], [Kontoya y Sommerville 1996] o [Nuseibeh *et al.* 1994].

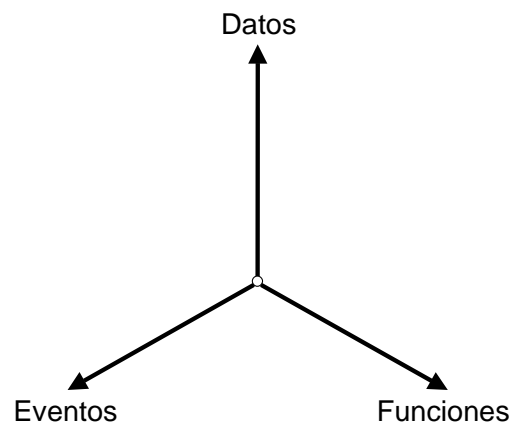


Figura 4.1: Dimensiones del modelado de sistemas de información

[D'Souza 1996]. El estado se describe mediante el modelado de la información que maneja, aspecto que no es directamente observable desde fuera del sistema. No es exactamente una *caja blanca* porque los mecanismos internos que hacen evolucionar los datos se dejan, o al menos deberían dejarse, abiertos para que se definan en la fase de diseño. Por eso en la figura 4.2, en el modelo de caja gris, el interior del sistema se ve a través de una pared translúcida que permite ver el estado del sistema pero no los mecanismos que lo hacen evolucionar.

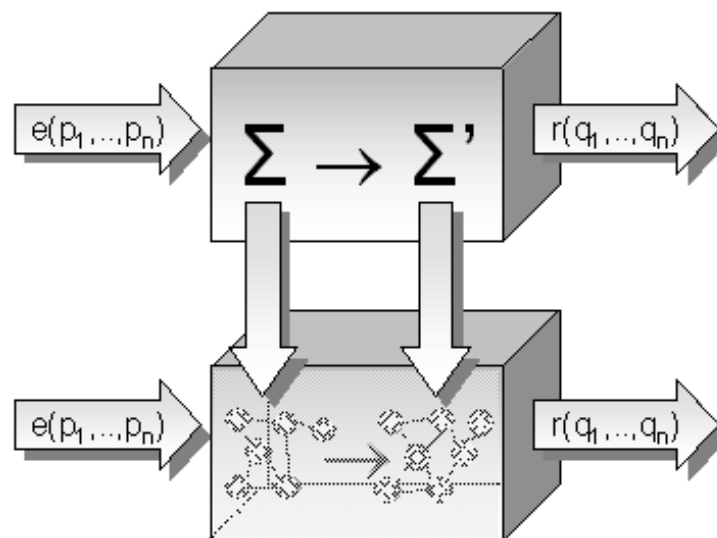


Figura 4.2: Modelos de caja negra (requisitos-C) y de caja gris (requisitos-D)

En el modelado de sistemas de información, el sistema puede considerarse como un objeto compuesto que reacciona ante eventos o estímulos externos y que genera eventos o respuestas hacia su entorno, tal como se describe, por ejemplo, en [Coleman *et al.* 1994, pág. 25], en [Cook y Daniels 1994, pág. 134] o en [Iturrioz 1998].

Los eventos generados son una función de los eventos de entrada y del estado abstracto, Σ , del sistema. A la vez que genera una respuesta hacia su entorno, el sistema experimenta una transición hacia otro estado Σ' . En el caso de las consultas, Σ y Σ' coinciden.

Durante la elicitación, el estado abstracto del sistema y las transiciones eran implícitas. En análisis, el estado abstracto y las transiciones se hacen explícitas.

4.2.1 Rastreabilidad entre requisitos-C y requisitos-D

Es obvio que si los requisitos-D modelan un sistema que está definido por los requisitos-C, debe haber una relación de rastreabilidad entre ambos tipos de requisitos.

Normalmente, el estado abstracto Σ sirve para recordar hechos ocurridos en el entorno del sistema que son relevantes para los objetivos de la organización para la que se desarrolla el sistema. Por lo tanto, su estructura viene marcada por los requisitos-C de almacenamiento de información identificados durante las actividades de elicitación de requisitos, por lo que aquellos requisitos-D (elementos del modelo) que representen el estado del sistema deberán ser rastreables desde los requisitos-C de almacenamiento de información correspondientes.

Las transiciones del sistema y sus respuestas a los estímulos externos deben corresponder con la conducta definida en los requisitos-C funcionales, por lo que los requisitos-D que modelen los aspectos dinámicos y funcionales deberán estar relacionados con dichos requisitos-C.

Los requisitos-C no funcionales no suelen tenerse en cuenta en las técnicas de modelado actuales, por lo que establecer relaciones de rastreabilidad desde estos requisitos hasta los requisitos-D no es sencillo. Lo habitual es establecer relaciones de rastreabilidad desde los requisitos-C no funcionales hacia elementos de diseño, sobre todo diseño arquitectónico, en los que se tengan en cuenta las características impuestas por este tipo de requisitos-C [Chung *et al.* 1999, Franch 1998, Ruiz *et al.* 2000].

En cualquier caso, es fundamental mantener las relaciones de rastreabilidad entre los requisitos-C y requisitos-D para una correcta gestión, tanto de los requisitos como del proyecto de desarrollo [Davis 1995, Sawyer y Kontoya 1999] y para favorecer la reutilización [García 2000].

4.3 Técnicas estructuradas de modelado

Las técnicas estructuradas de modelado han sido las predominantes en el desarrollo de sistemas de información durante los últimos 30 años y aún juegan un importante papel, sobre todo en metodologías oficiales europeas como la española Métrica [MAP 1995a] o la británica SSADM [CCT 1990], y también en propuestas no oficiales pero ampliamente seguidas como el *Yourdon System Method* [Yourdon Inc. 1993].

Los métodos estructurados han ido evolucionando en el tiempo, de hecho pueden establecerse tres generaciones [Yourdon Inc. 1993, págs. 3–11]:

- **Métodos de primera generación:** corresponden a las técnicas desarrolladas a finales de los 60 y los 70. Estas técnicas incluyen la *programación estructurada*, el *diseño modular* y los *diagramas de estructura de cuadros* de Constantine [Yourdon y Constantine 1979], y como única técnica de modelado el *diagrama de flujo de datos* (DFD) y la estrategia de descomposición funcional descendente (*top-down*).
- **Métodos de segunda generación:** en estos métodos se utilizan otras técnicas además de los DFDs para modelar distintos aspectos de los sistemas, aunque la integración entre los modelos no está bien definida. Las técnicas de modelado tienden a ser más abstractas, aunque hay una clara diferencia entre las *dirigidas por los datos*, que tienen en los *diagramas entidad-interrelación* su principal técnica de modelado y que se ven potenciadas por la disponibilidad de sistemas de gestión de bases de datos, y las *dirigidas por los procesos*, que continúan con los DFDs como herramienta principal. Comienzan a utilizarse las herramientas CASE (*Computer Aided Software Engineering*).
- **Métodos de tercera generación:** la integración entre las distintas técnicas de modelado es mayor y tienen una base teórica más sólida. El enfoque *top-down* empieza a reemplazarse por el *particionamiento de eventos* [Yourdon Inc. 1993, págs. 491–517]. La necesidad de herramientas CASE se hace evidente. Algunos ejemplos de métodos de esta generación son [MAP 1995a] o [Yourdon Inc. 1993].

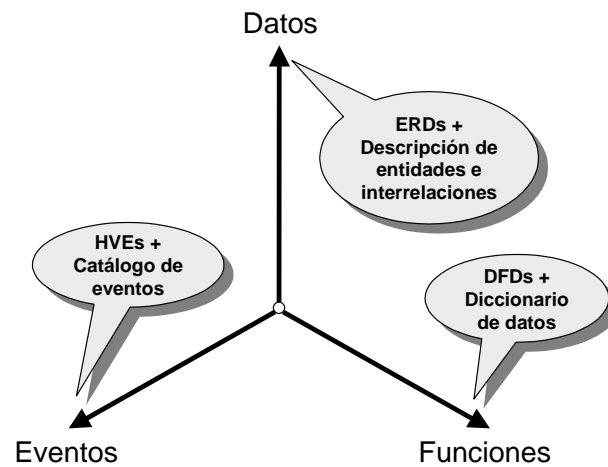


Figura 4.3: Técnicas de modelado estructuradas

En la figura 4.3 se puede ver esquemáticamente las técnicas estructuradas más empleadas para modelar cada una de las dimensiones de un sistema en los métodos estructurados de tercera generación, que son los que se tendrán en cuenta en este trabajo.

4.3.1 Modelado estático

La técnica más habitual para el modelado estático dentro de las metodologías estructuradas son los *diagramas entidad-interrelación* (ERDs, *Entity-Relationship Diagrams*) descritos originalmente en [Chen 1976] y con varias propuestas de variaciones posteriores (por ejemplo [Yourdon Inc. 1993], [Nijssen y Halpin 1989], etc.).

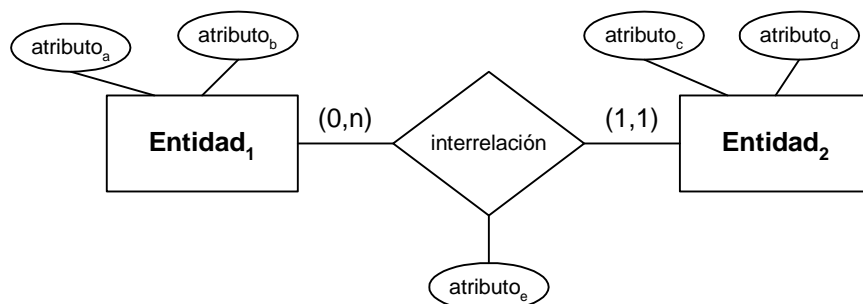


Figura 4.4: Ejemplo de diagrama entidad-interrelación

Los conceptos básicos de los ERDs son las *entidades*, que representan los conceptos relevantes sobre los que el sistema debe almacenar información, y las *interrelaciones* entre dichas entidades (ver figura 4.4).

Las entidades poseen atributos que las caracterizan. Las interrelaciones, que también pueden tener atributos, llevan asociadas cardinalidades que indican el número mínimo y máximo de instancias de entidades que pueden participar en dichas interrelaciones. En el ejemplo de la figura 4.4, las cardinalidades de la interrelación indican que cada instancia de la *Entidad₁* deberá relacionarse con una y sólo una instancia de la *Entidad₂*, mientras que cada instancia de la *Entidad₂* podrá relacionarse con cualquier número de instancias de la *Entidad₁*, incluyendo la posibilidad de que no se relacione con ninguna.

Aparte de los diagramas, son necesarias también descripciones textuales de la entidades, de sus atributos y de las interrelaciones. Un ejemplo de descripciones de este tipo puede consultarse en [Yourdon Inc. 1993].

Para una definición más completa de estos conceptos, así como otras representaciones gráficas, pueden consultarse, entre otros [Piatinni *et al.* 1996] o [Wieringa 1996].

4.3.2 Modelado funcional

El modelado funcional es el que tradicionalmente ha tenido más influencia en el análisis estructurado. La técnica por excelencia en este caso es el *diagrama de flujo de datos* o DFD [Yourdon 1989] y las correspondientes descripciones textuales de sus elementos organizadas en lo que suele denominarse *diccionario de datos*.

En los DFDs los elementos básicos son los procesos, los flujos de datos entre los procesos, los almacenes de datos y las entidades externas (ver figura 4.5).

Un concepto básico en los DFDs es la *descomposición funcional*, de forma que un proceso puede descomponerse en dos o más procesos de nivel inferior, permitiendo así manejar la complejidad de los modelos. En el nivel superior de la jerarquía de procesos está el proceso correspondiente a toda la funcionalidad del sistema y que define los límites del mismo, el cual se representa en el denominado *diagrama de contexto*.

Los procesos de los DFDs representan transformaciones de información, de forma que los flujos de salida son siempre una función de los flujos de entrada. Estos flujos de entrada pueden provenir de otros pro-

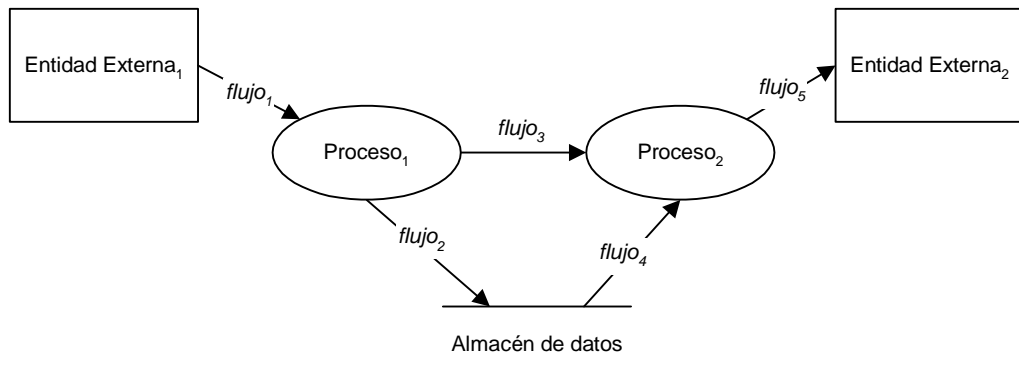


Figura 4.5: Ejemplo de diagrama de flujo de datos

cesos, de entidades externas o de almacenes de datos, que representan el estado del sistema (el Σ de la figura 4.2).

Para la descripción de los procesos existen diversas técnicas que van desde el lenguaje natural hasta el pseudocódigo pasando por lenguaje estructurado, tablas de decisión, etc. Una de las más populares son las *minispecs* descritas en [Yourdon Inc. 1993], en las que se incluyen pre y postcondiciones en un lenguaje estructurado similar al lenguaje natural.

Es importante destacar que los DFDs representan transformaciones de información, por lo que no especifican *cuándo* tienen lugar dichas transformaciones. Existen algunas propuestas para incorporar aspectos dinámicos a los DFDs en forma de flujos y procesos de control, por ejemplo [Ward y Mellor 1985, Yourdon Inc. 1993], cuyo objetivo final es incorporar en un sólo tipo de diagrama las tres dimensiones que se describieron en la sección 4.2. Sin embargo, en el modelado de sistemas de información, habitual es utilizar los DFDs sin este tipo de extensiones.

4.3.3 Modelado dinámico

Los aspectos dinámicos en las técnicas estructuradas han sido tradicionalmente los que menos atención han recibido. En general, se puede hablar de dos técnicas básicas: el *diagrama de transición de estados* (DTE) y, en los métodos más modernos, la *historia de vida de entidad* (HVE) basada en los diagramas de Jackson (en [Wieringa 1998] se puede ver una excelente comparación de estos métodos).

A ambas técnicas se les acompaña con la descripción textual de los eventos, normalmente en forma de *catálogo de eventos* [MAP 1995a].

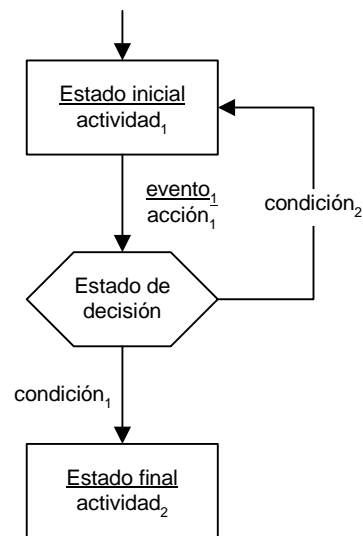


Figura 4.6: Ejemplo de diagrama de transición de estados

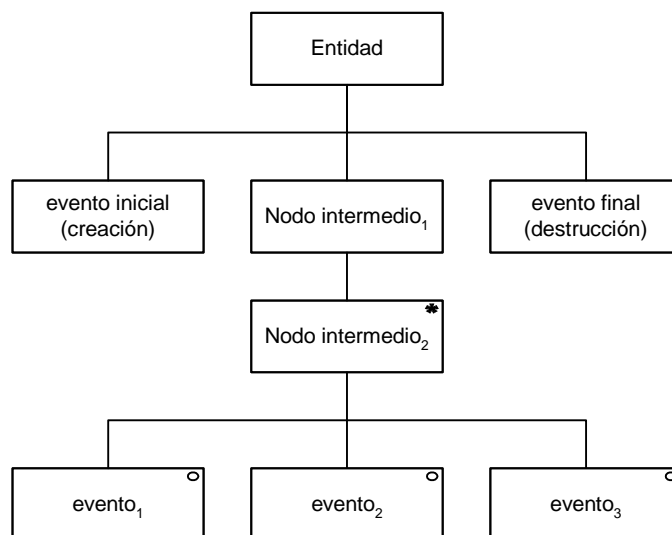


Figura 4.7: Ejemplo de diagrama de historia de vida de entidad

Los DTEs se suelen utilizar para representar la respuesta del sistema, o de una parte del sistema, ante un determinado conjunto de eventos, y suelen utilizarse rara vez para describir sistemas de información. Los elementos principales, descritos con detalle en [Wieringa 1996], son los estados (algunos de los cuales pueden ser *estados de decisión*), los eventos que provocan las transiciones, las acciones asociadas a las transiciones y las actividades asociadas a los estados (ver figura 4.6).

Los diagramas HVE [Wieringa 1996, MAP 1995a] son semánticamente equivalentes a los DTEs pero suelen usarse para describir aspectos dinámicos a nivel de instancias de entidades del modelo de datos en lugar de a nivel del sistema completo.

Los diagramas HVE (ver figura 4.7) muestran la relación temporal entre los eventos que pueden afectar al estado de las instancias de una entidad del modelo de datos. Dichos eventos pueden componerse de forma secuencial (indicado por el orden de izquierda a derecha), iterativa (indicada por un asterisco), alternativa (indicada por un círculo) o entrelazada (indicada por una barra transversal gruesa).

Los HVEs están ligados al *particionamiento de eventos* descrito, entre otros, en [Yourdon Inc. 1993] y [Wieringa 1996]. El particionamiento de eventos propone una forma de obtener los procesos de los DFDs sin seguir estrictamente un enfoque *top-down* de descomposición funcional².

Para ello, propone primero realizar el diagrama de contexto, para lo cual es necesario identificar las entidades externas y los flujos de datos entre el sistema y dichas entidades.

A continuación, en lugar de ir descomponiendo el proceso que representa al sistema en el diagrama de contexto en otros procesos, se identifican los eventos que provocan el intercambio de información entre las entidades externas y el sistema y se crean DFDs parciales en los que se muestra como responde el sistema para cada evento identificado.

Una vez contruidos todos los DFDs parciales, se agrupan en niveles hasta llegar al diagrama de contexto, es decir, se aplica un enfoque *bottom-up*.

²En [Yourdon Inc. 1993, págs. 497–499] se realiza una interesante comparación sobre las ventajas e inconvenientes del uso de la descomposición funcional descendiente (TDFD, *Top-Down Functional Decomposition*) y el particionamiento de eventos para construir el modelo funcional de un sistema utilizando DFDs.

4.3.4 Integración de modelos

Aunque pueda parecer sorprendente, la preocupación por la integración de las distintas técnicas de modelado estructurado es relativamente reciente. Tal como se comenta en [Wieringa 1996], no es hasta principios de los 80 cuando se empieza a proponer la integración de los ERDs con los DFDs. En los métodos de tercera generación, la integración entre las tres dimensiones de modelado se considera un aspecto fundamental.

Asumiendo las tres dimensiones de modelado de la figura 4.1, para estudiar la integración de los modelos asociados los ejes es necesario estudiar los planos *datos/eventos*, *eventos/funciones* y *funciones/datos* [Piattini *et al.* 1996].

4.3.4.1 Plano datos/eventos

El plano *datos/eventos* indica qué secuencia de eventos acepta una determinada entidad del modelo de datos según su estado. La integración de ambas dimensiones se produce al utilizar la técnica HVE, comentada anteriormente, por lo que se puede considerar la integración de ambas dimensiones como satisfactoria en las técnicas estructuradas de modelado.

4.3.4.2 Plano eventos/funciones

El plano *eventos/funciones* indica qué funciones realiza el sistema como respuesta a los eventos del entorno. Como ya se comentó, el *particionamiento de eventos* es el enfoque que ofrece una mayor integración entre ambas dimensiones, al ir identificando los procesos (funciones) como respuesta a los eventos.

Siguiendo la técnica del particionamiento de eventos, cuando se produzca un evento al que el sistema tenga que responder, se realizará el proceso, normalmente primitivo, de los DFDs asociado a dicho evento en el *catálogo de eventos*. En dicho catálogo se describen los eventos a los que debe responder el sistema y se indican los procesos que deben realizarse como respuesta a cada evento.

Normalmente la relación entre eventos y procesos es 1:1, aunque puede haber situaciones en la que un mismo proceso se realice como respuesta a más de un evento³.

³También podría darse el caso de que la relación eventos/procesos fuera 1:*n* si se estuvieran modelando sistemas concurrentes, aunque no es lo habitual en el caso de los sistemas de información.

4.3.4.3 Plano funciones/datos

El plano *funciones/datos* representa el efecto que tiene la realización de las funciones de los DFDs sobre las entidades e interrelaciones del modelo de datos.

En principio, esta relación entre modelos se indica de forma textual en las especificaciones de los procesos de los DFDs, y de forma gráfica como los accesos a los almacenes de datos que realizan dichos procesos.

Otra técnica que permite expresar la relación entre ambas dimensiones del modelo son las matrices proceso/entidad [MAP 1995a, Yourdon Inc. 1993], en las que se expresa el efecto, *creación, modificación, eliminación o lectura*, que tiene la realización de un proceso sobre cada entidad del modelo de datos. Estas matrices permiten comprobar que no existen entidades que no se ven afectadas por ningún proceso (y que por tanto serían innecesarias), y que no existen procesos que no provocan ningún efecto sobre ninguna entidad.

Los problemas en la integración de ambas dimensiones de modelado surgen al analizar la relación entre las entidades e interrelaciones de los ERDs y los almacenes de datos de los DFDs. Es obvio que, dado que los ERDs describen la estructura del estado del sistema y los almacenes de datos de los DFDs almacenan dicho estado del sistema, debe haber una relación entre ambos.

Los primeros problemas surgen al tener dos descripciones de los mismos conceptos, una en la descripción textual que acompaña al modelo de datos y otra al describir los almacenes de datos en el diccionario de datos asociado a los DFDs. Si ambos están relacionados, una descripción debería hacer uso de la otra, pero no ser redundantes.

El principal problema es que los almacenes de datos de los DFDs imponen de forma implícita una *relacionalización* del modelo de datos, es decir, asumen una estructura de almacenamiento de datos siguiendo el modelo relacional de Codd [Codd 1970].

Esta *relacionalización* viene provocada porque, en teoría, los almacenes de datos se supone que contienen instancias de las entidades y de las interrelaciones de cardinalidad $m:n$ del modelo de datos, tal como se expone en [Wieringa 1996, pág. 202], [MAP 1995b, págs. 90–91], [Semmens y Allen 1991] o [Pollack *et al.* 1993].

Esta suposición, que es la única que teóricamente permite *dividir* el estado del sistema en varios almacenes de datos, implica una estructura sub-

yacente muy similar al resultado de aplicar a los ERDs las heurísticas para la obtención de esquemas relacionales [Date 1995] o a un supuesto sistema de ficheros COBOL. De hecho, el propio nombre de *almacén de datos* implica aspectos de implementación de la persistencia del estado del sistema que, como ya se discutió en el capítulo 1, deben evitarse.

La única opción para evitar estos problemas y conseguir una integración plena de ambas técnicas es utilizar DFDs *degenerados* [Wieringa 1996, pág. 205] en los que haya un único almacén de datos que contenga todas las instancias de entidades e interrelaciones, es decir, todo el estado del sistema.

Dado que esta opción no es la habitual, la integración de ambos modelos no puede considerarse satisfactoria en las técnicas estructuradas. De hecho, no hace sino reflejar la tradicional división entre los partidarios de los métodos *conducidos por los datos* y los métodos *conducidos por los procesos* [D'Souza 1996].

4.4 Técnicas orientadas a objetos de modelado

Las técnicas orientadas a objetos de modelado han recibido especial atención en la última década. La evolución que han experimentado ha llevado incluso a establecer *generaciones* dentro de los métodos propuestos.

- **Métodos de primera generación:** se desarrollan principalmente en la primera mitad de los 90, algunos como una evolución de las técnicas de modelado de datos, por ejemplo OOA (*Object-Oriented Analysis*) [Coad y Yourdon 1990] u OMT (*Object Modeling Technique*) [Rumbaugh *et al.* 1991], otros basados en aspectos funcionales como OOSE (*Object-Oriented Software Engineering*) [Jacobson *et al.* 1993] y otros basados en conceptos cercanos a la programación orientada a objetos como el método de Booch [Booch 1991, Booch 1994].
- **Métodos de segunda generación:** son técnicas que se han basado en las anteriores y que se han popularizado durante la segunda mitad de la década de los 90, como *Fusion* [Coleman *et al.* 1994], *Syntropy* [Cook y Daniels 1994], *UML* [Booch *et al.* 1999] o *Catalysis* [D'Souza y Wills 1999], aunque esta última también se denomina de tercera generación al estar basada en las anteriormente citadas *Fusion*, *Syntropy* y *UML*.

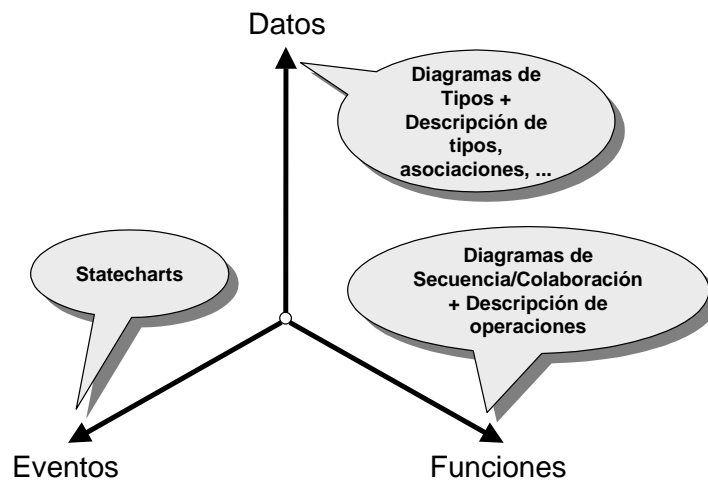


Figura 4.8: Técnicas de modelado orientadas a objetos

En esta sección se estudiarán brevemente las principales características de las técnicas orientadas a objetos de modelado que pueden verse en la figura 4.8 asociadas a las dimensiones descritas en la sección 4.2.

4.4.1 Modelado estático

La técnica utilizada por todas las propuestas orientadas a objetos para modelar el estado del sistema es, con pequeñas variaciones en cada caso particular, el *diagrama de clases* (ver figura 4.9), también denominado *diagrama de tipos* en las propuestas más recientes [Cook y Daniels 1994, D'Souza y Wills 1999]⁴.

Los elementos básicos de estos diagramas son los *tipos* de objetos, un concepto muy similar al de entidad en las técnicas estructuradas. Estos tipos pueden tener atributos y participar en asociaciones con otros tipos de objetos (al igual que las entidades de los ERDs tienen interrelaciones). A las instancias de los tipos se les denomina *objetos* y a las instancias de las asociaciones *enlaces*.

⁴En este trabajo se utilizará el término *tipo* en el mismo sentido que se utiliza en [Cook y Daniels 1994] o en [D'Souza y Wills 1999], dejando el término *clase* para los aspectos relacionados con la implementación de los tipos identificados durante el análisis de requisitos.

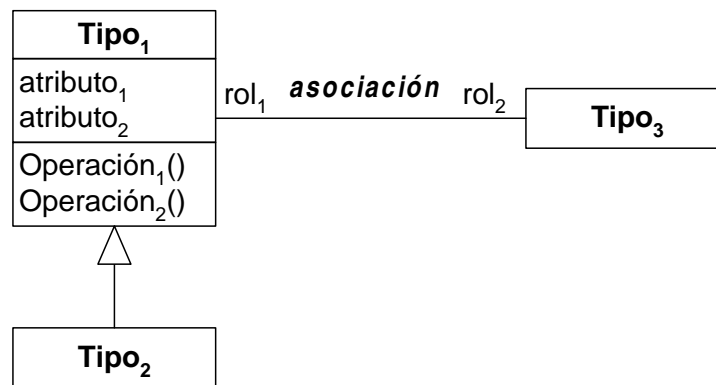


Figura 4.9: Ejemplo de diagrama de tipos

Las principales diferencias entre los ERDs y los diagramas de tipos son la capacidad semántica de estos últimos para representar relaciones de generalización/especialización que definan subtipos y la asignación de operaciones a los tipos.

Aunque casi todas las propuestas actuales de ERDs [Yourdon Inc. 1993, Nijssen y Halpin 1989, Wieringa 1996] permiten establecer interrelaciones *es-un*, las técnicas orientadas a objetos ofrecen una mayor riqueza semántica (subtipos disjuntos, subtipos solapados, subtipos dinámicos, etc.).

Ninguna propuesta de ERD contempla asociar operaciones a las entidades, probablemente porque si lo hiciera sería calificada como orientada a objetos y dejaría de ser una propuesta de ERD.

Otra diferencia entre ambas técnicas de modelado estático es la posibilidad en los diagramas de tipos de incluir anotaciones textuales que permitan expresar restricciones para las que no hay una alternativa gráfica de expresión. Estas restricciones pueden expresarse, por ejemplo, en OCL (*Object Constraining Language, Lenguaje de Restricciones de Objetos*) [Rat 1997, Warmer y Kleppe 1999].

A los diagramas de tipos les acompañan, a parte de las anotaciones textuales ya comentadas, descripciones textuales de los elementos que los componen: los tipos, sus atributos, las asociaciones, etc.

4.4.2 Modelado funcional

Aunque en propuestas como OMT [Rumbaugh *et al.* 1991] se propone, de forma no muy clara, el uso de DFDs para modelar los aspectos funcionales

del sistema, la mayor parte de las técnicas orientadas a objetos de primera generación utilizan la especificación de operaciones y los diagramas de secuencia o de colaboración para describir esta dimensión del sistema. En las figuras 4.10 y 4.11 pueden verse dos ejemplos semánticamente equivalentes de estos dos tipos de diagramas.

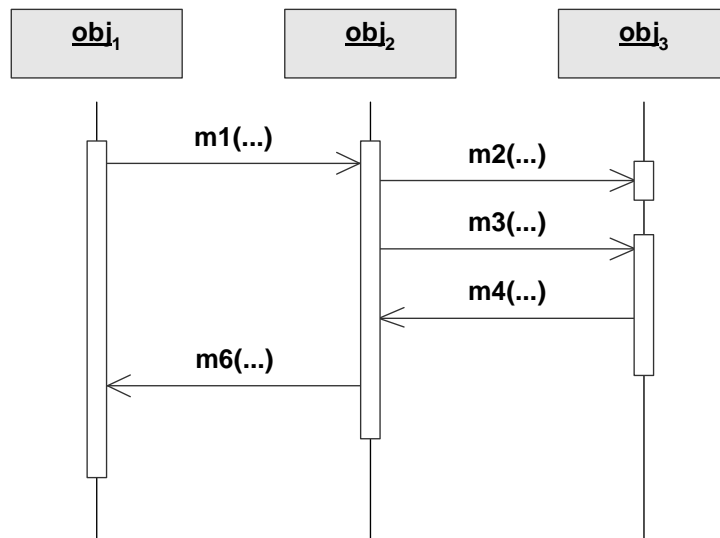


Figura 4.10: Ejemplo de diagrama de secuencia

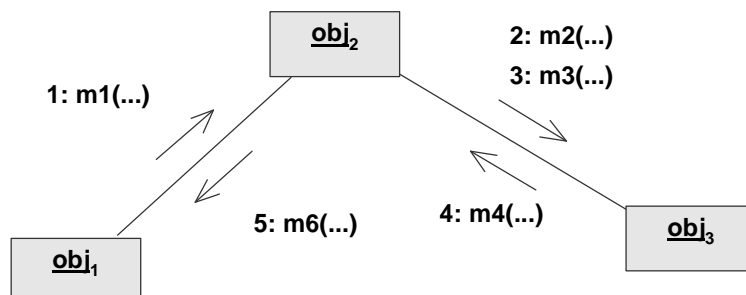


Figura 4.11: Ejemplo de diagrama de colaboración

Estas técnicas de primera generación asumen en sus modelos una comunicación entre objetos muy similar a la que se realiza en los lenguajes de programación orientados a objetos: el *paso de mensajes*. De esta forma, la funcionalidad del sistema se logra mediante una adecuada secuencia de envío de mensajes a los distintos objetos que lo forman.

Los inconvenientes de esta forma de modelar la funcionalidad de un sistema son varios. En primer lugar, tal como se expone por ejemplo en [Cook y Daniels 1994, págs. 78, 137–139] o en [D’Souza y Wills 1999, págs. 154–155], los modelos de paso de mensajes suelen implicar decisiones de diseño, ya que es necesario especificar la secuencia de mensajes entre los objetos, es decir, especificar el *protocolo* de comunicación.

En segundo lugar, implica la necesidad de tomar decisiones sobre la distribución de la funcionalidad global del sistema entre los distintos objetos que lo componen⁵, lo que lleva necesariamente a especificar las operaciones que realizarán los objetos cuando reciban un determinado mensaje.

En nuestra experiencia en la especificación de sistemas de información, este *reparto* de funcionalidad mediante la identificación y especificación de las operaciones de los objetos provoca un gran incremento del esfuerzo en la realización de los modelos de análisis pero no incrementa su rigor o precisión. De hecho, la mayor parte de estas operaciones son tan triviales como cambiar el valor de un atributo o establecer el enlace de una asociación, con lo que su nivel de abstracción es similar al de la implementación.

Para evitar estos problemas, los métodos de segunda generación como [Coleman *et al.* 1994], [Cook y Daniels 1994] o [D’Souza y Wills 1999] proponen distintas formas de expresar la funcionalidad del sistema en las que se asume otro modelo de comunicación entre objetos que evita tomar decisiones de diseño (*Syntropy*), o bien en las que la asignación de funcionalidades o responsabilidades entre los objetos no se realiza durante las actividades de análisis, sino que se pospone para el diseño (*Fusion, Catalysis*).

De hecho, existen propuestas que usan estos mecanismos alternativos al paso de mensajes como base para la especificación y el prototipado de sistemas distribuidos [Corchuelo 1999, Corchuelo *et al.* 2000].

⁵Este problema se denomina *behavior allocation* [Kovitz 1998, págs. 247–248]

4.4.2.1 Modelado funcional en Syntropy

En Syntropy [Cook y Daniels 1994], el modelado funcional y el dinámico están íntimamente ligados y se expresan mediante una única técnica: el *statechart* [Harel 1987] con anotaciones textuales. En la figura 4.12 puede verse un ejemplo de statechart para un determinado tipo de objetos, en el que aparecen dos estados en el *cuerpo* (la parte gráfica) y los distintos grupos de eventos en la *lista de eventos* (la parte textual).

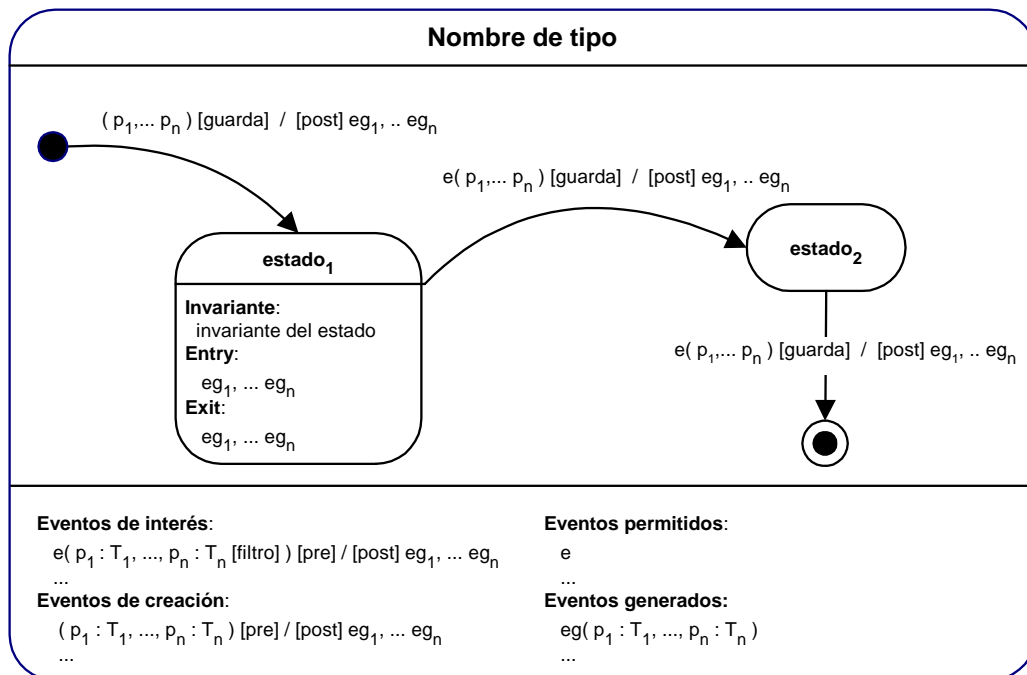


Figura 4.12: Ejemplo de statechart de Syntropy

En Syntropy, la lista de eventos de los statechart puede contener hasta cuatro grupos de eventos:

- **Eventos de interés:** son los eventos en los que los objetos del tipo al que corresponde el statechart están interesados y que provocan alguna transición. El formato para definir un evento es:

$$e(p_1:T_1, \dots, p_n:T_n \text{ [filtro] }) [pre] / [post] eg_1, \dots, eg_n$$

donde:

- **e** es el nombre, global y único, del evento.

- $p_1:T_1, \dots, p_n:T_n$ son los parámetros del evento, cada uno con un nombre y un tipo correspondiente.
- *[filtro]* es un predicado que para cada ocurrencia del evento indica si el objeto está o no interesado en dicha ocurrencia.
- *[pre]* es la precondition del evento, un predicado que indica, en el caso de que el filtro sea cierto, si el objeto está o no preparado para aceptar una determinada ocurrencia del evento.
- *[post]* es la postcondición del evento, un predicado que define las condiciones que se deben cumplir si el objeto experimenta una transición como consecuencia del evento. En el cuerpo del statechart pueden aparecer otras postcondiciones asociadas a transiciones específicas del evento. La postcondición de una transición es la conjunción lógica (\wedge) de la expresada en la lista de eventos y la expresada en el cuerpo del statechart.
En otras palabras, la postcondición de la lista de eventos es común a todas las transiciones del evento en el cuerpo del statechart.
- eg_1, \dots, eg_n son los eventos que se generan si el objeto experimenta una transición como consecuencia del evento. Al igual que en el caso de las postcondiciones, pueden indicarse eventos generados específicos de ciertas transiciones del evento, que en caso de producirse generarían tanto los especificados en la lista de eventos como en el cuerpo del statechart.

Tanto las postcondiciones como los eventos generados asociados a un evento en la lista de eventos son comunes a todas las transiciones provocadas por dicho evento, permitiendo así simplificar el cuerpo del statechart.

- **Eventos de creación:** son los eventos que provocan la creación de los objetos. Este tipo de eventos recibe un tratamiento especial en Syntropy, ya que según su modelo de comunicación entre objetos un objeto no puede detectar la ocurrencia de un evento si dicho objeto no existe todavía. Esta es la razón de que este tipo de eventos no tenga nombre y sólo se diferencien por los tipos de los parámetros.

Para que se cree un objeto como consecuencia de un evento e , es necesario que algún objeto ya existente experimente una transición con e en la que se incluya como postcondición una expresión de la forma:

new $T(p_1, \dots, p_n)$

donde T es el nombre del tipo de objeto que se quiere crear y p_1, \dots, p_n son los valores de los parámetros que servirán para distinguir entre las distintas posibilidades de creación.

Esta forma de modelar la creación de objetos hace necesario asumir la *preexistencia* de al menos un *objeto inicial*, ya que de no ser así, el sistema no podría evolucionar al no permitir la ocurrencia de eventos y por tanto la creación de objetos [Cook y Daniels 1994, págs. 81–84]. Los objetos iniciales se distinguen en los diagramas de tipos por estar dibujados con un trazo más grueso.

- **Eventos permitidos:** son aquellos eventos de interés que pueden ocurrir en cualquier estado aunque no aparezcan en ninguna transición en el cuerpo del statechart. Los eventos permitidos son un mecanismo que permite simplificar los statecharts evitando numerosas transiciones de un estado hacia si mismo [Cook y Daniels 1994, págs. 100–101].
- **Eventos generados:** los eventos generados son aquellos eventos que se generan durante las transiciones y que permiten al sistema enviar respuestas al entorno. Para simplificar los diagramas, es posible indicar eventos generados de *entrada* (*entry*) y de *salida* (*exit*) de los estados. Estos eventos de entrada y salida se generan cuando cualquier transición hace que el objeto entre o salga de un determinado estado⁶, simplificando de esta forma las listas de eventos generados de todas las transiciones que comiencen o terminen en el estado en cuestión.

Dentro del cuerpo del statechart, la notación para las transiciones es la siguiente:

$e(p_1, \dots, p_n) [guarda] / [post] eg_1, \dots, eg_n$

donde:

- e es el nombre de un evento de interés en la lista de eventos o, en el caso de que se trate de una transición que parta del estado inicial, es un evento de creación y no tiene nombre.

⁶A excepción de las *autotransiciones* provocadas por los eventos permitidos, que dejan al objeto en el mismo estado pero no generan los eventos de entrada o de salida [Cook y Daniels 1994, págs. 144–145].

- p_1, \dots, p_n son los valores de los parámetros del evento, que deben coincidir en tipo y en número con los de la signatura del evento en la lista de eventos.
- *[guarda]* es un predicado que sólo en el caso de que sea verdadero permitirá que el objeto experimente la transición. Un estado puede tener transiciones de salida con el mismo evento y con diferentes guardas, que no deben nunca satisfacerse a la vez. Las guardas pueden actuar como selectores de transiciones (siempre que alguna sea cierta), o bien como precondiciones específicas de cada transición [Cook y Daniels 1994, pág. 99].
- *[post]* es la postcondición específica de la transición.
- eg_1, \dots, eg_n son los eventos generados específicos de la transición.

El modelo de comunicación que se sigue en Syntropy asume que los eventos enviados por el entorno al sistema son *detectados* por todos los objetos que componen el sistema. Un evento provoca una transición en el estado global del sistema si se cumplen las siguientes tres condiciones [Cook y Daniels 1994, pág. 99]:

- **los parámetros no son erróneos:** los valores de los parámetros del evento son de los tipos correctos y, si se trata de identificadores de objetos, corresponden a objetos conocidos por el sistema.
- **los parámetros no son inválidos:** los valores de los parámetros, asumiendo que el evento ocurra, no llevarán a violar ninguna restricción del modelo, es decir, ninguna invariante de clase o de estado.
- **el modelo está en el estado correcto:** todos los statecharts de los objetos del sistema permiten que el evento ocurra. Un statechart permite que un evento ocurra cuando se da alguna de las siguientes circunstancias [Cook y Daniels 1994, pág. 125]:
 - el evento no está en la lista de eventos del statechart, con lo que se asume que no le interesa y por lo tanto permite su ocurrencia.
 - el evento está en la lista de eventos pero no satisface el filtro asociado al evento en la lista de eventos del statechart, con lo que también se asume que no le interesa.
 - el evento está en la lista de eventos, satisface el filtro, satisface las precondiciones y provoca al menos una transición.

- el evento está en la lista de eventos, satisface el filtro, satisface las precondiciones y es uno de los eventos permitidos.

En general, la forma de modelar los aspectos dinámico/funcionales en Syntropy puede considerarse mucho más formal y de un mayor nivel de abstracción que las propuestas de los métodos de primera generación, ya que evita tomar decisiones sobre la secuencia de mensajes que se deben enviar a los objetos del sistema.

Sin embargo, en nuestra opinión presenta un inconveniente importante, y es que para poder comprender la conducta del sistema en su conjunto como respuesta a un evento externo es necesario deducir dicha conducta a partir de las conductas locales de cada tipo de objetos. En interacciones complejas en las que intervengan varios objetos de distintos tipos puede ser bastante complejo llegar a entender la conducta global.

Otro posible problema que presenta Syntropy en los aspectos funcionales es la redundancia de las postcondiciones cuando un evento es detectado por objetos de más de un tipo, por ejemplo en eventos que provocan el establecimiento de los enlaces en asociaciones bidireccionales. Los autores rechazan la idea de expresar estas postcondiciones de forma global porque [Cook y Daniels 1994, pág. 119]:

"[...] entra en conflicto con uno de los principios de la orientación a objetos, que es localizar las descripciones de la conducta en los tipos de objetos, en lugar de hacerlo separadamente."

Nuestra opinión al respecto es que no hay que olvidar que los objetivos de la construcción de modelos conceptuales en análisis son la detección de conflictos en los requisitos-C, profundizar en el conocimiento del problema y sentar las bases del diseño. Si la aplicación de un principio dificulta alguno de estos objetivos es conveniente revisar dicho principio y no caer en posiciones *puristas*. El no asignar conducta específica a los objetos durante el modelado conceptual y posponerla para el diseño es una opción tan válida o más como sí hacerlo.

En nuestra opinión, y coincidiendo con [Coleman *et al.* 1994], [Iturrioz 1998] y [D'Souza y Wills 1999], creemos que los objetivos del análisis de requisitos se alcanzan de una forma más satisfactoria si los aspectos funcionales se expresan a nivel global que a nivel de tipos de objetos, ya que los modelos resultantes son más fáciles de entender por los desarrolladores y se evita tener que especificar detalles propios del diseño.

4.4.2.2 Modelado funcional en Fusion

El modelado funcional en Fusion [Coleman *et al.* 1994] se realiza mediante el denominado *modelo de operaciones*, en el que se describen las operaciones que se realizan a nivel de todo el sistema como respuesta a los estímulos de su entorno. Estas operaciones actúan sobre el estado del sistema, compuesto por los objetos, sus atributos y sus enlaces, que se considera global.

Para describir estas operaciones del sistema, que forman su interfaz con el entorno, se propone la utilización de *esquemas de operaciones de sistema*, con una estructura similar a las plantillas descritas en el capítulo anterior. La estructura de estos esquemas puede verse en la figura 4.13.

Operación:	nombre de la operación del sistema, que debe ser único. Se recomienda usar nombres intuitivos y en formato libre.
Descripción:	descripción breve e informal de la operación del sistema.
Lee (Reads):	lista de objetos, atributos o asociaciones que la operación lee pero no cambia. Si un elemento se prefija con suministrado (<i>supplied</i>) significa que es un parámetro de la operación.
Cambia (Changes):	lista de objetos, atributos o asociaciones que la operación cambia. Si un elemento se prefija con nuevo (<i>new</i>) significa que se crea durante la operación.
Envía (Sends):	lista de todos los eventos y agentes del entorno a los que el sistema les envía dichos eventos al realizar la operación.
Asume (Assumes):	Precondición, descrita en lenguaje natural, que debe satisfacerse para que la operación del sistema pueda realizarse.
Resultado:	Postcondición, descrita en lenguaje natural, que debe satisfacerse después de la realización de la operación del sistema. Para hacer referencia a los estados previo y posterior a la realización de la operación pueden utilizarse los prefijos inicial y final respectivamente.

Figura 4.13: Esquema de *operación de sistema* de Fusion

En resumen, Fusion no distribuye la funcionalidad entre los objetos durante las actividades de análisis, no identifica por lo tanto operaciones a nivel de objetos y especifica la funcionalidad del sistema a nivel global, siguiendo fielmente el modelo caja gris de la figura 4.2 (pág. 129).

En nuestra opinión, el principal inconveniente es la utilización de lenguaje natural para la especificación de las pre y postcondiciones. La au-

diencia de los requisitos–D son los desarrolladores, y por este motivo pueden usarse notaciones específicas que permitan un mayor rigor en la especificación. El uso del lenguaje natural está justificado para la redacción de los requisitos–C, pero para los requisitos–D es mejor utilizar lenguajes más formales que no presentan los problemas inherentes al lenguaje natural [D’Souza y Wills 1999, págs. 585–586].

4.4.2.3 Modelado funcional en Catalysis

En Catalysis [D’Souza y Wills 1999], el modelo funcional en las actividades de análisis se expresa mediante *acciones conjuntas*, que es la forma directa de modelar los casos de uso [D’Souza y Wills 1999, págs. 158 y 164] y que coinciden básicamente con la *operaciones de sistema* de Fusion y con las *interacciones entre múltiples participantes* de [Corchuelo 1999, Corchuelo *et al.* 2000].

Una acción conjunta se define como [D’Souza y Wills 1999, pág. 158]:

acción conjunta (*joint action*): *una acción conjunta representa un cambio en el estado de varios objetos participantes sin especificar cómo ocurre y sin atribuir aún la responsabilidad de ello a ninguno de los participantes.*

Las acciones conjuntas tienen una representación gráfica que puede verse en la figura 4.14, en la que aparecen la acción conjunta con sus parámetros representada como una elipse (el mismo símbolo que para los casos de uso) y los tipos de los objetos que intervienen en la acción conjunta junto con los nombres de los roles o papeles que desempeñan y la cardinalidad, es decir, el número de objetos de cada tipo que interviene en la acción, que no siempre tiene porqué ser uno (en el ejemplo de la 4.14 participa un conjunto de objetos del Tipo₃).

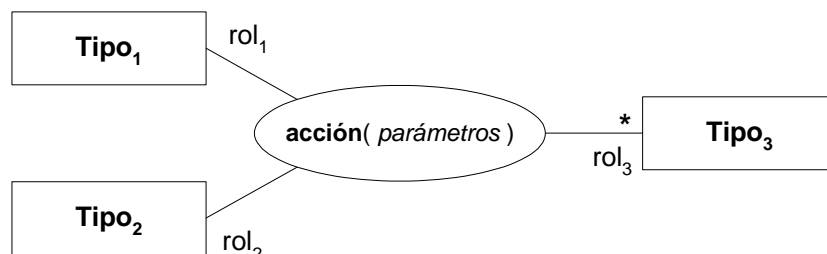


Figura 4.14: Representación gráfica de una acción conjunta en Catalysis

La definición textual que debe acompañar a la representación gráfica es la siguiente:

acción $(rol_1:T_1, \dots, rol_n:T_n)::ac(p_1:T_p, \dots, p_n:T_q)$
pre: *precondición de la acción conjunta en OCL*
post: *postcondición de la acción conjunta en OCL*

donde:

- $rol_1:T_1, \dots, rol_n:T_n$ son los participantes en la acción conjunta, cada uno definido con un nombre de rol y un tipo de objetos.
- *ac* es el nombre de la acción conjunta.
- $p_1:T_p, \dots, p_n:T_q$ son los parámetros (valores) de la acción conjunta.

Las acciones conjuntas que se realicen como consecuencia de las interacciones del sistema con el entorno pueden representarse como las operaciones del sistema que se está especificando, en cuyo caso los objetos participantes se consideran como parámetros. Por ejemplo, suponiendo que la anterior definición correspondía a un sistema de nombre *S*, se podría tener la siguiente definición equivalente:

acción $S::ac(rol_1:T_1, \dots, rol_n:T_n, p_1:T_p, \dots, p_n:T_q)$
pre: *precondición de la acción conjunta en OCL*
post: *postcondición de la acción conjunta en OCL*

La razón de esta equivalencia es que dichas acciones formarían la interfaz del sistema con el entorno, al igual que lo forman las operaciones de sistema en Fusion y los eventos en Syntropy. Considerando al sistema como un objeto, estas acciones serían los servicios que podrían solicitarse al sistema. Esta idea se refuerza mediante el uso habitual de los diagramas de tipos contenidos dentro del tipo que representa al sistema en su conjunto (ver figura 4.15).

Las acciones conjuntas definen sus pre y postcondiciones sobre el estado global del sistema sin asignar responsabilidades ni funcionalidades a ninguno de los participantes. Sin embargo, puede haber ocasiones en las que pueda ser oportuno *factorizar* algunas pre y postcondiciones comunes a varias acciones conjuntas y localizarlas como *efectos* asociados a uno o más tipos de objetos [D'Souza y Wills 1999, págs. 167–168].

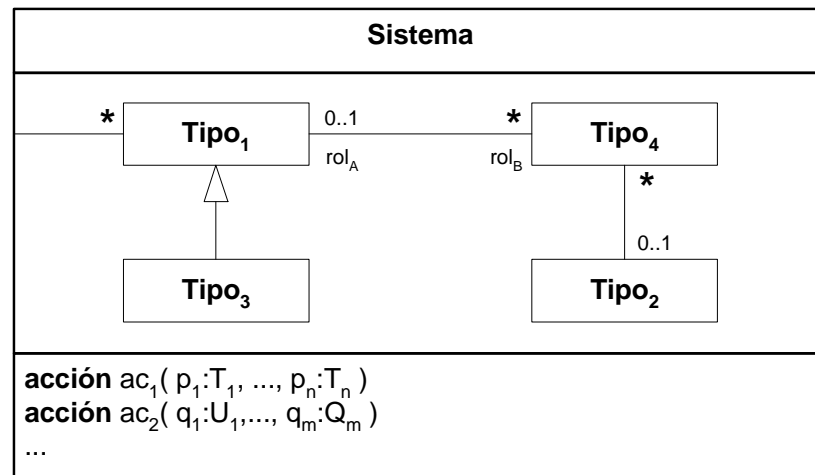


Figura 4.15: Representación de las acciones como interfaz del sistema

Por ejemplo, si un determinado tipo de objetos T participa en varias acciones y en la postcondición de la mayoría de ellas aparece la misma subexpresión e en la que aparece un objeto de tipo T , por ejemplo $t.a < x$, es posible sacar factor común dicha subexpresión y definirla como un *efecto* sobre el tipo T :

efecto $T::e(x:X)$
post: $a < x$

que luego se integraría en las acciones de la siguiente forma:

acción $(\dots, t:T, \dots)::ac(\dots, y:X, \dots)$
post: $t.e(y)$ and ...

En general, la propuesta de modelado funcional en Catalysis es la que nos parece más atractiva. Evita los problemas de los modelos de paso de mensajes, no está distribuida entre los tipos de objetos como en el caso de Syntropy sino que está definida en un único lugar⁷, y utiliza el lenguaje formal OCL en lugar del lenguaje natural propuesto en Fusion.

⁷Salvo que se factoricen algunas acciones mediante *efectos*, como ya se ha comentado, aunque podrían expandirse de forma automática en la documentación final.

4.4.3 Modelado dinámico

La técnica más utilizada dentro de las técnicas orientadas a objetos para representar los aspectos dinámicos es el *statechart* [Harel 1987], que normalmente se utiliza para modelar la conducta de los objetos de un determinado tipo.

Las características de modelado dinámico específicas de Syntropy se comentaron en la sección 4.4.2.1. Las características relativas a Fusion y a Catalysis se comentan a continuación.

4.4.3.1 Modelado dinámico en Fusion

En Fusion, los aspectos dinámicos, al igual que los funcionales, se describen también a nivel del sistema completo. Para ello proponen la utilización de un modelo denominado de *ciclo de vida*, en el que se utilizan *expresiones de ciclo de vida*, muy parecidas a las expresiones regulares concurrentes [Garg 1990], que expresan las posibles secuencias de operaciones de sistema y generación de eventos que pueden realizarse.

Las expresiones pueden formarse utilizando los siguientes operadores, donde x e y son expresiones de ciclo de vida:

- $x.y$ representa que x es seguida por y
- $x \mid y$ representa que ocurre x o que ocurre y
- x^* representa que x ocurre cero o más veces
- x^+ representa que x ocurre una o más veces
- $[x]$ representa que x es opcional
- $x \parallel y$ representa el entrelazado de x e y

También pueden nombrarse subexpresiones para facilitar la lectura de expresiones complejas y utilizar paréntesis para indicar la precedencia.

Tal como se reconoce en [Coleman *et al.* 1994, pág. 33], la expresividad de esta técnica aplicada al sistema en su conjunto es bastante limitada, entre otras razones, por la imposibilidad de expresar comportamientos condicionales.

En nuestra opinión, las expresiones de este tipo pueden ser una forma sencilla de expresar aspectos dinámicos a nivel de tipos de objetos⁸, pero a nivel del sistema completo se muestra ineficaz debido a su escaso poder expresivo.

4.4.3.2 Modelado dinámico en Catalysis

El modelado dinámico en Catalysis utiliza los statecharts para describir los aspectos locales de las acciones conjuntas, que son globales, sobre los objetos de un determinado tipo, aunque sin añadir la parte textual de Syntropy (la *lista de eventos*) y simplificando las transiciones para especificarlas de la forma:

$$[pre] a(p_1, \dots, p_n) / post$$

donde:

- *a* es un nombre de una acción conjunta, equivalente al concepto de evento en Syntropy.
- p_1, \dots, p_n son los valores de los parámetros de la acción.
- *[pre]* es la precondition de la transición, equivalente al concepto de *guarda* en Syntropy. La precondition de la transición puede considerarse como una *proyección* de la precondition de la acción conjunta que provoca la transición en la que sólo aparecen los aspectos relativos a los objetos del tipo cuyo statechart se está especificando.
- *[post]* es la postcondition de la transición, equivalente al concepto del mismo nombre en Syntropy, y que al igual que en el caso de las preconditiones, es una proyección de la postcondition de la acción conjunta sobre el tipo de objetos del statechart.

Los statecharts en Catalysis están integrados con los modelos estático y funcional de una forma más concreta que en métodos de primera generación como OMT o el método de Booch. Un problema que presentaban estos métodos era que, a la hora de especificar las operaciones de

⁸Como se hace en el lenguaje de especificación formal orientado a objetos TESORO [Torres 1997, Troyano 1998, Corchuelo 1999], en el que se denominan *expresiones de comportamiento*.

una clase, no existía una relación directa de éstas con el statechart salvo la coincidencia entre los nombres de las operaciones y el de los eventos en el statechart. Los estados del statechart definen estados de los objetos en los que suelen responder de forma diferente a los mensajes que se les envían, sin embargo en las definiciones de las operaciones no había ninguna referencia explícita al estado en que se encontraba el objeto.

Catalysis resuelve este problema integrado aún más los modelos dinámico y estático: un statechart no sólo está asociado a un tipo de objetos, sino que cada estado del statechart está representado mediante un atributo booleano con el mismo nombre que el estado y la estructura del statechart define invariantes sobre el tipo de objetos asociado.

Por ejemplo, el statechart de la figura 4.12 (pág. 144) definiría dos atributos booleanos de nombre `estado1` y `estado2` y además definiría una invariante con la expresión OCL:

`estado1 xor estado2`

ya que el objeto no puede estar en los dos estados a la vez. Una discusión más amplia sobre las invariantes que impone la estructura del statechart, por ejemplo con estados anidados o concurrentes, puede consultarse en [D'Souza y Wills 1999, págs. 126–130].

Coincidiendo con Syntropy, en Catalysis se recomienda asumir un modelo de comunicación *broadcast* entre los objetos del sistema durante la especificación del mismo [D'Souza y Wills 1999, págs. 603–606], de forma que todos los objetos detectan la ocurrencia de las acciones del sistema. El paso de este modelo de comunicación broadcast a un modelo de paso de mensajes más cercano a la implementación se pospone para fases posteriores del desarrollo⁹.

En general, la semántica de los statecharts en Catalysis es más simple que en Syntropy y, normalmente, hay que remitirse al modelo funcional (las especificaciones de las acciones conjuntas) para poder comprender la conducta global del sistema. Por ejemplo, supongamos el statechart de un tipo de objetos T que puede verse en la figura 4.16, del que puede deducirse lo siguiente:

- Que los objetos del tipo T tendrán tres atributos booleanos de nombre `s1`, `s2` y `s3`, correspondientes a los estados del statechart.

⁹Al igual que en [Kovitz 1998], el problema de la *behavior allocation* se considera un problema de diseño en Catalysis.

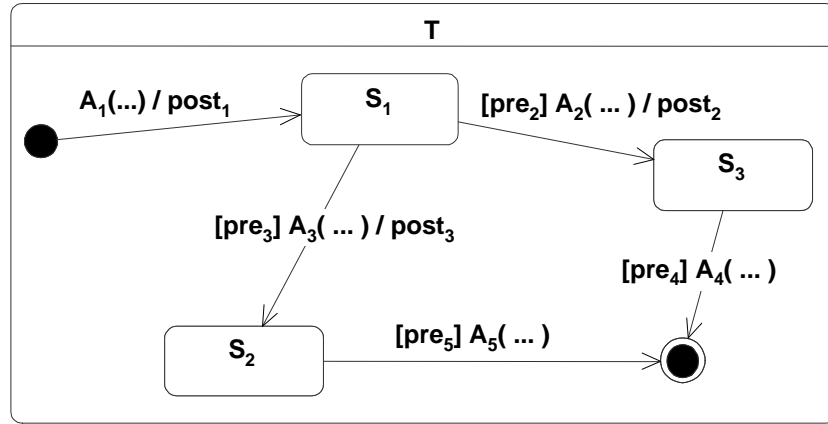


Figura 4.16: Ejemplo de Statechart (Catalysis)

- Que en la invariante del tipo T debe aparecer la expresión $s_1 \text{ xor } s_2 \text{ xor } s_3$, ya que un objeto de tipo T sólo puede estar en un estado a la vez.
- Que la acción conjunta A_1 provocará la creación de algún objeto (no se puede deducir cuántos a partir del statechart) de tipo T. Es decir, que en la especificación de A_1 , una de las expresiones de la postcondición será que todos los objetos de tipo T creados cumplan la expresión post_1 **and** s_1 .
- Que una de las expresiones de la postcondición de la acción conjunta A_2 especificará que todos aquellos objetos de tipo T que estaban en el estado s_1 y que cumplían pre_2 deben estar ahora en el estado s_2 y deben cumplir post_2 , es decir¹⁰:

T.allInstances->forAll(t |
 (t.s1 @pre **and** t.pre2 @pre) **implies** (t.s2 **and** t.post2))

En el caso de que la acción afecte a un único objeto de tipo T, por ejemplo uno pasado como parámetro, puede simplificarse la expresión anterior eliminando el cuantificador universal.

El hecho de que no exista ningún objeto de tipo T en condiciones de experimentar una transición con A_2 no implica que A_2 no pueda

¹⁰ Aunque pre_2 y post_2 serán normalmente expresiones complejas, se ha simplificado la notación, de forma que $t.\text{pre}_2 @\text{pre}$ y $t.\text{post}_2$ representan a las expresiones al evaluarse sobre t.

ocurrir, como en Syntropy. Si se considera necesario que exista al menos un objeto de tipo T que pueda tener una transición con A_2 , debe contemplarse explícitamente mediante una precondition en la especificación de A_2 , por ejemplo:

```
T.allInstances->exists( t |
    t.s1@pre and t.pre2@pre )
```

Un razonamiento similar puede hacerse con A_3 .

En el caso de A_4 y A_5 , la única diferencia con las demás operaciones es que a nivel local no se suelen especificar postcondiciones, ya que el objeto no existe. A nivel global sí que deberá aparecer una expresión en las postcondiciones de A_4 y A_5 que indique que los objetos participantes de tipo T no pertenecen ya al conjunto de instancias de dicho tipo, es decir, han dejado de existir.

En resumen, el modelado dinámico en Catalysis no es más que "*una vista alternativa de la especificación [textual]*" [D'Souza y Wills 1999, pág. 126], ya que puede representarse a partir ésta. Está integrado con el modelado funcional y tiene una semántica más simple que los statecharts de Syntropy.

4.4.4 Integración de modelos

La integración entre las tres dimensiones del modelado en las técnicas orientadas a objetos es más fuerte que entre las técnicas estructuradas. El concepto principal es el de clase o tipo de objetos, que además de ser el eje del modelado estático, sirve de elemento de integración entre los tres aspectos del modelado.

En las técnicas de primera generación, los aspectos funcionales están integrados con el modelo estático mediante las operaciones asignadas a los tipos de objetos, aunque la integración con los aspectos dinámicos no está clara.

En los modelos de segunda generación, el propio sistema se considera un objeto que ofrece una serie de servicios u operaciones a su entorno. Los efectos de la realización de las operaciones a nivel del sistema sobre los distintos objetos que lo componen pueden expresarse en las descripciones textuales de dichas operaciones del sistema y complementarse con los statecharts de aquellos tipos de objetos con un dinamismo significativo.

En general, la integración entre los tres modelos puede considerarse satisfactoria en los métodos de segunda generación.

4.5 Comparación de técnicas de modelado

En las secciones 4.3 y 4.4 se han comentado brevemente las técnicas de modelado conceptual más habituales, tanto estructuradas como orientadas a objetos. Ambos tipos de técnicas siguen, más o menos fielmente, el modelo de sistema de caja gris que se describió en la sección 4.2.

Si hacemos una comparación entre las técnicas estructuradas de tercera generación (aquellas que no se basan exclusivamente en los DFDs y que se preocupan de la integración de los distintos modelos) y las técnicas orientadas a objetos de segunda generación (las que no distribuyen responsabilidades entre los objetos durante el modelado conceptual), puede observarse que no existen tantas diferencias como se supone y que pueden considerarse casi equivalentes, tal como se expone en [Wieringa 1998].

En la figura 4.17 puede verse la comparación entre las técnicas más habituales de modelado. Nuestra conclusión es que ambas técnicas son similares, aunque las orientadas a objetos permiten una mayor expresividad y tienen un mayor nivel de integración entre ellas. Las mayores diferencias se encuentran, coincidiendo con [Wieringa 1998], en los DFDs, como ya se comentó en la sección 4.3.4, pág. 137.

	Estructuradas	Objetos	Comparación
D a t a s	Diagrama entidad-interrelación (ERD)	Diagrama de tipos	El concepto de entidad es equivalente al de tipo y el de interrelación es equivalente al de asociación. Los diagramas de tipos son más ricos semánticamente, principalmente debido a la posibilidad de definir subtipos.
F u n c i o n e s	Diagrama de flujo de datos (DFD)	Operaciones de sistema (Fusion) o acciones conjuntas (Catalysis)	La integración de los DFDs con el modelo de datos no es satisfactoria al asumir una estructura relacional o de ficheros tipo COBOL para la persistencia del estado del sistema. Si se aplica el particionamiento de eventos, los procesos de los DFDs asociados a un evento son equivalentes a las operaciones del sistema de Fusion o a las acciones conjuntas de Catalysis.
E v e n t o s	Historia de la vida de las entidades (HVE)	Statechart	Ambas técnicas describen las secuencias válidas de eventos que pueden afectar una entidad/tipo de objetos. Los statecharts tienen una mayor expresividad que los diagramas HVE (posibilidad de anidar estados, estados concurrentes, guardas, precondiciones, etc.).

Figura 4.17: Comparación de técnicas estructuradas y orientadas a objetos

4.6 Propuesta metodológica para el análisis de requisitos

La propuesta metodológica para el análisis de requisitos que se propone en este trabajo, y que se incluye en el apéndice C, propone una serie de tareas a realizar y productos a obtener durante la realización de la actividad de análisis de requisitos del modelo de procesos de ingeniería de requisitos descrito en el capítulo 2.

En la propuesta se contemplan las cuatro tareas que pueden verse en la figura 4.18, en la que se propone un posible orden de realización orientativo en el que las tareas de analizar los tres tipos de requisitos–C se realizarían en paralelo. En las siguientes secciones se describen cada una de ellas.

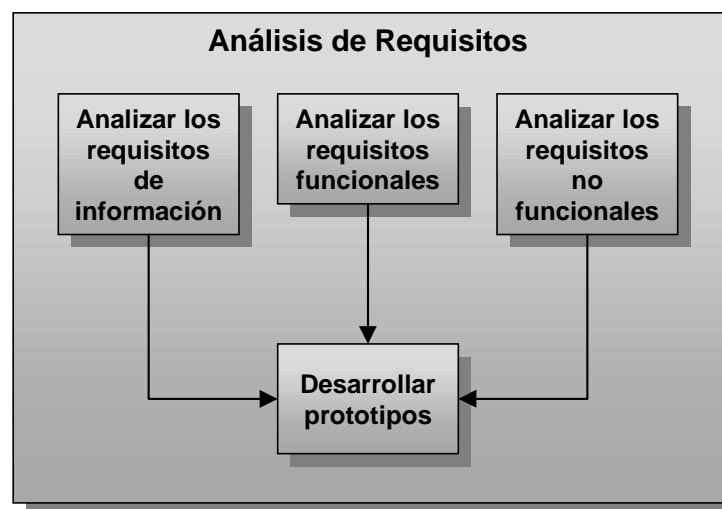


Figura 4.18: Tareas de análisis de requisitos

4.6.1 Tarea 1: Analizar los requisitos de almacenamiento de información

El objetivo principal de esta tarea, al igual que el de la siguiente, es descubrir conflictos en los requisitos–C profundizando en el conocimiento del problema y estableciendo las bases para un futuro diseño del sistema. En el caso concreto de esta tarea, el objetivo es descubrirlos en los requisitos–C de almacenamiento de información.

La forma habitual de alcanzar estos objetivos es mediante la construcción de modelos abstractos. En esta propuesta se recomienda la utilización de técnicas de modelado orientadas a objetos, aunque también podrían utilizarse técnicas estructuradas, ya que, como se comentó en la sección 4.5, las diferencias entre ambas no son insalvables.

Los productos resultantes de la realización de esta tarea son el modelo estático del sistema, compuesto por los diagramas de tipos y la descripción textual de los elementos que lo integran, y los posibles conflictos que se detecten al construir el modelo.

Las técnicas que pueden utilizarse para realizar esta tarea son los diagramas de tipos [D'Souza y Wills 1999], las plantillas para describir los elementos de los diagramas de tipo (ver sección 4.7.1) y las plantillas para conflictos descritas en la sección 3.6.6 (pág. 95).

4.6.2 Tarea 2: Analizar los requisitos funcionales

Esta tarea es similar a la anterior, con la diferencia de que se centra en los requisitos-C funcionales, expresados mediante casos de uso, en lugar de hacerlo en los requisitos de almacenamiento de información.

Para analizar los requisitos funcionales lo habitual es construir modelos funcionales y, si se considera oportuno, modelos dinámicos. Por lo tanto, los productos resultantes de la realización de esta tarea son los modelos funcional y dinámico y los conflictos que se hayan detectado al construir ambos modelos.

Dentro de las técnicas orientadas a objetos comentadas en la sección 4.4, en esta propuesta se recomienda utilizar las *acciones conjuntas* de Catalysis, aunque utilizando la nomenclatura de *Fusion de operaciones de sistema*, para modelar los aspectos funcionales del sistema.

Para modelar los aspectos dinámicos, se recomienda el uso de statecharts con la semántica descrita en Catalysis, en la que cada statechart está asociado a un tipo de objetos, los estados del statechart corresponden a atributos booleanos de los objetos, la estructura del statechart se expresa en la invariante y las transiciones se especifican textualmente en las pre y postcondiciones de las operaciones del sistema.

Los statecharts se utilizarán para modelar los aspectos dinámicos de aquellas clases con un dinamismo significativo, es decir, aquellas para las que se identifique claramente más de un estado.

4.6.3 Tarea 3: Analizar los requisitos no funcionales

Esta tarea tiene también como objetivo descubrir conflictos en los requisitos—C, en este caso en los requisitos no funcionales. Sin embargo, a diferencia de las dos tareas anteriores, las escasas propuestas de modelado que integran este tipo de requisitos aun no pueden considerarse satisfactorias [Cysneiros y Leite 1999].

La naturaleza heterogénea de este tipo de requisitos hace que su análisis sea necesario realizarlo mediante una lectura detenida de su contenido, y combinando esta lectura con la experiencia, detectar posibles conflictos como la imposibilidad técnica de la implementación de ciertos requisitos, la necesidad de optar por unas características u otras (por ejemplo flexibilidad frente a eficiencia), etc.

Normalmente, estos requisitos se tendrán en consideración durante el diseño de la arquitectura del sistema [Chung *et al.* 1999, Ruiz *et al.* 2000], lo que probablemente provocará iteraciones entre el resto del desarrollo y la fase de ingeniería de requisitos.

Los productos resultantes de esta tarea son, por lo tanto, aquellos conflictos que se hayan detectado al realizar el análisis de los requisitos no funcionales.

4.6.4 Tarea 4: Desarrollar prototipos

El objetivo de esta tarea es el desarrollo de prototipos que puedan utilizarse durante la elicitación o validación de los requisitos, y que por lo tanto deben centrarse en la interfaz de usuario [Gomaa 1990], independientemente de que su naturaleza sea de *usar y tirar* o *evolutiva*.

Las actuales herramientas de *desarrollo rápido de aplicaciones* (RAD, *Rapid Application Development*) permiten la construcción rápida de interfaces de usuario que no tienen porqué desecharse una vez utilizadas para obtener información de los clientes [Corchuelo *et al.* 1998].

Otra posibilidad consiste en la prototipación de la funcionalidad de la aplicación mediante prototipos generados a partir de lenguajes de especificación formal como Z [Spivey 1992], Oasis [Pastor y Ramos 1995, Letelier *et al.* 1998] o Tesoro [Torres 1997, Troyano 1998, Corchuelo 1999].

Estos prototipos funcionales generados automáticamente no suelen tener interfaces de usuario parecidas a las que puede tener el sistema final, por lo que su uso se restringe a la comprobación de algunas propiedades del modelo por parte de los ingenieros de requisitos, sin que puedan servir para la validación o elicitación con los clientes y usuarios.

4.6.5 El documento de análisis del sistema

En la propuesta metodológica, el *Documento de Análisis del Sistema* (DAS) es el principal producto entregable, junto con el prototipo si se considera oportuna su realización, de las actividades de análisis de requisitos. Los conflictos detectados pueden registrarse en el *Documento de Requisitos del Sistema*, en el DAS o en un documento aparte.

La estructura del DAS, que se describe con detalle en el apéndice C, puede verse en la figura 4.19.

Este documento coincide básicamente con el *Software Requirements Specification* de la norma MIL-STD-498 [DoD 1994], el *Software Requirements Document* de las normas PSS-05 [Mazza *et al.* 1996] y el *Documento de Especificación Funcional* de Métrica V2.1 [MAP 1995a], aunque utilizando técnicas orientadas a objetos en lugar de estructuradas. En el apéndice A puede consultarse más información sobre estos documentos.

Para especificar los elementos del modelo (tipos, atributos, asociaciones, operaciones del sistema, etc.) se ha desarrollado plantillas similares a las propuestas para los requisitos-C que se describen en la siguiente sección.

Dentro del DAS se incluye una matriz de rastreabilidad que asocia elementos del modelo (requisitos-D) con requisitos-C, permitiendo así continuar con las relaciones de rastreabilidad descritas en el capítulo anterior.

Otro aspecto importante de este documento son los conflictos pendientes de resolución, que deben recogerse explícitamente usando la plantilla descrita en la sección 3.6.6 (pág. 95) para poder acometer su posterior resolución. Como ya se ha comentado, la localización física de estos conflictos puede hacerse en el *Documento de Requisitos del Sistema*, en el DAS o, si se considera oportuno, en un documento específico para los conflictos.

4.7 Plantillas y patrones lingüísticos para análisis de requisitos

Las plantillas y patrones lingüísticos (patrones-L) que se presentan en los siguientes apartados están pensados para utilizarse como forma de organizar la información correspondiente a los requisitos-D.

La notación utilizada para los patrones-L es la misma que la que se usó en las plantillas para elicitación de requisitos (ver sección 3.6, pág. 84).

Portada
Lista de cambios
Índice
Lista de figuras
Lista de tablas
1 Introducción
2 Diagramas de tipos
3 Catálogo de tipos y asociaciones del sistema
3.1 Tipo X
3.1.1 Descripción tipo X
3.1.2 Atributos tipo X
3.1.3 Enlaces tipo X
3.1.4 Invariante tipo X
3.1.5 Diagrama de estados tipo X <i>[opcional]</i>
3.2 Asociación Y entre A,B,...,C
3.2.1 Descripción asociación Y(A,B,...,C)
3.2.2 Roles asociación Y(A,B,...,C)
3.2.3 Atributos asociación Y(A,B,...,C) <i>[opcional]</i>
3.2.4 Enlaces asociación Y(A,B,...,C) <i>[opcional]</i>
3.2.5 Invariante asociación Y(A,B,...,C) <i>[opcional]</i>
3.2.6 Diagrama de estados asociación Y(A,B,...,C) <i>[opcional]</i>
4 Catálogo de operaciones del sistema
4.1 Operación Z
4.1.1 Descripción operación Z
4.1.2 Diagrama de acción conjunta de la operación Z <i>[opcional]</i>
4.1.3 Diagrama de secuencia de la operación Z <i>[opcional]</i>
4.1.4 Interfaz de usuario de la operación Z <i>[opcional]</i>
5 Matriz de rastreabilidad requisitos/modelos
6 Conflictos pendientes de resolución <i>[opcional, pueden ir en un documento aparte]</i>
7 Glosario de términos <i>[opcional]</i>
Apéndices <i>[opcional]</i>

Figura 4.19: Estructura del Documento de Análisis del Sistema

4.7.1 Plantillas para especificación de tipos y asociaciones

En la propuesta metodológica se recomienda el siguiente conjunto de plantillas para especificar los tipos de objetos, y sus asociaciones, identificados en el análisis de los requisitos de almacenamiento de información:

- **Plantilla de tipo:** una por cada tipo de objetos.
- **Plantilla de atributo:** una por cada atributo de cada tipo de objetos, incluyendo los atributos correspondientes a los estados, que normalmente serán una función de los demás atributos y enlaces [D'Souza y Wills 1999, pág. 133], es decir, serán atributos derivados. Hay tres versiones distintas de esta plantilla, según el atributo sea constante, variable o derivado.
- **Plantilla de enlace:** una por cada enlace de cada tipo de objetos, que también podrán ser constantes, variables o derivados.
- **Plantilla de invariante:** una por cada tipo de objetos.
- **Plantilla de asociación:** una por cada asociación entre tipos de objetos.
- **Plantilla de rol:** una por cada rol de cada asociación.

En las siguientes secciones se describen cada una de estas plantillas. Aquellos campos con el mismo nombre que en el caso de las plantillas para requisitos-C descritas en el capítulo 3 tienen el mismo significado.

4.7.1.1 Plantilla de tipo

El formato de la plantilla para describir los tipos de objetos puede verse en la figura 4.20. Los apartados específicos de esta plantilla son los siguientes:

- **Nombre tipo:** cada tipo se identifica por un nombre, que deberá ser único en el modelo y que se recomienda que sea un sustantivo o similar en singular.
- **Descripción:** en este apartado se indica si el tipo es abstracto o concreto y se describe el concepto que representa. En este contexto, se entenderá que un tipo es abstracto si los conjuntos de instancias o *extensiones* de sus subtipos forman una clasificación completa [Booch *et*

Tipo	<nombre tipo>		
Versión	<nº de la versión actual> (<fecha de la versión actual>)		
Autores	<ul style="list-style-type: none"> • <autor de la versión actual> (<organización del autor>) ...		
Requisitos asociados	<ul style="list-style-type: none"> • RI-x <nombre del requisito> ...		
Descripción	Este tipo { abstracto, concreto } representa <concepto que representa el tipo>		
Supertipos	<supertipos directos del tipo>		
Subtipos	<subtipos directos del tipo> ({ disjuntos, solapados }, { completos, incompletos })		
Componentes	Nombre	Tipo OCL	Mult.
	<nombre componente>	<tipo OCL del enlace>	mult.

Comentarios	<comentarios adicionales sobre el tipo>		

Figura 4.20: Plantilla de descripción de tipo de objetos

al. 1999], es decir, todas las instancias de la extensión del tipo que se está describiendo deben ser también instancias del algún subtipo.

- **Requisitos asociados:** en este apartado se indican los requisitos del *Documento de Requisitos del Sistema* (DRS) a los que está asociado el tipo, es decir, aquellos requisitos que justifican su existencia.
- **Supertipos:** en este apartado se indican los supertipos directos del tipo.
- **Subtipos:** en este apartado se indican los subtipos directos del tipo, especificando si son disjuntos o solapados y si la clasificación es o no completa [Booch *et al.* 1999].
- **Componentes:** en este apartado se indican los componentes del tipo, es decir aquellos tipos *agregados* por el tipo. Para cada componente se incluyen los siguientes apartados:
 - **Nombre:** nombre del enlace por el que el objeto agregado accede a sus componentes.
 - **Tipo OCL:** tipo OCL del enlace por el que el objeto agregado accede a sus componentes.
 - **Multiplicidad:** multiplicidad de la agregación.

4.7.1.2 Plantillas de atributos

Dentro de los atributos de un tipo se distingue entre atributos constantes, variables y derivados. Los formatos de las plantillas para describir estos tres tipos de atributos pueden verse en la figura 4.21. Los apartados específicos de estas tres plantillas son los siguientes:

Atributo constante	<i><nombre tipo>::<nombre atributo></i>
Descripción	<i><descripción del atributo></i>
Tipo OCL	<i><tipo OCL del atributo></i>
Comentarios	<i><comentarios adicionales sobre el atributo></i>

Atributo variable	<i><nombre tipo>::<nombre atributo></i>
Descripción	<i><descripción del atributo></i>
Tipo OCL	<i><tipo OCL del atributo></i>
Valor inicial	<i><valor inicial del atributo> [opcional]</i>
Comentarios	<i><comentarios adicionales sobre el atributo></i>

Atributo derivado	<i><nombre tipo>::<nombre atributo></i>
Descripción	<i><descripción del atributo></i>
Tipo OCL	<i><tipo OCL del atributo></i>
Expresión	<i><expresión OCL asociada al atributo></i>
Comentarios	<i><comentarios adicionales sobre el atributo></i>

Figura 4.21: Plantillas de descripción de atributos

- **Nombre atributo:** cada atributo se identifica por un nombre que deberá ser único dentro de un tipo, recomendándose que sea un sustantivo o similar.
- **Descripción:** en este apartado se proporciona una breve descripción del atributo.
- **Tipo OCL:** en este apartado se indica el tipo OCL del atributo.
- **Valor inicial:** para los atributos variables, en este apartado se indica, opcionalmente, el valor inicial del atributo.
- **Expresión:** para los atributos derivados, en este apartado se indica la expresión OCL asociada a la evaluación del atributo.

4.7.1.3 Plantillas de enlaces

Los enlaces se tratan de una forma similar a los atributos. Los formatos de las plantillas para describirlos pueden verse en la figura 4.22. Sus apartados específicos son los siguientes:

- **Nombre enlace:** cada enlace se identifica por un nombre único dentro de un tipo. Dicho nombre deberá coincidir con el nombre del rol que juegue el tipo en la asociación de la que proviene el enlace. Se recomienda que el nombre sea un sustantivo o similar.
- **Descripción:** en este apartado se proporciona una breve descripción del enlace.
- **Asociación:** en este apartado se indica la asociación de la que proviene el enlace.
- **Tipo OCL:** en este apartado se indica el tipo OCL del enlace. Dicho tipo deberá coincidir con el tipo del rol del mismo nombre que el enlace en la asociación de la que proviene el enlace.

Enlace constante	<i><nombre tipo>::<i>nombre enlace</i>></i>
Descripción	<i><descripción del enlace></i>
Tipo OCL	<i><tipo OCL del enlace></i>
Asociación	<i><asociación de la que proviene el enlace></i>
Comentarios	<i><comentarios adicionales sobre el enlace></i>

Enlace variable	<i><nombre tipo>::<i>nombre enlace</i>></i>
Descripción	<i><descripción del enlace></i>
Tipo OCL	<i><tipo OCL del enlace></i>
Valor inicial	<i><valor inicial del enlace> [opcional]</i>
Asociación	<i><asociación de la que proviene el enlace></i>
Comentarios	<i><comentarios adicionales sobre el enlace></i>

Enlace derivado	<i><nombre tipo>::<i>nombre enlace</i>></i>
Descripción	<i><descripción del enlace></i>
Tipo OCL	<i><tipo OCL del enlace></i>
Expresión	<i><expresión OCL asociada al enlace></i>
Asociación	<i><asociación de la que proviene el enlace></i>
Comentarios	<i><comentarios adicionales sobre el enlace></i>

Figura 4.22: Plantillas de descripción de enlaces

- **Valor inicial:** para los enlaces variables, en este apartado se indica, opcionalmente, el valor inicial del enlace.
- **Expresión:** para los enlaces derivados, en este apartado se indica la expresión OCL asociada a la evaluación del enlace.

4.7.1.4 Plantilla de invariante

El formato de la plantilla para describir la invariante de los tipos de objetos puede verse en la figura 4.23. Los apartados específicos de esta plantilla son los siguientes:

Invariante	<i><nombre tipo></i>
Descripción	• <i><descripción de la invariante en lenguaje natural></i>
Expresión	<i><expresión OCL de la invariante></i>
Comentarios	<i><comentarios adicionales sobre la invariante></i>

Figura 4.23: Plantilla de descripción de invariante de tipo

- **Descripción:** en este apartado se proporciona una descripción en lenguaje natural de la invariante para facilitar la comprensión de la expresión OCL del apartado siguiente.
- **Expresión:** en este apartado se indica la expresión OCL que especifica la invariante del tipo, asumiendo el contexto del propio tipo [Rat 1997, pág. 3].

En esta plantilla se incluirán explícitamente aquellas invariantes relativas a los atributos que representa los estados y que vienen dadas por la estructura del statechart, tal como se comentó en la sección 4.4.3.2.

4.7.1.5 Plantilla de asociación

El formato de la plantilla para describir las asociaciones, entre tipos de objetos puede verse en la figura 4.24. Los apartados específicos de esta plantilla son los siguientes:

- **Nombre asociación y participantes:** cada asociación se identifica por una combinación de nombre de asociación y de nombres de participantes única en el modelo. Se recomienda que el nombre sea una

Asociación	<nombre asociación> entre <participantes>
Versión	<nº de la versión actual> (<fecha de la versión actual>)
Autores	• <autor de la versión actual> (<organización del autor>) ...
Requisitos asociados	• RI-x <nombre del requisito> ...
Descripción	Esta asociación representa <concepto que representa la asociación>
Comentarios	<comentarios adicionales sobre la asociación>

Asociación derivada	<nombre asociación> entre <participantes>
Versión	<nº de la versión actual> (<fecha de la versión actual>)
Autores	• <autor de la versión actual> (<organización del autor>) ...
Requisitos asociados	• RI-x <nombre del requisito> ...
Descripción	Esta asociación derivada representa <concepto que representa la asociación>
Comentarios	<comentarios adicionales sobre la asociación>

Figura 4.24: Plantillas de descripción de asociación

forma verbal o similar, de forma que tenga sentido al construir una frase con el nombre de la asociación y los participantes. Por ejemplo la asociación *trabajaEn* entre *Empleado* y *Empresa*.

Para hacer referencia a la asociación se utilizará el formato:

asociación(A, B, ..., C)

por ejemplo: *trabajaEn*(*Empleado*, *Empresa*)

- **Descripción:** en este apartado se describe el concepto que representa la asociación y si se trata de una asociación derivada o no.

En el caso de las asociaciones derivadas, deberá acompañarse una descripción de su invariante en la que se especifique el criterio para saber qué objetos están asociados. La estructura de esta invariante suele ser la siguiente, donde $P_1 \dots P_n$ son los tipos de los objetos participantes en la asociación, y A es la asociación derivada:

```
P1.allInstances->forAll( p1 |
...
Pn.allInstances->forAll( pn |
  A.allInstances->forAll( a |
    ( a.rol1 = p1 and ... and a.roln = pn ) = < expresión sobre p1 ... pn > )
```

4.7.1.6 Plantilla de rol

El formato de la plantilla para describir los roles de las asociaciones entre tipos de objetos puede verse en la figura 4.25. Los apartados específicos de esta plantilla son los siguientes:

Rol	< <i>nombre tipo</i> > juega rol < <i>nombre rol</i> > en < <i>nombre asociación</i> >
Descripción	< <i>descripción del rol</i> >
Tipo OCL	< <i>tipo OCL del rol</i> >
Multiplicidad	< <i>multiplicidad del rol</i> >
Comentarios	< <i>comentarios adicionales sobre el rol</i> >

Figura 4.25: Plantilla de descripción de rol

- **Nombre rol:** cada rol se identifica por una combinación de nombre de asociación y de nombre de rol única en el modelo, siempre que no haya posibilidad de ambigüedades [Rat 1997, pág. 8]. Se recomienda que el nombre sea un sustantivo o similar.
- **Descripción:** en este apartado se proporciona una breve descripción del rol.
- **Tipo OCL:** en este apartado se especifica el tipo OCL del rol.
- **Multiplicidad:** en este apartado se indica la multiplicidad del rol.

4.7.2 Plantilla para especificación de operaciones de sistema

En la propuesta metodológica para el análisis de requisitos se recomienda utilizar las *acciones conjuntas* de Catalysis aunque denominándolas *operaciones de sistema* como en Fusion, ya que las acciones conjuntas que se especifican son las que forma la interfaz del sistema, tal como se comentó en la sección 4.4.2.3, pág. 150.

Para el estilo de especificación se ha optado por uno similar al utilizado en las especificaciones realizadas en Z [Diller 1990, Ratcliff 1994] y al propuesto en [Iturrioz 1998] o en [D'Souza y Wills 1999], en el que la semántica de las operaciones se especifica mediante disyunciones en las que aparecen la conjunción de la precondition y de la postcondition, o bien la

conjunción de una expresión de excepción y la postcondición en dicho caso. Esto permite especificar de forma clara la conducta del sistema, tanto en situaciones normales como en situaciones de error.

Para poder especificar la respuesta del sistema se utilizan dos *pseudovariables*: *respuesta*, de tipo OCL Set(String), donde se le comunican al usuario las respuestas del sistema y, para aquellas operaciones que sean de consulta o que devuelvan algún tipo de resultado, *resultado*¹¹, cuyo tipo depende de la consulta que se realice.

Siguiendo estas ideas, el esquema (en la notación de Catalysis) para especificar una operación de sistema que no sea de consulta es el que puede verse en la figura 4.26, donde p_1, \dots, p_q son los parámetros de la operación, *pre* es la precondición, *post* es la postcondición, $excep_1, \dots, excep_n$ son las condiciones de excepción y la respuesta del sistema se devuelve en *respuesta*.

```
acción Operación(  $p_1, \dots, p_q$  )
post: ( pre and post and respuesta = Set{ "<éxito>" } )
    or
    (  $excep_1$  and respuesta->includes( "<error nº 1>" ) )
    or
    ...
    or
    (  $excep_n$  and respuesta->includes( "<error nº n>" ) )
```

Figura 4.26: Esquema para la especificación de operaciones de sistema

Este esquema asume que la disyunción de la precondición y de las condiciones de excepción es siempre cierta y que la precondición no puede ser cierta si alguna condición de excepción también lo es. Es decir:

```
pre xor (  $excep_1$  or ... or  $excep_n$  )
```

En el caso de una operación de consulta (ver figura 4.27), el esquema es similar, aunque en este caso ha sido necesario ampliar OCL para que permita la utilización de *tipos sin nombre* para poder especificar los resultados de las consultas, que se devuelven usando *resultado*. Esta ampliación de OCL se comenta con más detalle en la sección 4.8.2, pág. 176.

La forma de expresar este estilo de especificación de operaciones es mediante la plantilla que puede verse en la figura 4.28, cuyos apartados específicos son los siguientes:

¹¹Denominada *result* en la documentación original de OCL [Rat 1997].

```

acción Consulta(  $p_1, \dots p_q$  ) : <tipo del resultado>
post: ( pre and
    resultado = <resultado de la consulta> and
    respuesta = Set{ "<éxito>" } )
or
    ( excep1 and respuesta->includes( "<error nº 1>" ) )
or
    ...
or
    ( excepn and respuesta->includes( "<error nº n>" ) )

```

Figura 4.27: Esquema para la especificación de operaciones de consulta al sistema

- **Nombre operación:** cada operación de sistema se identifica por un nombre único en el modelo que se recomienda que sea una forma verbal o similar.
- **Tipo del resultado:** en el caso de que la operación devuelva algún tipo de resultado, es necesario especificarlo en este apartado opcional.
- **Descripción:** en este apartado se proporciona una breve descripción de la operación en lenguaje natural.
- **Requisitos asociados:** en este apartado se indican los requisitos del *Documento de Requisitos del Sistema* (DRS) a los que está asociado la operación del sistema, es decir, aquellos requisitos que justifican su existencia, normalmente requisitos funcionales expresados como casos de uso.
- **Parámetros:** en este apartado se deben enumerar los parámetros de la operación siguiendo la sintaxis de OCL con el formato:

nombre : tipo - descripción

siendo:

- **nombre:** nombre del parámetro.
- **tipo:** tipo OCL del parámetro
- **descripción:** descripción del parámetro.

Operación Sistema	<nombre operación>
Tipo resultado	tipo del resultado [opcional]
Versión	<nº de la versión actual> (<fecha de la versión actual>)
Autores	• <autor de la versión actual> (<organización del autor>) ...
Requisitos asociados	• RF-x <nombre del requisito> ...
Descripción	<descripción de la operación en lenguaje natural>
Parámetros	p ₁ : Tipo ₀₁ -- descripción p ₁ ... p _q : Tipo _n -- descripción p _q
Precondiciones	pre ₁ : <descripción de pre ₁ en lenguaje natural> ... pre _n : <descripción de pre _n en lenguaje natural>
Precondiciones (OCL)	pre ₁ : <descripción de pre ₁ en OCL> ... pre _n : <descripción de pre _n en OCL>
Postcondiciones	post ₁ : <descripción de post ₁ en lenguaje natural> ... post _m : <descripción de post _m en lenguaje natural>
Postcondiciones (OCL)	post ₁ : <descripción de post ₁ en OCL> ... post _m : <descripción de post _m en OCL>
Excepciones	excep ₁ : <postcond. si se da excep ₁ en lenguaje natural> ... excep _n : <postcond. si se da excep _n en lenguaje natural>
Excepciones (OCL)	excep ₁ : <postcond. si se da excep _n en OCL> ... excep _n : <postcond. si se da excep _n en OCL>
Comentarios	<comentarios adicionales sobre la operación>

Figura 4.28: Plantilla de descripción de operación de sistema

- **Precondiciones:** en este apartado se describen en lenguaje natural las precondiciones de la operación. Para facilitar la legibilidad y hacer patente la relación entre las precondiciones en lenguaje natural y las expresadas en OCL, se ha optado por etiquetar cada frase de las que forman la precondición. La precondición de la operación será la conjunción de todas las expresiones (precondiciones) etiquetadas.
- **Precondiciones (OCL):** en este apartado se describen en OCL las precondiciones de la operación, definidas previamente en lenguaje natural. Cada expresión en OCL asociada con una frase de la precondición expresada en lenguaje natural debe tener la misma etiqueta.
- **Postcondiciones:** en este apartado se describen en lenguaje natural las postcondiciones de la operación en el caso de que todas las expresiones de la precondición sean ciertas. Al igual que en el caso de las precondiciones, la postcondición de la operación será la conjunción de todas las postcondiciones y cada frase se etiquetará de forma única.
- **Postcondiciones (OCL):** en este apartado se describen en OCL las postcondiciones de la operación, previamente descritas en lenguaje natural, en el caso de que todas las expresiones de la precondición sean ciertas. Al igual que en el caso de las precondiciones, cada expresión OCL se etiquetará coincidiendo con su expresión en lenguaje natural.
- **Excepciones:** en este apartado se describen en lenguaje natural las postcondiciones de la operación en el caso de que produzca alguna condición de excepción. Para expresar las condiciones de excepción se puede recurrir a las etiquetas de las precondiciones previamente definidas, por ejemplo: $\neg pre_i \wedge \neg pre_j$ o $\neg pre_i \vee \neg pre_j$.
- **Excepciones (OCL):** en este apartado se describen en OCL las postcondiciones de la operación en el caso de que produzca alguna condición de excepción. Al igual que en el apartado anterior, para expresar las condiciones de excepción se puede recurrir a las etiquetas de las precondiciones previamente definidas, con lo que se facilita la legibilidad de las expresiones.

4.8 Patrones de reutilización de requisitos-D

En el capítulo anterior se describieron algunos patrones de reutilización a nivel de requisitos-C, los patrones- R_C . Este tipo de patrones es posible también identificarlos en los requisitos-D siguiendo las relaciones de rastreabilidad establecidas, es decir, aquellos requisitos-D que modelen los patrones- R_C serán los patrones-R de requisitos-D correspondientes, abreviadamente *patrones- R_D* .

4.8.1 Patrones- R_D para requisitos de almacenamiento de información

Para los patrones- R_C de requisitos de almacenamiento de información, los patrones- R_D estarán formadas por aquellos tipos, con sus correspondientes atributos y asociaciones, que modelen la información que describen.

Por ejemplo, para modelar el patrón- R_C *Cliente/Socio* (ver figura 3.19, pág. 98) se tendría el tipo *Cliente/Socio* que puede verse en la figura 4.29 asociado al patrón- R_C mediante la correspondiente relación de rastreabilidad.

Aunque en el ejemplo la relación entre patrón- R_C y patrón- R_D es 1:1, no siempre tiene porqué ser el caso. Puede que la estructura de rastreabilidad sea más compleja y tenga la forma de una relación $m:n$, de forma que un grupo de requisitos-C pueda rastrearse hasta un grupo de requisitos-D, o dicha de otra forma, que un grupo de requisitos-C pueda rastrearse hasta un *framework* [D'Souza y Wills 1999].

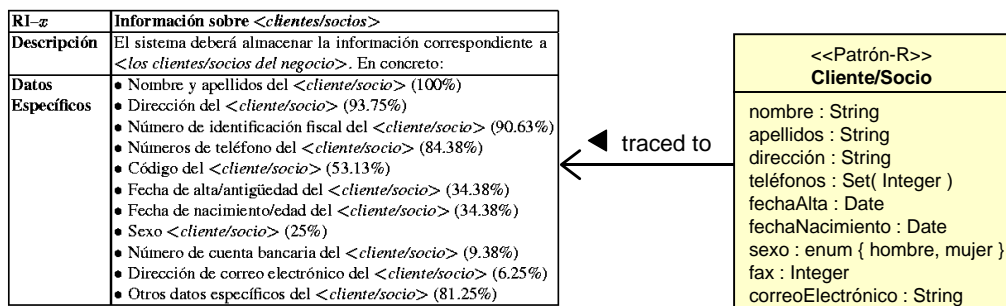


Figura 4.29: Patrón- R_D *Cliente/Socio*

4.8.2 Patrones- R_D para requisitos funcionales

Para los patrones- R_C de requisitos funcionales descritos en la sección 3.7.2, los patrones- R_D estarán compuestos por descripciones de operaciones del sistema que modelen la funcionalidad descrita en los casos de uso.

En estas descripciones se utilizarán las siguientes abreviaturas, dado un tipo de objetos T:

$$\begin{aligned} T.new &= (T.allInstances - T.allInstances@pre) \\ T.dead &= (T.allInstances@pre - T.allInstances) \end{aligned}$$

es decir, T.new es el conjunto de los objetos que se crean y T.dead es el conjunto de los objetos que se destruyen durante la operación. En el caso de T.new, esta notación es la misma que la utilizada en la versión de OCL propuesta en [D'Souza y Wills 1999].

Operación Sistema	{Crear, DarDeAlta, Registrar}<X>
Descripción	{Crea, Da de alta, Registra} un nuevo <X>
Parámetros	p_1 : Tipo ₁ ... p_q : Tipo _q k_1 : Tipo _k -- parámetro clave ... k_r : Tipo _r -- parámetro clave
Precondiciones	pre₁ : No existe un <X> en el sistema con el mismo valor de k_1 , ..., k_r
Precondiciones (OCL)	pre₁ : not <X>.allInstances->exists(x x.k ₁ = k_1 and ... and x.k _r = k_r)
Postcondiciones	post₁ : Existe un nuevo <X> cuyos atributos coinciden con los parámetros post₂ : Sólo se ha creado un nuevo <X> post₃ : El sistema informa de que el proceso ha terminado con éxito
Postcondiciones (OCL)	post₁ : <X>.new->exists(x x.p _i = p_i and x.k _j = k_j) -- i:1..q, j:1..r post₂ : <X>.new->size = 1 post₃ : respuesta = Set{"<X> { creado, dado de alta, registrado } con éxito"}
Excepciones	pre₁ : El sistema informa de que ya existe un <X> en el sistema con los mismos datos
Excepciones (OCL)	not pre₁ : respuesta ->includes("Error: <X> ya existente")

Figura 4.30: Patrón- R_D para la operación de sistema *Crear<X>*

4.8.2.1 Patrón- R_D *Crear/DarDeAlta/Registrar*

El patrón- R_D correspondiente al patrón- R_C *Crear/Dar de alta/Registrar* (pág. 100) puede verse en la figura 4.30.

Este patrón admite diversas variaciones. Por ejemplo, puede que no todos los atributos del objeto que se crea se pasen como parámetros, algunos pueden tomar un determinado valor inicial. También es posible que no se hayan identificado atributos clave para el tipo de objetos en cuestión, en cuyo caso no habría ninguna precondition. Otra posibilidad sería crear más de un objeto a la vez, bien varios de tipo X , bien uno de tipo X y otros de otros tipos que tengan alguna asociación con el tipo X .

4.8.2.2 Patrón- R_D *Eliminar/DarDeBaja/Cancelar*

El patrón- R_D correspondiente al patrón- R_D *Eliminar/Dar de baja/Cancelar* (pág. 101) puede verse en la figura 4.31.

Al igual que en el patrón- R_D anterior, éste también admite variaciones sobre el número y los tipos de objetos a destruir.

Operación Sistema	{Eliminar, DarDeBaja, Cancelar}<X>
Descripción	{Elimina, Da de baja, Cancela} un <X>
Parámetros	$x : X$ -- el <X> a eliminar ...
Precondiciones	pre₁ : x <cumple las condiciones para poder ser eliminado>
Precondiciones (OCL)	pre₁ : x <cumple las condiciones para poder ser eliminado>
Postcondiciones	post₁ : x ha dejado de existir post₂ : Sólo se ha destruido un <X> post₃ : El sistema informa de que el proceso ha terminado con éxito
Postcondiciones (OCL)	post₁ : $\langle X \rangle.\text{dead} \rightarrow \text{includes}(x)$ post₂ : $\langle X \rangle.\text{dead} \rightarrow \text{size} = 1$ post₃ : respuesta = Set{"<X> { eliminado, dado de baja, cancelado } con éxito"}
Excepciones	pre₁ : El sistema informa de el <X> no puede ser eliminado
Excepciones (OCL)	not pre₁ : respuesta $\rightarrow \text{includes}$ ("Error: <X> no puede ser eliminado")

Figura 4.31: Patrón- R_D para la operación de sistema *Eliminar<X>*

4.8.2.3 Patrón- R_D *Modificar/Editar*

El patrón- R_D correspondiente al patrón- R_C *Modificar/Editar* (pág. 102) puede verse en la figura 4.32.

Al igual que los dos patrones- R_D anteriores, este patrón admite varias alternativas. Por ejemplo, crear operaciones para modificar determinada información sobre el objeto, en lugar de tener una única operación para modificar todas las características modificables del objeto, o permitir la modificación de cierta información del objeto en función del tipo de usuario que realice la operación.

Operación Sistema	{<i>Modificar, Editar</i>}<X>
Descripción	{Modifica, Edita } un <X>
Parámetros	<i>x</i> : <i>X</i> -- el <X> a eliminar <i>p</i> ₁ : Tipo ₁ ... <i>p</i> _q : Tipo _q
Precondiciones	pre ₁ : <los parámetros no violan ninguna restricción> pre ₂ : <i>x</i> <cumple las condiciones para poder ser modificado>
Precondiciones (OCL)	pre ₁ : not < <i>p</i> ₁ ... <i>p</i> _q violan alguna restricción>
Postcondiciones	post ₁ : Los atributos de <i>x</i> coinciden con los parámetros post ₂ : El sistema informa de que el proceso ha terminado con éxito
Postcondiciones (OCL)	post ₁ : <i>x.p</i> ₁ = <i>p</i> ₁ and ... <i>x.p</i> _q = <i>p</i> _q post ₂ : respuesta = Set{"<X> { modificado, editado } con éxito"}
Excepciones	pre ₁ : El sistema informa de que los nuevos valores no son aceptables pre ₂ : El sistema informa de que el <X> no puede ser modificado
Excepciones (OCL)	not pre ₁ : respuesta ->includes("Error: los nuevos valores no son aceptables") not pre ₂ : respuesta ->includes("Error: <X> no puede ser modificado")

Figura 4.32: Patrón- R_D para la operación de sistema *Modificar<X>*

4.8.2.4 Patrón- R_D *Consultar*

El patrón- R_D correspondiente al patrón- R_C *Consultar* (pág. 103) puede verse en la figura 4.33.

Este es probablemente el patrón- R_D que más variaciones puede presentar, ya que cada consulta específica tendrá sus propios parámetros (o incluso ninguno) y tendrá un criterio de selección distinto.

Operación Sistema	<i>Consultar</i> <X>
Tipo resultado	Collection(<X>)
Descripción	Consulta los datos relativos a <X>
Parámetros	q_1 : Tipo $_1$ -- parámetro de la consulta ... q_n : Tipo $_n$ -- parámetro de la consulta
Postcondiciones	post $_1$: El resultado son los <X> que cumplen el criterio de consulta post $_2$: El sistema informa de que el proceso ha terminado con éxito
Postcondiciones (OCL)	post $_1$: resultado = <X>.allInstances->select(<criterio usando q_1 ... q_n >) post $_2$: respuesta = Set{"Consulta terminada con éxito"}

Figura 4.33: Patrón- R_D para la operación de sistema *Consultar*<X>

En general, el objetivo del análisis respecto a las consultas es asegurar que las consultas pueden realizarse, y la expresividad de un lenguaje como OCL no siempre permite especificar una consulta de forma completa.

Si después de analizar el caso de uso en el que se especifica qué información debe devolver el sistema como respuesta a la consulta, se puede comprobar que el modelo estático contempla dicha información y que dicha información es accesible, el objetivo del análisis está cubierto.

En [D'Souza y Wills 1999, pág. 598] se propone realizar diagramas de instancias (denominados *snapshots* o *instantáneas* en [D'Souza y Wills 1999]) representando la información a obtener del sistema. En nuestra opinión, es necesario ampliar OCL para permitir la definición de tipos sin nombre que puedan utilizarse para especificar los resultados de las consultas.

Por ejemplo, supongamos un modelo sencillo como el de la figura 4.34 para el que es necesario especificar una operación de consulta que genere un listado de todos los socios, indicando para cada socio los alquileres realizados. El listado debe estar ordenado por nombre de socio y por fecha de alquiler.

Para expresar esta consulta en OCL es necesario indicar que el resultado será de tipo Sequence(String x Sequence(String x Date)), que siguiendo estrictamente [Rat 1997], sólo podría expresarse incluyendo en el modelo los tipos necesarios para ello. En el caso de las consultas, esto implicaría tener que definir al menos un tipo de objetos por cada consulta que se deseara especificar¹².

¹²Esta es la opción elegida en [Troyano 1998] para usar el lenguaje de especificación Tesoro.

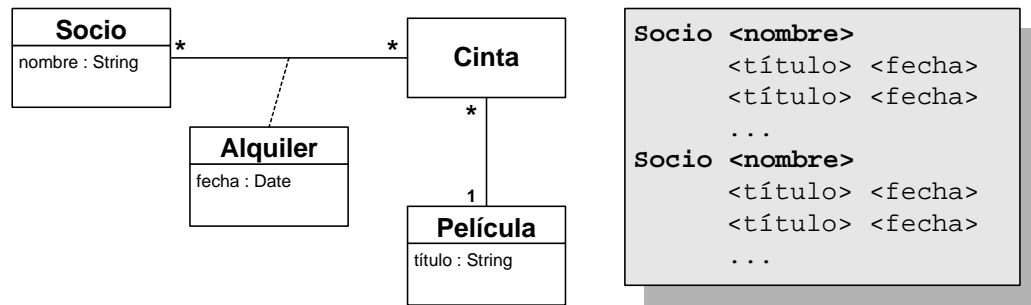


Figura 4.34: Diagrama de tipos para el ejemplo de consulta en OCL

Como ya se comentó en la sección 4.7.2, se ha ampliado OCL para permitir formar *tipos sin nombre*, de forma que se permitan expresiones de la forma (exp_1, \dots, exp_n) , cuyo tipo sería $(T_1 \times \dots \times T_n)$, siendo cada T_i el tipo correspondiente a cada exp_i .

Además de permitir tipos sin nombre, se ha considerado también la posibilidad de poder nombrar los distintos elementos de dichos tipos, de forma similar a los esquemas en Z [Diller 1990, Ratcliff 1994]. De esta forma, el tipo del resultado de la consulta de la figura 4.34 se expresaría como:

Sequence(nombre:String x alquileres:Sequence(título:String x fecha:Date))

Con lo que la consulta podría especificarse de la siguiente forma:

acción ObtenerListadoSociosAlquileres() :
 Sequence(nombre:String x alquileres:Sequence(título:String x fecha:Date))
post: **resultado**->isOrderedBy(<=, nombre) **and**
 resultado->forAll(r | r.alquileres->isOrderedBy(<=, fecha)) **and**
 resultado->collect(nombre) = Socio.allInstances->collect(nombre) **and**
 resultado->forAll(r | Socio.allInstances->exists(s |
 s.nombre = r.nombre **and**
 r.alquileres->collect(título, fecha) =
 s.alquiler->collect(cinta.película.título, fecha))) **and**
respuesta = Set{"Consulta realizada con éxito"}

donde isOrderedBy es una *macro* definida sobre secuencias de la forma:

Sequence(T)::isOrderedBy(op, exp) : Boolean
resultado = Sequence{1..**self**->size - 1}->forAll(i |
 self->at(i).exp op **self**->at(i + 1).exp)

y se ha permitido que la operación *collect* sobre colecciones pueda manejar también tipos sin nombre.

Si se hubiera querido restringir de alguna forma el resultado filtrando los socios o los alquileres, podría haberse hecho mediante el uso de la operación *select*, definida sobre colecciones, tal como se muestra en el patrón- R_D de la figura 4.33.

4.9 Conclusiones

En este capítulo se ha presentado la propuesta metodológica para el análisis de requisitos dentro del marco del modelo de procesos descrito en el capítulo 2. Para llegar a dicha propuesta se han analizado las técnicas habituales de análisis de requisitos, tanto estructuradas como orientadas a objetos.

Este análisis nos ha hecho coincidir con otros autores como [Wieringa 1998] en la idea de que tanto las técnicas estructuradas modernas, basadas en el particionamiento de eventos e integrando las tres dimensiones de modelado, como las técnicas orientadas a objetos de segunda generación (incluyendo las técnicas *post-UML* como Catalysis) tienen un modelo subyacente de sistema de información muy similar.

Para la propuesta metodológica se ha optado por la utilización de las técnicas orientadas a objetos por su mayor expresividad, integrando aspectos formales como OCL de forma similar a la descrita en Catalysis. Al igual que en el caso de los requisitos- C , también se han identificado algunos patrones- R a nivel de requisitos- D , relacionados con los anteriores a través de las relaciones de rastreabilidad que se comentaron en el capítulo anterior.

Durante la aplicación de las ideas expuestas en este artículo en prácticas académicas, pudimos constatar las deficiencias del modelo de paso de mensajes para el modelado conceptual de sistemas de información y como su posterior sustitución por un modelo más abstracto, en el que las operaciones se especificaban únicamente a nivel de sistema sobre el estado global, mejoró sensiblemente la calidad y facilidad de comprensión de los modelos resultantes.

Es necesario destacar la falta de técnicas para el análisis de los requisitos no funcionales, que son prácticamente ignorados por las técnicas de modelado conceptual y relegados para las fases posteriores de desarrollo

[Bass *et al.* 1998, Chung *et al.* 1999, Ruiz *et al.* 2000]. Resulta evidente la necesidad de un mayor esfuerzo de investigación para la clasificación y modelado de este tipo de requisitos, especialmente complejos por su heterogeneidad.

4.10 Bibliografía

- [Bass *et al.* 1998] L. Bass, P. Clements, y R. Kazman. Nonfunctional Requirements Is a Dysfunctional Term. En *Software Architecture in Practice*, páginas 76–77. Addison–Wesley, 1998.
- [Booch *et al.* 1999] G. Booch, J. Rumbaugh, y I. Jacobson. *The Unified Modeling Language User Guide*. Addison–Wesley, 1999.
- [Booch 1991] G. Booch. *Object–Oriented Design with Applications*. Benjamin/Cummings, 1991.
- [Booch 1994] G. Booch. *Object–Oriented Analysis and Design with Applications*. Benjamin/Cummings, 2ª edición, 1994.
- [Brackett 1990] J. W. Brackett. Software Requirements. Curriculum Module SEI–CM–19–1.2, Software Engineering Institute, Carnegie Mellon University, 1990. Disponible en <http://www.sei.cmu.edu>.
- [CCT 1990] CCTA. *SSADM Version 4.2*, 1990.
- [Chen 1976] P. P.-S. Chen. The Entity–Relationship Model — Toward a Unified View of Data. *ACM Transactions on Database Systems*, 1:9–36, 1976. También aparece en [Thayer y Dorfman 1990].
- [Chung *et al.* 1999] L. Chung, D. Gross, y E. Yu. Architectural Desing to Meet Stakeholder Requirements. En *Proceedings of the First Working IFIP Conference on Software Architecture (WICSA1)*, San Antonio, (Texas, USA), 1999.
- [Coad y Yourdon 1990] P. Coad y E. Yourdon. *Object–Oriented Analysis*. Yourdon Press/Prentice–Hall, 1990.
- [Codd 1970] E. F. Codd. A Relational Model of Data for Large Shared Data Banks. *Communications of the ACM*, 13(6), 1970.

- [Coleman *et al.* 1994] D. Coleman, P. Arnold, S. Bodoff, C. Dollin, H. Gilchrist, F. Hayes, y P. Jeremaes. *Object-Oriented Development. The Fusion Method*. Prentice-Hall, 1994.
- [Cook y Daniels 1994] S. Cook y J. Daniels. *Designing Object Systems: Object-Oriented Modelling with Syntropy*. The Object-Oriented Series. Prentice-Hall, 1994.
- [Corchuelo *et al.* 1998] R. Corchuelo, A. Durán, J. Pérez, y M. Toro. ¿Desarrollo OO de Software Utilizando Herramientas RAD? En *Actas de las III Jornadas de Ingeniería del Software*, Murcia, 1998.
- [Corchuelo *et al.* 2000] R. Corchuelo, D. Ruiz, M. Toro, y A. Durán. Avances en la Coordinación de Objetos Activos. *Novática*, (143), Enero-Febrero 2000.
- [Corchuelo 1999] R. Corchuelo. *Prototipado de Especificaciones de Sistemas Distribuidos Basadas en Retricciones. Aplicación al Lenguaje TESORO*. Tesis doctoral, Universidad de Sevilla, 1999.
- [Cysneiros y Leite 1999] L. M. Cysneiros y J. C. S. P. Leite. Integrating Non-Functional Requirements into Data Modeling. En *Proceedings of the Fourth International Symposium on Requirements Engineering*, 1999.
- [Date 1995] C. J. Date. *An Introduction to Database Systems*. Addison-Wesley, 6ª edición, 1995.
- [Davis 1995] A. M. Davis. Tracing: A Simple Necessity Neglected. *IEEE Software*, 12(5), Septiembre 1995.
- [Diller 1990] A. Diller. *Z: An Introduction to Formal Methods*. John Willey, 1990.
- [DoD 1994] DoD. *Military Standard 498: Software Development and Documentation*. Departament of Defense of the United States of America, 1994. Disponible en http://www-library.itsi.disa.mil/mil_std/498_win3.exe.
- [Dorfman y Thayer 1990] M. Dorfman y R. H. Thayer, editores. *Standards, Guidelines, and Examples on System and Software Requirements Engineering*. IEEE Computer Society Press, 1990.
- [D'Souza y Wills 1999] D. F. D'Souza y A. C. Wills. *Objects, Components, and Frameworks with UML: The Catalysis Approach*. Addison-Wesley, 1999.

- [D'Souza 1996] D. F. D'Souza. Behavior-Driven vs. Data-Driven: *A Non-Issue?* Informe técnico, Platinum Technology, 1996. Disponible en <http://www.iconcomp.com/papers>.
- [Franch 1998] X. Franch. Systematic Formulation of Non-Functional Characteristics of Software. En *Proceedings of the 3rd IEEE International Conference on Requirements Engineering*, Colorado Springs (Colorado, USA), 1998. Disponible en <http://www.upc.es>.
- [García 2000] F. J. García. *Modelo de Reutilización Soportado por Estructuras Complejas de Reutilización Denominadas Mecanos*. Tesis doctoral, Universidad de Salamanca, 2000.
- [Garg 1990] V. K. Garg. Modelling of Distributed Systems by Concurrent Regular Expressions. En *Formal Description Techniques*, volumen 2, páginas 313-327. North Holland, 1990.
- [Gomaa 1990] H. Gomaa. The Impact of Prototyping on Software System Engineering. En Thayer y Dorfman [1990], páginas 543-552. Aparece también en [Thayer y Dorfman 1997].
- [Harel 1987] D. Harel. Statecharts: A Visual Formalism for Complex Systems. *Science of Computer Programming*, 8:231-274, 1987.
- [Iturrioz 1998] J. I. Iturrioz. *Una Metodología para el Desarrollo de Sistemas de Bases de Datos Objeto-Relacionales*. Tesis doctoral, Universidad del País Vasco, 1998.
- [Jacobson et al. 1993] I. Jacobson, M. Christerson, P. Jonsson, y G. Övergaard. *Object-Oriented Software Engineering: A Use Case Driven Approach*. Addison-Wesley, 4ª edición, 1993.
- [Kontoya y Sommerville 1996] G. Kontoya y I. Sommerville. Requirements Engineering with Viewpoints. *Software Engineering Journal*, 11(1), Enero 1996.
- [Kovitz 1998] B. L. Kovitz. *Practical Software Requirements: A Manual of Content & Style*. Manning, 1998.
- [Leite y Freeman 1991] J. C. S. P. Leite y P. A. Freeman. Requirements Validation through Viewpoint Resolution. *IEEE Transactions on Software Engineering*, 17(12), Diciembre 1991.

- [Letelier *et al.* 1998] P. Letelier, P. Sánchez, I. Ramos, y O. Pastor. *OASIS 3.0: Un Enfoque Formal para el Modelado Conceptual Orientado a Objeto*. Servicio de Publicaciones de la Universidad Politécnica de Valencia, 1998. Disponible en <http://www.dsic.upv.es/users/oom/oasis.html>.
- [MAP 1995a] MAP. *Metodología de Planificación y Desarrollo de Sistemas de Información. MÉTRICA Versión 2.1*. Tecnos/Ministerio para las Administraciones Públicas, 1995.
- [MAP 1995b] MAP. *Metodología de Planificación y Desarrollo de Sistemas de Información. MÉTRICA Versión 2.1: Guía de Técnicas*. Tecnos/Ministerio para las Administraciones Públicas, 1995.
- [Mazza *et al.* 1996] C. Mazza, J. Fairclough, B. Melton, D. de Pablo, A. Scheffer, R. Stevens, M. Jones, y G. Alvisi. *Software Engineering Guides*. Prentice-Hall, 1996.
- [Nijssen y Halpin 1989] G. M. Nijssen y T. A. Halpin. *Conceptual Schema and Relational Database Design*. Prentice-Hall, 1989.
- [Nuseibeh *et al.* 1994] B. Nuseibeh, J. Kramer, y A. Finkelstein. A Framework for Expressing the Relationships between Multiple Views in Requirements Specification. *IEEE Transactions on Software Engineering*, 20(10), Octubre 1994.
- [Pastor y Ramos 1995] O. Pastor y I. Ramos. *OASIS version 2 (2.2): A Class-Definition Language to Model Information Systems Using an Object-Oriented Approach*. Servicio de Publicaciones de la Universidad Politécnica de Valencia, 1995.
- [Piattini *et al.* 1996] M. G. Piattini, J. A. Calvo-Manzano, J. Cervera, y L. Fernández. *Análisis y Diseño Detallado de Aplicaciones Informáticas de Gestión*. ra-ma, 1996.
- [Pohl 1994] K. Pohl. The Three Dimensions of Requirements Engineering: A Framework and its Application. *Information Systems*, 3(19), Junio 1994.
- [Pohl 1997] K. Pohl. Requirements Engineering: An Overview. *Encyclopedia of Computer Science and Technology*, 36, 1997. Disponible en <http://sunsite.informatik.rwth-aachen.de/CREWS/reports96.htm>.

- [Pollack *et al.* 1993] F. Pollack, M. Whiston, y K. C. Mander. The SAZ Project: Integrating SSADM and Z. En J. C. P. Woodcock y P. G. Larson, editores, *FME'93: Industrial-Strength Formal Methods*, volumen 670 de *Lecture Notes in Computer Science*. Springer Verlag, 1993.
- [Rat 1997] Rational Software Corporation. *Object Constraint Language Specification*, 1.1 edición, Septiembre 1997. Disponible en <http://www.rational.com>.
- [Ratcliff 1994] B. Ratcliff. *Introducing Specification Using Z: A Practical Case Study Approach*. International Series in Software Engineering. McGraw-Hill, 1994.
- [Rombach 1990] H. D. Rombach. Software Specifications: A Framework. Curriculum Module SEI-CM-11-2.1, Software Engineering Institute, Carnegie Mellon University, 1990. No está disponible en <http://www.sei.cmu.edu>, aunque aparece en [Dorfman y Thayer 1990].
- [Ruiz *et al.* 2000] A. Ruiz, R. Corchuelo, A. Durán, O. Martín, y J. A. Pérez. Prototipado Arquitectónico de Sistemas Abiertos Distribuidos. En *Actas de las V Jornadas de Trabajo Menhir*, Granada, 2000.
- [Rumbaugh *et al.* 1991] J. Rumbaugh, M. Blaha, W. Premerlani, F. Eddy, y W. Lorensen. *Object-Oriented Modeling and Design*. Prentice-Hall, 1991.
- [Sawyer y Kontoya 1999] P. Sawyer y G. Kontoya. SWEBOK: Software Requirements Engineering Knowledge Area Description. Informe Técnico Versión 0.5, SWEBOK Project, 1999. Disponible en <http://www.swebok.org>.
- [Sommers y Allen 1991] L. Sommers y P. Allen. Using Yourdon and Z: an Approach to Formal Specification. En John E. Nicholls, editor, *Proceedings of the 5th Annual Z User Meeting, Workshops in Computing*, páginas 228–253, Oxford, 1991. Springer-Verlag.
- [Spivey 1992] J. M. Spivey. *The Z notation: a Reference Manual*. Prentice Hall, 2ª edición, 1992.
- [Thayer y Dorfman 1990] R. H. Thayer y M. Dorfman, editores. *System and Software Requirements Engineering*. IEEE Computer Society Press, 1990.

- [Thayer y Dorfman 1997] R. H. Thayer y M. Dorfman, editores. *Software Requirements Engineering*. IEEE Computer Society Press, 2ª edición, 1997. Es la 2ª edición de [Thayer y Dorfman 1990].
- [Torres 1997] J. Torres. *Especificaciones Orientadas a Objetos Basadas en Restricciones*. Tesis doctoral, Universidad de Sevilla, 1997.
- [Troyano 1998] J. A. Troyano. *Herencia y Clasificación en un Lenguaje de Especificación Orientado a Objetos*. Tesis doctoral, Universidad de Sevilla, 1998.
- [Ward y Mellor 1985] P. T. Ward y S. J. Mellor. *Structured Development for Real-Time Systems*. Yourdon Press, 1985.
- [Warmer y Kleppe 1999] J. B. Warmer y A. G. Kleppe. *The Object Constraint Language: Precise Modeling with UML*. Addison-Wesley, 1999.
- [Wieringa 1996] R. J. Wieringa. *Requirements Engineering: Frameworks for Understanding*. John Wiley & Sons, 1996.
- [Wieringa 1998] R. J. Wieringa. A Survey of Structured and Object-Oriented Software Specification Methods and Techniques. *ACM Computing Surveys*, 30(4), Diciembre 1998.
- [Yourdon Inc. 1993] Yourdon Inc. *Yourdon Systems Method: Model-Driven Systems Development*. Prentice-Hall, 1993.
- [Yourdon y Constantine 1979] E. Yourdon y B. Constantine. *Structured Design*. Prentice-Hall, 1979.
- [Yourdon 1989] E. Yourdon. *Modern Structured Analysis*. Prentice-Hall, 1989.

Capítulo 5

Validación de Requisitos

5.1 Introducción

En este capítulo se estudia la tercera actividad del modelo de procesos de ingeniería de requisitos propuesto en el capítulo 2, la *validación de requisitos*.

Dentro del entorno metodológico propuesto en esta tesis, la validación de requisitos se considera como la actividad de la ingeniería de requisitos en la que clientes y usuarios, junto con la ayuda de los ingenieros de requisitos, revisan los productos obtenidos durante las actividades anteriores para confirmar que realmente reflejan sus necesidades y que definen el producto deseado.

La validación de requisitos es otra de las actividades de la ingeniería de requisitos, que junto con la elicitación, han recibido tradicionalmente poca atención. La razón, al igual que en el caso de la elicitación, ha sido la suposición de que los requisitos eran proporcionados directamente por el cliente, con lo que, implícitamente, se asumían validados.

La ingeniería de requisitos actual asume como una necesidad la participación de los ingenieros de requisitos tanto en las actividades de elicitación, para ayudar a clientes y usuarios a encontrar y describir sus necesidades, como en las actividades de validación, para ayudar a clientes y usuarios a comprobar que los requisitos elicitados y analizados resultantes del proceso describen realmente todas sus necesidades correctamente.

La estructura del capítulo es la siguiente: en la sección 5.2 se describen los conceptos básicos de la validación de requisitos y su diferencia con la verificación. En la sección 5.3 se comentan los derechos y responsabilida-

des de los clientes y usuarios que conviene hacer explícitos al comienzo del desarrollo para conseguir su mayor participación en el proceso y facilitar la obtención del producto correcto. En la sección 5.4 se describen algunas técnicas empleadas habitualmente en la validación de requisitos. En la sección 5.5 se presenta la propuesta metodológica para la validación de requisitos y, finalmente, en la sección 5.6 se exponen las conclusiones del capítulo.

5.2 Validación y verificación de requisitos

Los términos validación y verificación (V&V) suelen usarse a veces como sinónimos o aparecer unidos en la mayor parte de la bibliografía sobre ingeniería de software e ingeniería de requisitos.

Generalmente, la V&V (sobre todo la validación) suele asociarse a los distintos tipos de técnicas de prueba, principalmente sobre el producto final [Boehm 1984, Wallace y Ippolito 1997, Piattini *et al.* 1996].

La idea de considerar la V&V como actividades prácticamente inseparables se ve reforzada por el tratamiento que se les da en varios estándares, por ejemplo los del IEEE [IEEE 1997] como el 1012 (*Standard for Software Verification and Validation Plans*), el 1059 (*Guide for Software Verification and Validation Plans*) o dentro de las actividades definidas en el 1074 [IEEE 1995a, IEEE 1995b].

En el glosario de términos del IEEE [IEEE 1990] aparecen ambos términos en una única entrada:

verificación y validación (1): *el proceso de determinar si los requisitos para un sistema o componente son completos y correctos, los productos de cada fase de desarrollo satisfacen los requisitos o condiciones impuestas por la fase previa y el sistema o componente final es acorde con los requisitos especificados.*

en la que se abordan tres aspectos fundamentales: la calidad de los requisitos, la calidad de los productos intermedios y las pruebas de aceptación del sistema. Curiosamente, en el mismo glosario aparecen también los términos V&V como entradas separadas que, sin embargo, no coinciden completamente con la definición anterior al dejar fuera los aspectos relativos a los requisitos y referirse únicamente al sistema a desarrollar o a uno de sus componentes:

validación (2): *el proceso de evaluar un sistema o componente durante o al final del proceso de desarrollo para determinar si satisface los requisitos especificados.*

verificación (2): *(a) el proceso de evaluar un sistema o componente para determinar si los productos de una fase de desarrollo dada satisfacen las condiciones impuestas al comienzo de la fase. (b) prueba formal de la corrección de un programa.*

En [ISO/IEC 1995] pueden leerse las siguientes definiciones:

validación (3): *confirmación por examen y provisión de pruebas objetivas de que los requisitos para un determinado uso específico se satisfacen.*

NOTAS: (1) En diseño y desarrollo, la validación se ocupa del proceso de examinar un producto para determinar su conformidad con las necesidades de usuario (2) La validación se suele realizar normalmente en el producto final bajo condiciones de operación definidas. Puede ser necesaria en fases más tempranas. (3) "Validado" se utiliza para designar el correspondiente estado. (4) Pueden llevarse a cabo múltiples validaciones si hay diferentes usos determinados.

verificación (3): *confirmación por examen y provisión de pruebas objetivas de que los requisitos se han satisfecho.*

NOTAS: (1) En diseño y desarrollo, la verificación se ocupa del proceso de examinar el resultado de una determinada actividad para determinar su conformidad con los requisitos establecidos para dicha actividad. (2) "Verificado" se utiliza para designar el correspondiente estado.

En [Boehm 1984], se definen ambos términos de manera informal como los procesos que llevan a contestar a las siguientes preguntas:

verificación (4): *¿se está construyendo correctamente el producto? (am I building the product right?)*

validación (4): *¿se está construyendo el producto correcto? (am I building the right product?)*

Esta definición de Boehm se reinterpreta en [Pohl 1997] de forma relativa a la ingeniería de requisitos como:

verificación (5): *la tarea de comprobar que la especificación [de requisitos] es acorde a restricciones definidas formalmente.*

validación (5): *la tarea de certificar que los requisitos especificados son consistentes con las intenciones de clientes y usuarios.*

En nuestra opinión, ambos términos tienen un significado distinto, especialmente durante la fase de ingeniería de requisitos, y no tienen porqué ir necesariamente unidos.

En este trabajo, coincidiendo básicamente con [Pohl 1997], se entenderá por validación de requisitos el conjunto de actividades encaminadas a llegar a un acuerdo entre todos los participantes en el que se ratifique que los requisitos-C elicitados y analizados representan realmente las necesidades de clientes y usuarios y que, por lo tanto, deberían llevar a la construcción de software *útil*.

Por verificación se entenderá el conjunto de actividades relacionadas con la calidad de las especificaciones de requisitos-C y D respecto a las propiedades deseables descritas en la sección 1.3.3 (pág. 16)¹.

El mecanismo más importante para la verificación de los requisitos-C que se ha contemplado en este trabajo es el *análisis de requisitos*, cuyo principal objetivo es descubrir posibles conflictos en los requisitos-C, y al que se le dedicó el capítulo 4. Otras técnicas complementarias de verificación como las inspecciones, las revisiones técnicas o las auditorías pueden consultarse, entre otros, en [IEEE 1998] y [Wallace y Ippolito 1997].

Aunque no se aborda directamente en esta tesis, existen diversas técnicas para la verificación de requisitos-D. Por ejemplo, en [Galán 2000] se plantea la verificación de modelos (requisitos-D) orientados a objetos expresados en UML [Booch *et al.* 1999] y su prototipado automático. A partir del modelo estático, compuesto por los tipos de objetos y sus operaciones en notación pre/post en OCL [Rat 1997, Warmer y Kleppe 1999], los statechart y los diagramas de colaboración, Galán describe como comprobar la corrección sintáctica de los modelos y su integración mediante la ejecución de prototipos mediante la herramienta CASE desarrollada para ello [Cañete *et al.* 2000].

¹Las propiedades deseables de los requisitos serían los requisitos de los requisitos, es decir, los *metarequisitos*, que siguiendo la definición de verificación de [ISO/IEC 1995] (nota número 1), serían los requisitos establecidos para los productos resultantes de las actividades de ingeniería de requisitos.

5.3 Derechos y responsabilidades de clientes y usuarios

Tal como se comentó en el primer capítulo, los resultados de varios estudios [TSG 1995, ESP 1996] han revelado que la principal causa del éxito de un desarrollo software es la involucración de los usuarios en el desarrollo, mientras que la principal causa de fracaso es, precisamente, la falta de comunicación por parte de los usuarios con el equipo de desarrollo.

Parece claro que para conseguir un producto que satisfaga las necesidades que justificaron su desarrollo no basta con un equipo de desarrolladores con grandes habilidades técnicas. Es necesario que los clientes y usuarios se involucren y participen activamente, especialmente durante la fase del desarrollo en la que más contacto se mantiene con ellos y que es el objeto de este estudio: la *ingeniería de requisitos*.

Una posibilidad para conseguir una mayor involucración y participación es dejar claro desde un primer momento qué es lo que esperan los clientes y usuarios de los desarrolladores y viceversa, para lo cual puede llegarse a un acuerdo o a una declaración de intenciones sobre la base de los *derechos y responsabilidades de clientes de software* descritas en [Wiegers 1999] y que se comentan a continuación.

5.3.1 Derechos de los clientes y usuarios de software

El decálogo de derechos de los clientes y usuarios de software que se expone a continuación es el presentado en [Wiegers 1999], aunque contextualizado dentro del modelo de procesos de ingeniería de requisitos que se ha presentado a lo largo de este trabajo.

- 1 **Derecho a que los ingenieros de requisitos hablen su lenguaje.** Como se comentó en el capítulo 3, los ingenieros de requisitos deben comunicarse con los clientes y usuarios en el lenguaje de éstos, evitando el vocabulario técnico. Si alguien debe aprender un nuevo vocabulario, deben ser los ingenieros de requisitos, no los clientes [Goguen y Linde 1993, Goguen 1993].
- 2 **Derecho a que los ingenieros de requisitos aprendan las características de su negocio.** Los sistemas de información son dependientes del contexto, tanto organizacional como operacional, en el que vayan a implantarse [Sawyer y Kontoya 1999, Berry y Lawrence 1998].

Por lo tanto, es necesario que los ingenieros de requisitos estudien las características del negocio del cliente, normalmente mediante la realización de un estudio del sistema actual.

3 Derecho a que los ingenieros de requisitos escriban una especificación de los requisitos del sistema software. El documento de especificación de requisitos expresa el acuerdo entre el cliente y los desarrolladores sobre las características del sistema a desarrollar. La necesidad de este documento es obvia, y de su calidad depende en gran medida el éxito del desarrollo, tal como demuestran los estudios comentados en el capítulo 1 [GAO 1979, TSG 1995, ESP 1996].

4 Derecho a que los ingenieros de requisitos expliquen el contenido de los documentos obtenidos. En general, el documento de requisitos debe estar redactado de forma que los clientes y usuarios puedan entenderlo, es decir, debe contener requisitos-C.

En el caso de que en dicho documento figuren notaciones específicas no habituales para los clientes, los ingenieros de requisitos tienen la obligación de explicar dichas notaciones para evitar una interpretación errónea de su contenido y los posteriores problemas que ello podría ocasionar. La comunicabilidad es la principal característica que debe buscarse [Wieringa 1996].

5 Derecho a que los ingenieros de requisitos le traten con respeto. Como se comentó en el capítulo 3, uno de los posibles problemas en la elicitación, y por extensión en la validación y en cualquier interacción cliente/desarrollador, es que clientes y usuarios se sientan coartados por miedo a su posible falta de conocimientos técnicos y a parecer ignorantes o incompetentes. El respeto y la atención necesaria son fundamentales para que los clientes y usuarios expresen sus necesidades sin sentirse cohibidos [Raghavan *et al.* 1994].

6 Derecho a que los ingenieros de requisitos presenten ideas y alternativas para los requisitos y la implementación. Los ingenieros de requisitos deben estudiar las características del negocio del cliente (segundo derecho) y detectar posibles problemas en los procesos de negocio, de forma que se evite desarrollar un sistema que automatice procesos que no son eficientes.

Esto debe resultar en el planteamiento de propuestas de alternativas a los procesos de negocio actuales en la organización en la que se va a implantar el sistema a desarrollar.

- 7 Derecho a que los ingenieros de requisitos describan características que hagan el producto fácil y agradable de usar.** Los ingenieros de requisitos deben participar activamente en el proceso de elicitación de requisitos y consultar a clientes y usuarios sobre características no funcionales que desean en el sistema a desarrollar.

Si el cliente manifiesta que el sistema debe ser fácil de usar, eficiente o robusto, es labor de los ingenieros de requisitos investigar cuáles son los deseos reales del cliente y definir de forma más precisa sus necesidades [Goguen 1994].

- 8 Derecho a que los ingenieros de requisitos presenten posibilidades para adaptar los requisitos y permitir reutilización.** En el caso de que los ingenieros de requisitos conozcan la existencia de componentes software de funcionalidad similar a la deseada por el cliente, es su obligación informarle de este hecho y plantear la posibilidad de adaptar los requisitos a las características de los componentes disponibles con el consiguiente ahorro de tiempo y dinero [Davis 1995, pág. 24]², [Maiden y Ncube 1998].

- 9 Derecho a que se le proporcionen estimaciones sinceras de los costes de los cambios de requisitos.** En el caso, más que probable, de que se planteen cambios en los requisitos, el cliente tiene derecho a recibir estimaciones sinceras sobre el coste de dichos cambios para poder decidir en consecuencia. Los desarrolladores deben ser honestos y no *inflar* las estimaciones del coste de aquellos cambios que no desean implementar.

- 10 Derecho a recibir un sistema que satisfaga sus necesidades funcionales y de calidad.** Conseguir el objetivo final, tener un sistema que satisfaga los requisitos planteados, sólo es posible si ha habido una comunicación fluida en ambos sentidos: clientes y usuarios han manifestado claramente sus necesidades y los desarrolladores han expuesto las posibles opciones y las restricciones técnicas.

²Principio nº 17: "*Si es posible, comprar en lugar de construir*".

5.3.2 Responsabilidades de los clientes y usuarios de software

Como contrapartida a los derechos de clientes y usuarios están sus responsabilidades, que vistas desde otro punto de vista, serían los derechos de los desarrolladores. En [Wiegers 1999] se exponen las diez siguientes:

- 1 Responsabilidad de enseñar a los ingenieros de requisitos las características de su negocio.** Esta responsabilidad es la complementaria a los derechos a que los ingenieros de requisitos hablen el lenguaje del cliente y aprendan las características de su negocio (derechos primero y segundo, respectivamente).

Es evidente que los ingenieros no pueden alcanzar dichos objetivos sin la colaboración de clientes y usuarios, y al igual que los ingenieros de requisitos no pueden asumir que los clientes y usuarios tengan conocimientos técnicos, éstos últimos no deben esperar que los ingenieros de requisitos conozcan todos los detalles de su negocio.

Según [Wiegers 1999], no se trata de que los ingenieros de requisitos se conviertan en expertos en el dominio de problemas del negocio del cliente, sino que sean capaces de entender sus problemas y plantear soluciones.

- 2 Responsabilidad de dedicar el tiempo necesario para proporcionar y clarificar los requisitos.** Para poder conseguir una comunicación satisfactoria, es necesario que los clientes y usuarios dediquen parte de su tiempo para mantener reuniones de elicitación/negociación y de validación, teniendo en cuenta que la naturaleza iterativa del proceso de ingeniería de requisitos hará que dichas reuniones se realicen varias veces.

Por su parte, los ingenieros de requisitos deben intentar minimizar este tipo de reuniones que alteran el trabajo cotidiano de clientes y usuarios, tal como se comentó en el capítulo 3.

- 3 Responsabilidad de ser específico y preciso sobre los requisitos.** La tentación de describir los requisitos de forma general y poco precisa es comprensible, ya que ir teniendo en cuenta todos los posibles detalles es un proceso largo y, normalmente, tedioso. El cliente debe ser consciente de que ceder a esta tentación implica hacer una dejación de sus responsabilidades y dejar en manos de los desarrolladores la interpretación de sus necesidades.

Puede que en un momento dado sea difícil especificar con detalle algunos aspectos, en cuyo caso puede llegarse al acuerdo de posponerlos para intentar especificarlos con ayuda de otras técnicas, por ejemplo el prototipado [Gomaa 1990, Davis 1993, Davis 1995].

- 4 Responsabilidad de tomar decisiones sin retrasos.** Durante las negociaciones necesarias para resolver los posibles conflictos en los requisitos, los ingenieros de requisitos necesitan que los clientes y usuarios tomen las decisiones necesarias. Aquellos participantes con capacidad para tomar este tipo de decisiones deben hacerlo sin una demora excesiva, teniendo en cuenta que hasta que no tomen una decisión puede que el proceso no pueda continuar o que los desarrolladores asuman la decisión que consideren más probable y tengan posteriormente que rectificar.
- 5 Responsabilidad de respetar la estimación de coste e implementabilidad de los desarrolladores.** Los desarrolladores son las personas que están en una mejor posición para realizar estimaciones de coste e implementabilidad. Puede que algunos requisitos planteados por los usuarios tengan un coste excesivo o que no puedan llevarse a cabo con la tecnología actual.
- 6 Responsabilidad de establecer prioridades en los requisitos.** Aunque en un principio todos los requisitos parecen imprescindibles, los clientes deben ser conscientes de que la mayor parte de los proyectos sobrepasan su plazo de entrega y su presupuesto [TSG 1995].

Por lo tanto, es fundamental que establezcan prioridades entre los requisitos que permitan a los desarrolladores entregar un producto que cumpla al menos aquellos requisitos considerados esenciales, independientemente de que posteriores ampliaciones del presupuesto permitan sucesivas entregas de productos que implementen la mayoría de los requisitos [Davis 1995, pág. 60]³.
- 7 Responsabilidad de revisar los documentos de requisitos y los prototipos.** La participación de clientes y usuarios en las revisiones de los documentos de requisitos es fundamental para asegurar la validación de los mismos y asegurar que se está especificando el producto correcto.

Puede que los clientes y usuarios tengan dificultades para hacerse una idea de cómo funcionará el software a partir de una especifi-

³Principio nº 50: "Priorizar requisitos".

cación de requisitos, por lo que cada vez es más frecuente el uso de prototipos [Gomaa 1990, Davis 1993], sobre todo de interfaces de usuario, como instrumentos de validación para asegurar que todos los participantes entienden los requisitos. Los clientes y usuarios deben ser conscientes de que los prototipos no constituyen el sistema final, sino una herramienta para obtener más información sobre sus necesidades.

- 8 Responsabilidad de comunicar cambios en los requisitos sin demora.** Un número elevado de cambios en los requisitos durante el desarrollo, además de ser algo muy costoso [Boehm 1984, Pressman 1997], indica que las actividades de elicitación no se han realizado apropiadamente.

Los posibles cambios en los requisitos deben comunicarse lo antes posible para evitar los elevados costes que supone su introducción cuando el desarrollo está avanzado. Aquellos clientes con capacidad para hacerlo, deben tomar la decisión de aceptar o no el cambio en función de las estimaciones de costes ofrecidas por los desarrolladores.

- 9 Responsabilidad de seguir el procedimiento de cambio de requisitos de la organización desarrolladora.** Los cambios no deben realizarse de forma incontrolada, sino seguir el procedimiento de cambio de requisitos establecido por la organización desarrolladora dentro de su proceso de gestión de requisitos.

Ello debe permitir realizar una adecuada evaluación del impacto de los cambios y una estimación del coste asociado, permitiendo así tomar las decisiones de aceptar o no los cambios con la información oportuna.

- 10 Responsabilidad de respetar los procesos de ingeniería de requisitos que realicen los desarrolladores.** El proceso de ingeniería de requisitos puede resultar frustrante para algunos clientes y usuarios si no han recibido la formación oportuna y no entienden sus ventajas.

La realización de cursos de formación en ingeniería de requisitos en los que participen clientes, usuarios y desarrolladores puede resultar en una mayor involucración de todos al tener conocimiento de porqué se realizan ciertas actividades.

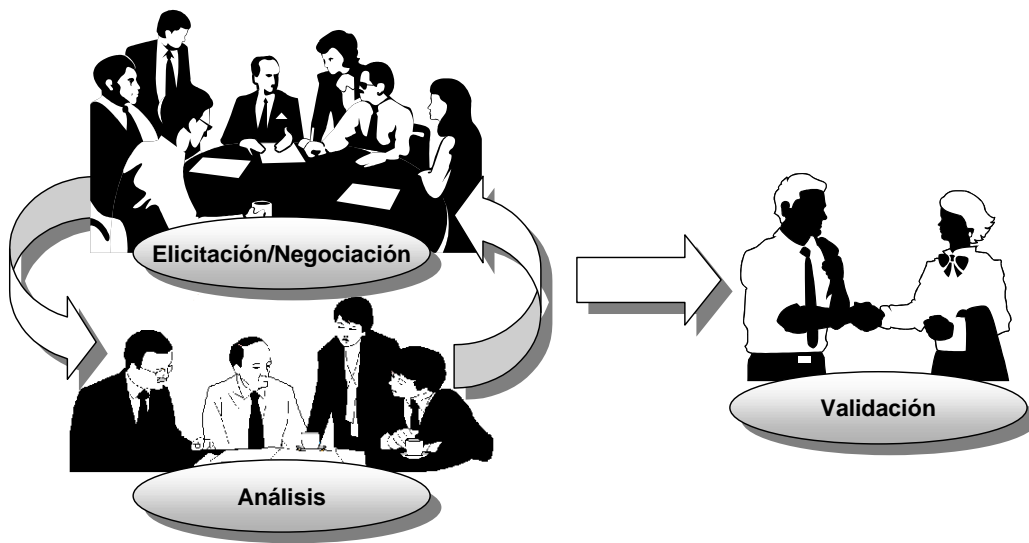


Figura 5.1: La validación como acuerdo entre clientes y desarrolladores

5.3.3 El compromiso de la validación de requisitos

La validación de los requisitos representa un compromiso entre los clientes y los desarrolladores (ver figura 5.1). En [Wieggers 1999] se exponen algunos de los problemas que puede producir un acuerdo si cada parte lo entiende según sus intereses.

Los clientes pueden considerar el acuerdo como un ritual necesario pero sin sentido, que es aceptado porque es la única forma de que comience el desarrollo. Esta actitud suele ir acompañada de cierto desinterés en la revisión de las especificaciones de requisitos y por lo tanto es probable que surjan problemas durante el desarrollo. Si el cliente no recibe el producto que esperaba, suele achacarlo a la falta de profesionalidad de los desarrolladores, en los que confió para que interpretaran correctamente sus necesidades.

Los gestores del proyecto pueden considerar el acuerdo como una forma de *congelar* los requisitos [Gause y Weinberg 1989, pág. 280], y de esta forma rechazar cualquier solicitud de cambio posterior que conllevara la necesidad de cambios en la planificación o en la asignación de recursos⁴,

⁴Conviene recordar que el modelo de ciclo de vida en el que las labores de gestión de proyectos son más sencillas es el modelo en cascada, en el que cada etapa sólo se realiza una vez. Hasta cierto punto, es entendible la actitud de querer *congelar* una especificación de requisitos para eliminar incertidumbre en el desarrollo. Desgraciadamente, los cambios son inevitables, tal como se describe en el principio nº 16 de [Davis 1995], denominado "*El cambio durante el desarrollo es inevitable*".

argumentando que el cliente debería haberlo pensado antes de validar los requisitos.

Para evitar posibles interpretaciones erróneas de lo que puede significar llegar a un acuerdo sobre un documento de requisitos, en [Wiegers 1999] se propone considerar este acuerdo como un punto de partida en el proceso de desarrollo e incluir un texto similar al siguiente en los documentos de requisitos validados:

Los abajo firmantes acordamos que este documento representa nuestro conocimiento actual de los requisitos del proyecto al día de hoy. Los futuros cambios de este documento se realizarán de acuerdo al procedimiento de cambio definido para el proyecto. Entendemos que los cambios aprobados podrán requerir que se renegocien los acuerdos actuales sobre costes, recursos asignados y fecha de entrega del proyecto.

5.4 Técnicas de validación de requisitos

Dentro de las distintas técnicas que pueden aplicarse en la validación de requisitos [Boehm 1984, Wallace y Ippolito 1997], los *walktroughs* o *recorridos* [Yourdon 1985, IEEE 1998] junto con el prototipado [Boehm *et al.* 1984, Gomaa 1990, Davis 1993] son las que, desde nuestro punto de vista, resultan más interesantes, sobre todo en el caso de aplicarse conjuntamente, siguiendo la idea de combinar distintas técnicas de revisión propuesta en [Hollock 1990].

5.4.1 Walkthroughs

El *walkthrough* [Yourdon 1985, IEEE 1998] es una técnica de revisión de productos, tradicionalmente asociada con inspecciones de código, definida en el glosario de términos de IEEE [IEEE 1990] como:

walkthrough: *técnica de análisis estático en la que un programador o diseñador dirige a miembros del equipo de desarrollo u otras personas interesadas a través de un segmento de documentación o código y los participantes realizan comentarios sobre posibles errores, violaciones de estándares de desarrollo y otros problemas.*

El walkthrough puede considerarse, junto con las inspecciones, como una forma de revisión [Hollocker 1990], que según [IEEE 1990] puede definirse como:

revisión: *un proceso o reunión durante la que producto, o conjunto de productos, se presenta a personal del proyecto, gestores, usuarios, clientes u otras partes interesadas para comentario o aprobación. Los tipos de revisión incluyen revisión de código, revisión de diseño, revisión de cualificación formal, revisión de requisitos y revisión de disponibilidad para pruebas.*

Los objetivos de una sesión de walkthrough es encontrar conflictos (defectos, omisiones, contradicciones) en el producto que se revisa, de forma que puedan plantearse alternativas y los participantes aumenten su conocimiento sobre el producto en cuestión.

Durante las sesiones de walkthrough, el autor del producto *recorre* el producto a revisar en detalle, permitiendo que los participantes pongan de manifiesto sus opiniones sobre el mismo.

La estructura, preparación y roles de los participantes de una sesión de walkthrough es similar a los de las sesiones de JAD descritas en la sección 3.4.2 (pág. 66).

5.4.1.1 Participantes del walkthrough

Según se describe en [Yourdon 1985], se pueden distinguir los siguientes tipos de participantes o roles en las sesiones de walkthrough:

- **Autor:** es el responsable de la elaboración del producto que se evalúa durante la sesión de walkthrough. Debe asegurarse de que el producto tiene el tamaño adecuado para poder ser evaluado en un máximo de dos horas y que no contiene faltas ortográficas ni defectos de forma que puedan desviar la atención durante la sesión de walkthrough. Debe también preparar la documentación complementaria para situar a los participantes en el contexto del producto que se va a revisar.

El autor debe asistir a la reunión para resolver las cuestiones y dudas que puedan surgir en el transcurso de ésta, de forma que, una vez finalizada la reunión, incorpore al producto los cambios sugeridos en la reunión y lo reelabore en el caso de que haya que someterlo a una nueva sesión.

- **Presentador:** es el responsable de organizar y dirigir la sesión, de seleccionar a los participantes, entre los que debe encontrarse el autor del producto a revisar, y de acordar con ellos la fecha y el lugar de la reunión.

También debe elegir al moderador y al registrador, entregar la documentación a los revisores, dar una visión general del producto, resumir el estado de salida de la última sesión de walkthrough si la hubo, presentar el producto y tomar la decisión final si no hay consenso sobre el estado del producto. Además elaborará la documentación resultante de la reunión.

Es relativamente frecuente que el autor y el presentador sean la misma persona.

- **Revisores:** son los principales participantes de la sesión. Antes de la reunión deben revisar cuidadosamente la información que han recibido y anotar los comentarios que desean exponer durante la reunión.

En el transcurso de ésta serán participantes activos explicando dichos comentarios y realizando al autor las preguntas que consideren necesarias para entender el producto y poder calificarlo. Son los responsables de decidir si el producto se acepta, si es necesario corregirlo y volver a convocar una reunión o si es suficiente con revisarlo.

En el caso de los walkthrough sobre especificaciones de requisitos, los revisores serán principalmente los clientes y usuarios del producto final.

- **Moderador:** antes de la reunión, debe aclarar los roles y responsabilidades de los asistentes con el presentador. También es responsable de reconducir la reunión si se desvía de los puntos principales y de que se realice de forma ordenada. Una vez terminada la sesión, debe encargarse de consultar a los participantes su opinión sobre el resultado de la reunión.
- **Registrador:** es el responsable de registrar la hora de comienzo de la sesión, los conflictos expresados por los revisores durante la sesión, las acciones sugeridas por éstos y la hora de finalización de la sesión.

5.4.1.2 Fases del walkthrough

Dentro de la técnica de walkthrough se distinguen tres fases principales [Yourdon 1985, IEEE 1998]:

- 1 **Preparación:** para que la sesión se desarrolle satisfactoriamente, los participantes deben recibir con anterioridad a la celebración de la reunión el material correspondiente al producto a revisar para que puedan estudiarlo y preparar las preguntas o los temas a discutir que consideren oportunos.
- 2 **Revisión:** durante la celebración de la sesión de revisión, el presentador debe *recorrer* (*walk through*) con detalle el producto, de forma que los participantes puedan plantear las cuestiones que consideren oportunas.

El registrador tomará nota de éstas elaborando una lista de conflictos (defectos, omisiones, etc.). El moderador debe intentar que los participantes no busquen soluciones a los conflictos planteados sino que se limiten a exponerlos. Como resultado de la reunión, el producto puede aceptarse, recomendarse ciertos cambios o acordarse la celebración de una nueva sesión si la naturaleza de los cambios necesarios lo requiriese.
- 3 **Conclusión:** una vez terminada la sesión es necesario transformar la documentación generada en documentos formales que deben recoger la información referente a la sesión: nombre del producto revisado, fecha de la sesión, participantes, participantes implicados en la siguiente sesión de walkthrough si procede y la lista enumerada de aportaciones realizadas por los participantes así como las acciones correctivas que se hayan propuesto.

En el ámbito de la ingeniería de requisitos, el walkthrough puede aplicarse a las especificaciones de requisitos-C, de forma que los autores de dichas especificaciones revisan públicamente los requisitos-C en presencia de los clientes y usuarios con el objetivo de que éstos comprendan el significado de cada requisito y puedan manifestar su acuerdo o desacuerdo con los mismos. En el caso de utilizar casos de uso para especificar requisitos-C funcionales, la técnica es una forma *natural* de validar la secuencia de pasos de un caso de uso al *recorrerlos* en público [Fairley y Thayer 1997].

Como resultado de la realización de las sesiones de walkthrough pueden detectarse conflictos, descubrirse nuevos requisitos o validar los re-

quisitos ya existentes. En el caso de que se hayan desarrollado prototipos, se puede aumentar la efectividad de los walkthroughs relativos a los requisitos-C de almacenamiento de información y funcionales realizando los recorridos a la vez que se muestra el prototipo.

Otra posible aplicación de la técnica de walkthrough en ingeniería de requisitos es su uso, dentro del grupo de desarrolladores, para *recorrer* los modelos generados durante el análisis de los requisitos-C y encontrar posibles conflictos o contradicciones.

5.4.2 Prototipos orientados a clientes/usuarios

La utilización de prototipos como técnica tanto de validación como de elicitación está cada vez más extendida dentro de la práctica de la ingeniería de requisitos [Boehm *et al.* 1984, Gomaa 1990, Davis 1993].

Esta técnica permite que los usuarios tengan una idea más clara del producto que van a recibir, ya que su grado de comunicabilidad es más alto que el de las especificaciones textuales de requisitos.

Dentro de las posibles clases de prototipos, la más interesante para las labores de elicitación y validación son los *prototipos de interfaz de usuario*, independientemente de que su naturaleza sea *evolutiva* o de *usar y tirar* [Davis 1993, cap. 6], [Davis 1995, págs. 18–20]⁵, o de que sean prototipos en papel o ejecutables.

Utilizando una nomenclatura similar a la que se ha venido usando a lo largo de este trabajo respecto a los requisitos, los prototipos de interfaz de usuario podrían denominarse como *prototipos orientados a clientes/usuarios* o, abreviadamente, *prototipos-C*.

Aquellos prototipos centrados en los aspectos funcionales que se suelen generar a partir de especificaciones formales y que no suelen incluir aspectos de interfaz de usuario serían los *prototipos orientados a los desarrolladores* o, abreviadamente, *prototipos-D*, cuyo uso principal sería la comprobación de propiedades de los modelos desarrollados durante el análisis de los requisitos, tal como se comentó en el capítulo anterior.

⁵Principios nº 11 "*Construir la clase correcta de prototipo*", nº 12 "*Construir las características correctas en un prototipo*" y nº 13 "*Construir prototipos rápidamente*".

5.5 Propuesta metodológica para la validación de requisitos

La propuesta metodológica para la validación de requisitos que se propone en este trabajo, y que se incluye completa en el apéndice D, propone una serie de tareas a realizar y productos a obtener durante la realización de la actividad de validación de requisitos del modelo de procesos descrito en el capítulo 2.

En la propuesta se contemplan las tres tareas que pueden verse en la figura 5.2, en la que se propone un posible orden de realización orientativo en el que las dos tareas en las que se validan los requisitos—C se realizarían en paralelo. En las siguientes secciones se describen cada una de ellas.

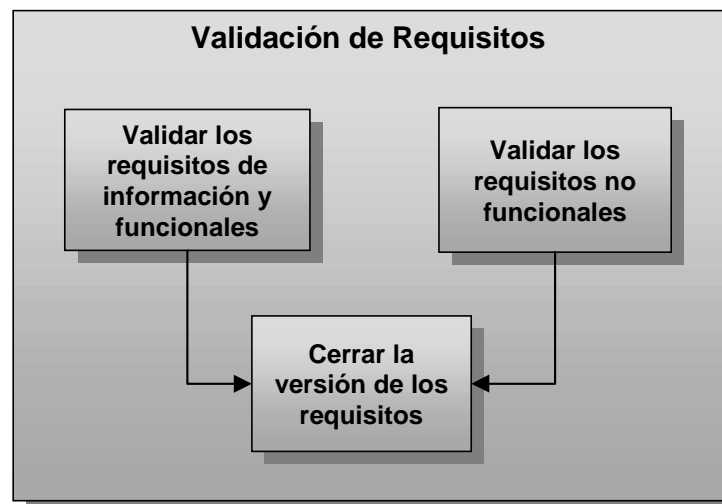


Figura 5.2: Tareas de validación de requisitos

5.5.1 Tarea 1: Validar los requisitos de almacenamiento de información y funcionales

El objetivo de esta tarea es validar los requisitos de almacenamiento de información y funcionales, es decir, asegurarse de que representan realmente las necesidades de clientes y usuarios.

Se han agrupado los dos tipos de requisitos en una única tarea porque, en nuestra opinión, su validación mediante walkthroughs asistidos con

prototipos de interfaz de usuario permite una validación conjunta de forma sencilla guiada por los requisitos funcionales descritos como casos de uso, teniendo siempre en cuenta las relaciones de rastreabilidad establecidas entre los requisitos de almacenamiento de información y los requisitos funcionales.

Los productos resultantes de esta tarea serían los requisitos de almacenamiento de información, los requisitos funcionales y el prototipo validados total o parcialmente. En el caso de que se descubran conflictos durante la validación, éstos serían también productos de esta tarea y deberían resolverse en nuevas sesiones de elicitación/negociación (ver capítulo 3).

Como ya se ha comentado, las técnicas recomendadas para la realización de esta tarea son una combinación de walkthrough de los casos de uso y de utilización de prototipos de interfaz de usuario para ayudar a los clientes y usuarios a revisar la documentación de requisitos y, en el caso de que aparezcan conflictos, las plantillas para conflictos descritas en la sección 3.6.6 (pág. 95).

5.5.2 Tarea 2: Validar los requisitos no funcionales

El objetivo de esta tarea es validar los requisitos no funcionales. Sus productos, de forma similar a la tarea anterior, serían los requisitos no funcionales validados total o parcialmente, y los posibles conflictos que pudieran aparecer.

A diferencia de la tarea anterior, la única técnica que parece aplicable para realizar esta tarea es una revisión por parte de los clientes y usuarios, ayudados por los ingenieros de requisitos para aclarar las posibles dudas que surjan durante la revisión.

Al igual que en las actividades de análisis, en las que los requisitos no funcionales no suelen contemplarse en las técnicas de modelado, en la validación tampoco parece existir una forma de integrarlos en un proceso de walkthrough o de prototipado.

5.5.3 Tarea 3: Cerrar la versión de los requisitos

Si no han aparecido nuevos conflictos durante el proceso de validación, se debe llegar a un acuerdo entre clientes y desarrolladores para cerrar la versión actual de los requisitos, siempre teniendo en cuenta que representa el

conocimiento actual de los mismos y que, probablemente, sufrirá cambios en el futuro [Wiegers 1999].

El producto de esta actividad es, por lo tanto, una versión cerrada de los requisitos y del prototipo.

5.6 Conclusiones

En este capítulo se han estudiado algunos aspectos relacionados con la validación de requisitos. Se han comentado los decálogos de [Wiegers 1999], como forma de llegar a acuerdos previos sobre los derechos y responsabilidades de los participantes en el proceso de ingeniería de requisitos, y su declaración de acuerdo como una forma consensuada de cerrar versiones de los requisitos.

Dentro de las técnicas de validación se ha prestado especial atención al *walkthrough* y a los prototipos, especialmente a los de interfaz de usuario, que desde nuestro punto de vista son los más interesantes en la fase de ingeniería de requisitos y especialmente durante las actividades de elicitación y validación, en las que la comunicación con los clientes es lo más importante.

También se ha presentado la propuesta metodológica para la validación de requisitos, en la que se ha optado por recomendar una combinación de *walkthroughs*, casos de uso y prototipado como la técnica que más se adapta a las necesidades de la validación. Al igual que en análisis, los requisitos no funcionales siguen necesitando de un mayor estudio que los integre en los procesos de validación mediante algún tipo de prototipado o alguna otra técnica.

5.7 Bibliografía

[Berry y Lawrence 1998] D. M. Berry y B. Lawrence. Requirements Engineering. *IEEE Software*, 15(2):26–29, Marzo/Abril 1998.

[Boehm *et al.* 1984] B. W. Boehm, T. E. Gray, y T. Seewalt. Prototyping versus Specifying: A Multiproject Experiment. *IEEE Transactions on Software Engineering*, 10(3):290–303, Mayo 1984. También aparece en [Thayer y Dorfman 1990].

- [Boehm 1984] B. W. Boehm. Verifying and Validating Software Requirements and Design Specifications. *IEEE Software*, 1(1):75–88, Enero 1984. También aparece en [Thayer y Dorfman 1990].
- [Booch *et al.* 1999] G. Booch, J. Rumbaugh, y I. Jacobson. *The Unified Modeling Language User Guide*. Addison–Wesley, 1999.
- [Cañete *et al.* 2000] J. M. Cañete, F. J. Galán, y M. Toro. An Application Framework for the Execution of UML/OCL Models. En *Actas de las V Jornadas de Trabajo Menhir*, Granada, 2000.
- [Christel y Kang 1992] M. G. Christel y K. C. Kang. Issues in Requirements Elicitation. Technical Report CMU/SEI-92-TR-12, Software Engineering Institute, Carnegie Mellon University, 1992. Disponible en <http://www.sei.cmu.edu>.
- [Davis 1993] A. M. Davis. *Software Requirements: Objects, Functions and States*. Prentice–Hall, 2ª edición, 1993.
- [Davis 1995] A. M. Davis. *201 Principles of Software Development*. McGraw–Hill, 1995.
- [ESP 1996] ESPITI – European User Survey Results. Informe Técnico ESI-1996-TR95104, European Software Institute, 1996. Disponible en <http://www.esi.es>.
- [Fairley y Thayer 1997] R. E. Fairley y R. H. Thayer. The Concept of Operations: The Bridge from Operational Requirements to Technical Specifications. En Thayer y Dorfman [1997], páginas 73–83.
- [Galán 2000] F. J. Galán. *Formalizaciones para Sistemas Software Orientados a Objetos*. Tesis doctoral, Universidad de Sevilla, 2000. Pendiente de presentación.
- [GAO 1979] Contracting for Computer Software Development: Serious Problems Require Management Attention to Avoid Wasting Additional Millions. Report FGMSD-80-4, U. S. Government Account Office, Noviembre 1979. Este documento es difícil de encontrar, pero está comentado y referenciado, entre otros, en [Davis 1993], [Christel y Kang 1992] y [Goguen 1994].
- [Gause y Weinberg 1989] D. C. Gause y G. M. Weinberg. *Exploring Requirements: Quality Before Design*. Dorset House, 1989.

- [Goguen y Linde 1993] J. A. Goguen y C. Linde. Techniques for Requirements Elicitation. En *Proceedings of the First International Symposium on Requirements Engineering*, 1993. También aparece en [Thayer y Dorfman 1997]. Disponible en <http://www.cse.ucsd.edu/~goguen>.
- [Goguen 1993] J. A. Goguen. Social Issues in Requirements Engineering. En *Proceedings of the First International Symposium on Requirements Engineering*, 1993. Disponible en <http://www.cse.ucsd.edu/~goguen>.
- [Goguen 1994] J. A. Goguen. Requirements Engineering as the Reconciliation of Social and Technical Issues. En *Requirements Engineering: Social and Technical Issues*, páginas 165–199. Academic Press, 1994. Disponible en <http://www.cse.ucsd.edu/~goguen>.
- [Gomaa 1990] H. Gomaa. The Impact of Prototyping on Software System Engineering. En Thayer y Dorfman [1990], páginas 543–552. Aparece también en [Thayer y Dorfman 1997].
- [Hollocker 1990] C. P. Hollocker. A Review Process Mix. En Thayer y Dorfman [1990], páginas 485–491.
- [IEEE 1990] IEEE. IEEE Standard Glossary of Software Engineering Terminology. IEEE Standard 610.12–1990, Institute of Electrical and Electronics Engineers, 1990.
- [IEEE 1995a] IEEE. IEEE Guide for Developing Software Life Cycle Processes. IEEE/ANSI Standard 1074.1–1995, Institute of Electrical and Electronics Engineers, 1995.
- [IEEE 1995b] IEEE. IEEE Standard for Developing Software Life Cycle Processes. IEEE/ANSI Standard 1074–1995, Institute of Electrical and Electronics Engineers, 1995.
- [IEEE 1997] IEEE. *IEEE Software Engineering Standards Collection*. Institute of Electrical and Electronics Engineers, 1997.
- [IEEE 1998] IEEE. IEEE Guide for Software Reviews and Audits. IEEE/ANSI Standard 1028–1988, Institute of Electrical and Electronics Engineers, 1998.
- [ISO/IEC 1995] ISO/IEC. Information Technology–Software Life Cycle Processes. International Standard 12207 : 1995, International Organization for Standardization, 1995.

- [Maiden y Ncube 1998] N. A. Maiden y C. Ncube. Acquiring COTS Software Selection Requirements. *IEEE Software*, 15(2), Marzo/Abril 1998.
- [Piattini *et al.* 1996] M. G. Piattini, J. A. Calvo-Manzano, J. Cervera, y L. Fernández. *Análisis y Diseño Detallado de Aplicaciones Informáticas de Gestión*. ra-ma, 1996.
- [Pohl 1997] K. Pohl. Requirements Engineering: An Overview. *Encyclopedia of Computer Science and Technology*, 36, 1997. Disponible en <http://sunsite.informatik.rwth-aachen.de/CREWS/reports96.htm>.
- [Pressman 1997] R. S. Pressman. *Ingeniería del Software. Un Enfoque Práctico*. McGraw-Hill, 4ª edición, 1997.
- [Raghavan *et al.* 1994] S. Raghavan, G. Zelesnik, y G. Ford. Lecture Notes on Requirements Elicitation. Educational Materials CMU/SEI-94-EM-10, Software Engineering Institute, Carnegie Mellon University, 1994. Disponible en <http://www.sei.cmu.edu>.
- [Rat 1997] Rational Software Corporation. *Object Constraint Language Specification*, 1.1 edición, Septiembre 1997. Disponible en <http://www.rational.com>.
- [Sawyer y Kontoya 1999] P. Sawyer y G. Kontoya. SWEBOK: Software Requirements Engineering Knowledge Area Description. Informe Técnico Versión 0.5, SWEBOK Project, 1999. Disponible en <http://www.swebok.org>.
- [Thayer y Dorfman 1990] R. H. Thayer y M. Dorfman, editores. *System and Software Requirements Engineering*. IEEE Computer Society Press, 1990.
- [Thayer y Dorfman 1997] R. H. Thayer y M. Dorfman, editores. *Software Requirements Engineering*. IEEE Computer Society Press, 2ª edición, 1997. Es la 2ª edición de [Thayer y Dorfman 1990].
- [TSG 1995] TSG. *The CHAOS Report*. The Standish Group, 1995. Disponible en <http://www.standishgroup.com/chaos.html>.
- [Wallace y Ippolito 1997] D. Wallace y L. Ippolito. Verifying and Validating Software Requirements Specifications. En Thayer y Dorfman [1997], páginas 389–404.

[Warmer y Kleppe 1999] J. B. Warmer y A. G. Kleppe. *The Object Constraint Language: Precise Modeling with UML*. Addison–Wesley, 1999.

[Wiegers 1999] K. Wiegers. Customer Rights and Responsibilities. *Software Development*, Diciembre 1999. Disponible en <http://www.sdmagazine.com/breakrm/features/s9912f2.shtml>.

[Wieringa 1996] R. J. Wieringa. *Requirements Engineering: Frameworks for Understanding*. John Wiley & Sons, 1996.

[Yourdon 1985] E. Yourdon. *Structured Walkthroughs*. Prentice–Hall, 3ª edición, 1985.

Capítulo 6

Conclusiones y trabajo futuro

6.1 Conclusiones

En esta tesis se ha abordado de forma general la problemática relacionada con la ingeniería de requisitos, y más específicamente con la ingeniería de requisitos de sistemas de información. Como resultado de la investigación realizada y de la experiencia obtenida, se han obtenido las conclusiones que se exponen a continuación.

6.1.1 La distancia entre investigación y práctica

La distancia entre la investigación académica sobre ingeniería de requisitos y los problemas reales que tienen los desarrolladores profesionales es preocupantemente grande y, lo que es peor, parece no disminuir [Sheptunov 1999].

La visión que tienen muchos desarrolladores es que todo lo relacionado con los requisitos no es más que la generación de documentación que nadie lee, exigida por los estándares que han de seguir o por los procesos de calidad implantados en su organización. En [Kovitz 1998, pág. 269] puede leerse:

"Muchos programadores están convencidos de que escribir documentos de requisitos es una pérdida de tiempo y que, por muchos defectos que pueda tener ir directamente al diseño después de algunas charlas informales con el cliente, por lo menos esta estrategia evita el tiempo y el coste de escribir un enorme documento que no sirve para nada."

Afortunadamente, la existencia de esta gran distancia es algo que empieza a tenerse en cuenta en la comunidad académica. Por ejemplo, estableciendo como criterios de aceptación de artículos en congresos sobre ingeniería de requisitos que se aborden propuestas contrastadas con al menos un proyecto real que contribuyan a reducir la distancia entre investigación y práctica [Berry y Lawrence 1998, Cheng y Weiss 2000].

Las ideas expuestas en esta tesis tienen como objetivo reducir dicha distancia. Han evolucionado enormemente a través de su aplicación en numerosas prácticas académicas realizadas por alumnos de las universidades de Sevilla, Salamanca y Valladolid y están siendo usadas en tres proyectos reales de Sadiel S.A. de forma satisfactoria.

De su aplicación en proyectos reales no disponemos de datos cuantitativos que apoyen la información cualitativa obtenida de los responsables de los proyectos, por ejemplo comparaciones de costes entre proyectos que usen o no las ideas presentadas.

Esto nos lleva a otra conclusión: la dificultad de la experimentación en ingeniería del software y, específicamente, en ingeniería de requisitos.

6.1.2 La dificultad de la experimentación

Lo ideal sería poder aplicar las propuestas académicas en varios proyectos reales, observar los resultados y poder compararlos con otros proyectos en los que no se hubieran aplicado. De esta forma, los resultados empíricos demostrarían la validez o no del trabajo de investigación.

Normalmente, la reticencia de la mayoría de las empresas hacia las ideas académicas, a hacer pública información que consideran confidencial y la duración habitual de los proyectos reales, hace que planteamientos como éste sean muy escasos en la comunidad investigadora.

En el caso de que se realicen experimentos, éstos suelen tener como sujetos a alumnos universitarios en prácticas o proyectos fin de carrera, con muy pocas características similares a los de los proyectos reales, por ejemplo el experimento descrito en [Boehm *et al.* 1984].

6.1.3 La formación en ingeniería de requisitos

La educación que se imparte actualmente en las universidades sobre los distintos aspectos de la ingeniería de requisitos tiene poco que ver con las necesidades reales de las empresas que desarrollan software [Davis 1998].

En una encuesta realizada como parte del ISRE'95 [Macaulay y Mylopoulos 1996] en la que se preguntaba a los responsables de desarrollo de varias empresas europeas y norteamericanas sobre las cualidades que desearían encontrar en los ingenieros de requisitos, las respuestas consideraban más importantes las habilidades de comunicación, asimilación y expresión de información, de trabajar en entornos en los que la ambigüedad y el cambio están presentes, tener una mente abierta e incluso tener sentido del humor, por encima de un conocimiento puramente tecnológico.

Respecto a la formación deseada, aparte de algún título relacionado con ingeniería de software, se citaba la posibilidad de tener formación sobre aspectos empresariales o en psicología.

En otras palabras, parece que los aspectos de *inteligencia emocional*, especialmente la *empatía* [Goleman 1996, Goleman 1999], son tan importantes o más en el ingeniero de requisitos que los puramente técnicos, o al menos así piensan los responsables de la contratación de este tipo de profesionales.

Si se comparan estos resultados con los ofrecidos en el mismo informe citado anteriormente sobre la educación en ingeniería de requisitos en varias universidades europeas y norteamericanas, es evidente que la distancia entre investigación y práctica se mantiene también entre formación y práctica.

Las ideas presentadas en esta memoria se han utilizado principalmente con alumnos, teniendo como objetivo acercar su formación lo más posible a la problemática real. Sin embargo, hay que reconocer que la imposibilidad de contar con clientes y usuarios reales hace muy difícil conseguir este objetivo.

6.1.4 Madurez del proceso de ingeniería de requisitos

Comparado las características del entorno metodológico presentado en esta tesis, incluyendo el prototipo de herramienta CASE descrito en el apéndice E, con las guías para incrementar la madurez del proceso de ingeniería de requisitos del proyecto europeo REAIMS [Sommerville y Sawyer 1997, Sawyer *et al.* 1997], los resultados, que pueden verse detalladamente en las figuras 6.1 y 6.2 son los siguientes:

- **General:** de las 57 guías propuestas, las metodologías presentadas siguen 41, un 72%
- **Guías básicas:** de las 10 guías básicas, se siguen 7, un 70%

- **Guías del nivel Inicial:** de las 34 guías propuestas, se siguen 27, un 79%
- **Guías del nivel Repetible:** de las 18 guías propuestas, se siguen 11, un 61%
- **Guías del nivel Definido:** de las 5 guías propuestas, se siguen 3, un 60%

Guías de nivel Inicial (en negrita las 10 guías básicas)	
Definir una estructura estándar de documento	✓
Explicar cómo usar el documento	✓
Incluir un resumen de los requisitos	✓
Incluir un estudio de la necesidades	✓
Definir términos especializados	✓
Editar el documento para su legibilidad	✓
Ayudar a los lectores a encontrar información	✓
Hacer el documento fácil de cambiar	✓
Valorar la implementabilidad del sistema	
Tener en cuenta las consideraciones organizacionales y políticas	✓
Identificar y consultar a los participantes	✓
Registrar las fuentes de los requisitos	✓
Definir el entorno operacional del sistema	✓
Dirigir la elicitación en función de las problemas del negocio	✓
Definir los límites del sistema	✓
Usar listas de comprobación para el análisis de requisitos	
Proporcionar software para dar soporte a las negociaciones	
Planificar los conflictos y su resolución	✓
Priorizar requisitos	✓
Definir plantillas estándar para describir requisitos	✓
Usar lenguaje sencillo y conciso	✓
Usar diagramas apropiadamente	✓
Suplementar el lenguaje natural con otras descripciones de requisitos	✓
Desarrollar modelos complementarios del sistema	✓
Modelar el entorno del sistema	
Modelar la arquitectura del sistema	
Comprobar que los documentos de requisitos cumplen los estándares	✓
Organizar inspecciones formales de los requisitos	✓
Usar equipos multidisciplinares para revisar los requisitos	✓
Definir listas de comprobación de validación	
Identificar de forma única cada requisito	✓
Definir políticas para la gestión de requisitos	
Definir políticas de rastreabilidad	✓
Mantener un manual de rastreabilidad	✓

Figura 6.1: Guías REAIMS (nivel Inicial)

Guías de nivel Repetible	
Buscar restricciones del dominio	✓
Registrar el porqué de los requisitos	
Obtener requisitos desde distintos puntos de vista	✓
Prototipar los requisitos con dificultades de comprensión	✓
Usar escenarios para elicitación de requisitos	✓
Definir los procesos operacionales actuales	✓
Clasificar los requisitos con un enfoque multidimensional	
Usar matrices de interacción para encontrar conflictos y solapamientos	
Especificar los requisitos de forma cuantitativa	✓
Usar métodos organizados para el modelado del sistema	✓
Usar un diccionario de datos	✓
Documentar los enlaces entre los requisitos y los modelos del sistema	✓
Usar prototipos para animar los requisitos	✓
Escribir un borrador del manual de usuario	
Proponer casos de prueba para los requisitos	
Usar una base de datos para gestionar requisitos	✓
Definir políticas para gestionar los cambios	
Identificar requisitos globales del sistema	

Guías de nivel Definido	
Reutilizar requisitos	✓
Valorar los riesgos de los requisitos	
Describir los modelos del sistema en lenguaje natural	✓
Identificar requisitos volátiles	✓
Registrar los requisitos rechazados	

Figura 6.2: Guías REAIMS (niveles Repetible y Avanzado)

Aplicando el baremo descrito en [Sommerville y Sawyer 1997], si una organización aplicara el entorno metodológico descrito en esta tesis, sería clasificada dentro del nivel 2, repetible, de madurez de proceso de ingeniería de requisitos.

En general, las razones por la que no se han seguido algunas guías se deben a que en esta tesis no se han abordado directamente los problemas relacionados con la gestión de requisitos, aunque se ha tenido en cuenta el principal mecanismo de la gestión de requisitos que es la rastreabilidad.

Tampoco se han abordado los problemas de aseguramiento de calidad, entre ellos la verificación, entendiendo como tal la comprobación de que los requisitos, tanto a nivel de requisitos-C como requisitos-D, son acordes a los estándares que deben seguir, o dicho de otro modo, que son *sin-tácticamente* correctos.

Tanto la gestión de requisitos como su calidad se incluyen entre las líneas de trabajo futuro que se indican a continuación.

6.2 Trabajo futuro

Dentro de las posibles líneas de continuación del trabajo presentado en esta tesis, pueden destacarse las siguientes, la mayor parte de las cuales son necesidades que se han puesto de manifiesto al aplicar las ideas presentadas, o bien son tareas no abordadas por la necesidad de imponer un límite al trabajo de investigación, ya que de no ser así se prolongaría indefinidamente en el tiempo.

6.2.1 Completar el entorno metodológico

Como ya se ha comentado al comprobar cuántas guías de madurez RE-AIMS cumplía, el entorno metodológico presentado no abarca todas necesidades dentro de la fase de ingeniería de requisitos, por lo que es necesario completarlo con aspectos de:

- **Gestión de requisitos:** es necesario introducir la gestión de requisitos como una actividad ortogonal a las descritas en los capítulos 3, 4 y 5 como parte del modelo de procesos propuesto en el capítulo 2.

Las bases para hacerlo están presentes, ya que se han tenido muy en cuenta los aspectos de rastreabilidad entre los distintos productos del proceso. Sin embargo, falta por establecer una metodología de gestión de requisitos que aborde la gestión de configuración, las políticas y procedimientos de cambio y el análisis de su impacto.

- **Verificación de requisitos:** como se comentó en el capítulo 5, normalmente se suelen tratar de forma conjunta las actividades de verificación y validación.

En la propuesta realizada en este trabajo se han querido diferenciar explícitamente ambas actividades: la validación como proceso que intenta asegurar que los requisitos corresponden realmente a las necesidades de clientes y usuarios y la verificación como una más de las actividades de aseguramiento de calidad que deberían ser ortogonales a las propias del desarrollo.

Por lo tanto, la metodología propuesta debería completarse con los aspectos de aseguramiento de calidad, entre ellos la verificación, teniendo en cuenta los distintos estándares aplicables (ISO 9000, SPI-CE, Plan General de Garantía de Calidad, etc.).

- **Desarrollo de herramientas CASE de soporte:** el éxito de una propuesta metodológica depende en gran medida de la existencia de herramientas de CASE que la soporten. En el apéndice E se describe un prototipo de herramienta CASE para el entorno metodológico descrito en esta tesis. En un futuro, dicho prototipo deberá ir incorporando nuevas funcionalidades que se detecten como necesarias con la experiencia de su utilización.

Dentro de las posibles ventajas del uso del prototipo descrito en el apéndice E está la posibilidad de disponer de un número considerable de proyectos de ingeniería de requisitos en un formato normalizado y abierto, al almacenarse en bases de datos de Microsoft Access. La utilización de este prototipo en las universidades de Sevilla, Salamanca y Valladolid puede permitir disponer a corto plazo de un número importante de proyectos sobre los que investigar distintas propiedades como descubrir nuevos patrones-R, desarrollar métricas, etc.

6.2.2 Reutilización de requisitos

Una de las características de un proceso de desarrollo maduro es la reutilización [Sommerville y Sawyer 1997, Paulk *et al.* 1993]. En los capítulos 3 y 4 se apuntaron algunas posibilidades para reutilizar requisitos, que sin embargo no son más que unas primeras aproximaciones al problema.

En un futuro, la reutilización debería contemplarse explícitamente en el modelo de procesos, así como tener un soporte automatizado como los repositorios descritos en [García 2000].

Sin embargo, los principales motivos de aplicar técnicas de reutilización en ingeniería de requisitos, al menos desde nuestro punto de vista, no es simplemente reducir costes e incrementar la calidad, que evidentemente son fundamentales. En nuestra opinión, la introducción sistemática de la reutilización en ingeniería de requisitos puede llevar a un cambio radical en los planteamientos habituales sobre el proceso a seguir, conduciendo hacia una *ingeniería de requisitos basada en componentes*, de forma similar a como ya se está produciendo a nivel de implementación.

Idealmente, podría llegarse a situaciones en las que, si el ingeniero de requisitos dispone de un repositorio lo suficientemente rico, el proceso *elicitación-análisis-validación* se vea sustituido por otro en el que, en función de las necesidades del cliente, se obtuvieran los requisitos seleccio-

nándolos de una serie de requisitos estandarizados, los cuales, mediante relaciones de rastreabilidad, indicarían qué componentes software deberían ensamblarse para implementar un sistema que diera al cliente la solución a sus problemas.

Esta situación sería similar a la que se produce cuando se adquiere un producto modular prefabricado: el cliente escoge de un catálogo en función de sus necesidades, conociendo desde el primer momento el coste de su elección.

6.2.3 Sacar más rendimiento a los requisitos

La introducción de procesos de ingeniería de requisitos en organizaciones desarrolladoras de software no es una tarea sencilla, sobre todo teniendo en cuenta la opinión desfavorable de la mayoría de los programadores, tal como se comentó en la sección 6.1.1.

Una de las formas de captar la atención de los más escépticos es demostrando que los resultados del proceso de ingeniería de requisitos no consisten sólo en documentos de requisitos.

Así, una posible línea de trabajo futuro debería consistir en sacar el mayor provecho a la información obtenida durante el proceso. Por ejemplo:

- **Generación de casos de prueba:** a partir de los casos de uso utilizados para describir los requisitos funcionales, debería ser posible generar automáticamente un borrador de las pruebas de aceptación del sistema, ya que describen la conducta externa del sistema desde el punto de vista de los usuarios.

A partir de las especificaciones de las operaciones del sistema mediante pre y postcondiciones en OCL, debería ser también posible generar pruebas de caja negra que comprobaran la conducta del sistema como un *macro-componente* que ofrece una serie de servicios a su entorno, complementando de esta manera el trabajo iniciado en [Corchuelo 1999] y [Galán 2000].

- **Generación de manuales de usuario:** utilizando también los casos de uso, podría generarse automáticamente la información para los manuales de usuario.

Esta información podría organizarse de forma que, para cada caso de uso que describe como un tipo de usuario utiliza el sistema para

conseguir un determinado objetivo, hubiera una sección en el manual de usuario con un título similar a "*¿Cómo hacer X?*". En esta sección se describirían, paso a paso, las instrucciones necesarias de forma similar a como están descritos los casos de uso.

- **Generación de código:** la evolución de los lenguajes de programación ha provocado que la distancia semántica entre éstos y los modelos conceptuales disminuya considerablemente. Si a esto unimos las heurísticas conocidas para transformar ciertos modelos en código equivalente, es normal que uno de los deseos de los desarrolladores sea que el código que deben escribir manualmente se genere a partir de modelos de más alto nivel.

Los modelos conceptuales que se realizan durante el análisis, es decir los requisitos-D, normalmente no contienen todos los detalles necesarios para poder generar completamente el código de la aplicación final, aunque sí podría generarse parte de él o, al menos, prototipos funcionales como los descritos en [Corchuelo 1999] y [Galán 2000].

Aplicar las heurísticas conocidas y descubrir otras nuevas para generar código a partir de los requisitos-D es, sin duda, una de las bazas que puede hacer que la ingeniería de requisitos cambie su imagen en los entornos no académicos.

6.2.4 Requisitos no funcionales y arquitectura de software

Los requisitos no funcionales han sido y siguen siendo, probablemente por su heterogeneidad, los menos estudiados en ingeniería de requisitos. Tradicionalmente se ha asumido que este tipo de propiedades del sistema eran tratados durante el diseño del software, sobre todo durante el diseño de la arquitectura.

El decidir la arquitectura del software es una tarea que forma parte de las actividades de diseño, y por lo tanto, parece que queda fuera del ámbito de la ingeniería de requisitos. Sin embargo, existen ya opiniones que apuntan a que la influencia de la arquitectura sobre los requisitos debe tenerse en cuenta durante su obtención [Berry y Lawrence 1998].

Uno de los trabajos futuros es el tratamiento apropiado de los requisitos no funcionales, y parece que la forma más apropiada de hacerlo es mediante la integración del diseño de la arquitectura de forma conjunta con las demás actividades de ingeniería de requisitos, sobre todo cuando el sistema a desarrollar es de gran tamaño o incluye otros elementos

además de software, que es el enfoque de la mayoría de los estándares de desarrollo [DoD 1994, IEEE/EIA 1998] (ver apéndice A).

6.3 Bibliografía

- [Berry y Lawrence 1998] D. M. Berry y B. Lawrence. Requirements Engineering. *IEEE Software*, 15(2):26–29, Marzo/Abril 1998.
- [Boehm *et al.* 1984] B. W. Boehm, T. E. Gray, y T. Seewalt. Prototyping versus Specifying: A Multiproject Experiment. *IEEE Transactions on Software Engineering*, 10(3):290–303, Mayo 1984. También aparece en [Thayer y Dorfman 1990].
- [Cheng y Weiss 2000] B. H. C. Cheng y D. M. Weiss. Requirements Engineering: Integrating Technology. *IEEE Software*, 17(3):28–36, Mayo/Junio 2000.
- [Corchuelo 1999] R. Corchuelo. *Prototipado de Especificaciones de Sistemas Distribuidos Basadas en Retricciones. Aplicación al Lenguaje TESORO*. Tesis doctoral, Universidad de Sevilla, 1999.
- [Davis 1998] A. M. Davis. The Harmony in Rechoirments. *IEEE Software*, 15(2):6–8, Marzo/Abril 1998.
- [DoD 1994] DoD. *Military Standard 498: Software Development and Documentation*. Departament of Defense of the United States of America, 1994. Disponible en http://www-library.itsi.disa.mil/mil_std/498_win3.exe.
- [Galán 2000] F. J. Galán. *Formalizaciones para Sistemas Software Orientados a Objetos*. Tesis doctoral, Universidad de Sevilla, 2000. Pendiente de presentación.
- [García 2000] F. J. García. *Modelo de Reutilización Soportado por Estructuras Complejas de Reutilización Denominadas Mecanos*. Tesis doctoral, Universidad de Salamanca, 2000.
- [Goleman 1996] D. Goleman. *La Inteligencia Emocional*. Kairós, 1996.
- [Goleman 1999] D. Goleman. *La Práctica de la Inteligencia Emocional*. Kairós, 1999.

- [IEEE/EIA 1998] IEEE/EIA. Industry Implementation of International Standard for Information Technology ISO/IEC 12207 : Software Life Cycle Processes. IEEE/EIA Standard 12207.0–1996, Institute of Electrical and Electronics Engineers/Electronic Industries Alliance, 1998.
- [Kovitz 1998] B. L. Kovitz. *Practical Software Requirements: A Manual of Content & Style*. Manning, 1998.
- [Macaulay y Mylopoulos 1996] L. Macaulay y J. Mylopoulos. *Requirements Engineering Education and Training*, 1996. Disponible en <http://www.jrcase.mq.edu.au/~didar/seweb/training.html>.
- [Paulk *et al.* 1993] M. C. Paulk, B. Curtis, M. B. Chrissis, y C. V. Weber. Capability Maturity Model for Software, Version 1.1. Technical Report CMU/SEI-93-TR-024, Software Engineering Institute, Carnegie Mellon University, 1993. Disponible en <http://www.sei.cmu.edu>.
- [Sawyer *et al.* 1997] P. Sawyer, I. Sommerville, y S. Viller. Requirements Process Improvement through The Phased Introduction of Good Practice. *Software Process – Improvement and Practice*, 3(1), 1997. Disponible en <http://www.comp.lancs.ac.uk/computing/research/cseg/reaims/publications.html>.
- [Sheptunov 1999] Bogdan Sheptunov. *Comunicación privada sobre la aplicación de la MIL-STD-498*. Tessart Corp., 1999.
- [Sommerville y Sawyer 1997] I. Sommerville y P. Sawyer. *Requirements Engineering: A Good Practice Guide*. Wiley, 1997.
- [Thayer y Dorfman 1990] R. H. Thayer y M. Dorfman, editores. *System and Software Requirements Engineering*. IEEE Computer Society Press, 1990.

Apéndice A

La ingeniería de requisitos en algunas normas de desarrollo de software

En este apéndice se analizan los aspectos relacionados con la ingeniería de requisitos en algunas normas de desarrollo de software, principalmente las actividades que proponen y los productos que describen.

En concreto, se estudian dos normas norteamericanas, las del IEEE [IEEE 1997] y las del Departamento de Defensa de los Estados Unidos [DoD 1994], y dos normas europeas, las de la Agencia Espacial Europea [Mazza *et al.* 1994, Mazza *et al.* 1996] y la del Ministerio de Administraciones Públicas de España [MAP 1995].

A.1 Ingeniería de requisitos en la norma MIL-STD-498

Aunque el Departamento de Defensa de los Estados Unidos (*Department of Defense*, DoD) ha adoptado recientemente las normas IEEE 12207 como estándar para los desarrollos de software¹, la norma denominada *Military Standard: Software Development and Documentation (MIL-STD-498)* [DoD

¹El 27 de mayo de 1998 el DoD tomó la decisión de dejar de producir sus propias normas y utilizar normas *civiles* para la producción de software. La norma IEEE 12207, que se divide en tres documentos, 12207.1, 12207.2 y 12207.3, es la versión del IEEE de la norma ISO/IEC 12207 [ISO/IEC 1995]. Como paso previo, el DoD realizó una versión *desmilitarizada* de la MIL-STD-498, la J-STD-016 [EIA/IEEE 1996].

1994] y sus antecesoras la DOD-STD-2167A (para software de sistemas de armas) y la DOD-STD-7935A (para sistemas de información automatizados), ha tenido y aun tiene una gran influencia², por lo que hemos creído oportuno incluirla en este estudio.

La característica principal la norma MIL-STD-498 es su clara orientación hacia los productos más que al proceso para obtener dichos productos, sin imponer ningún ciclo de vida concreto ni ninguna notación específica.

De hecho, permite al cliente adaptar³ la norma a cada desarrollo específico con total libertad, suprimiendo cualquiera de las 19 actividades o 22 productos⁴ que considere oportuno. La especificación de las tareas y productos deseados por el cliente debe recogerse en el documento denominado *Plan de Desarrollo de Software* (*Software Development Plan*, SDP).

En esta norma se recogen actividades de ingeniería de requisitos a dos niveles: de sistema y de software, que coinciden con la dimensión de *ámbito* que se describió en la sección 1.3.1 (pág. 12). En el caso de los sistemas hardware y software, la norma sólo cubre el desarrollo de la parte software (ver figura A.1).

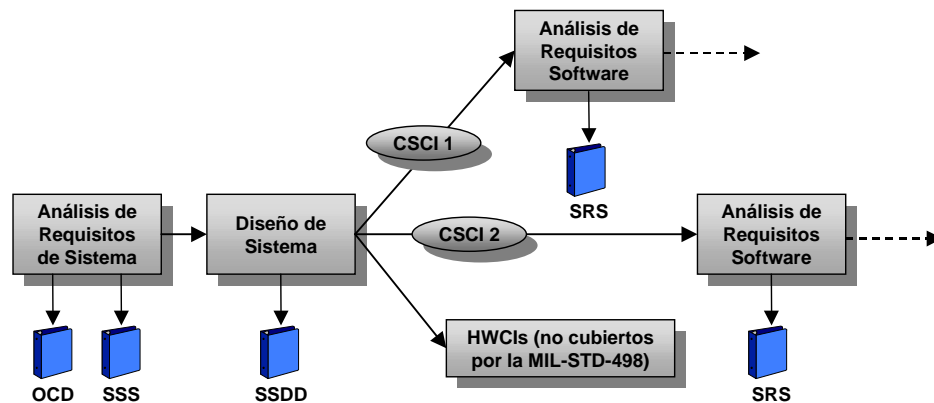


Figura A.1: MIL-STD-498: sistema hardware–software

²Sobre todo en las normas IEEE.

³El termino inglés empleado para designar la adaptación de la norma es *tailoring*.

⁴Los productos de esta norma se describen en las 22 *Descripciones de Elementos de Datos* (*Data Item Descriptions*, DIDs) que la acompañan.

Las actividades relativas a ingeniería de requisitos son las tres siguientes:

1 **Análisis de los requisitos del sistema** (*System Requirements Analysis*)

Esta actividad se divide en las siguientes tareas:

1.1 **Análisis de la información del usuario** (*Analysis of User Input*)

En esta tarea se debe analizar la información aportada por los clientes y usuarios para poder realizar las tareas posteriores. La información del usuario puede obtenerse utilizando cualquier técnica habitual de elicitación de requisitos (ver capítulo 3).

1.2 **Concepto operacional** (*Operational Concept*)

En esta tarea se debe estudiar la situación actual, describir y justificar los cambios que habría que introducir en el sistema actual, identificar a los usuarios afectados, describir el nuevo sistema, describir mediante *escenarios operacionales*⁵ el funcionamiento del nuevo sistema, estudiar su impacto en la organización, las ventajas e inconvenientes que supone y las posibles alternativas. Toda esta información debe recogerse en el documento denominado *Descripción de Concepto Operacional* (*Operational Concept Description, OCD*)⁶, cuya estructura puede verse en la figura A.2 y que se comenta ampliamente en [Fairley y Thayer 1997].

Según la propia norma MIL-STD-498⁷, "el OCD se utiliza para obtener consenso entre el comprador (*acquirer*), desarrollador (*developer*), personal de mantenimiento (*support*) y agencias de usuarios (*user agencies*) sobre el concepto operacional (ideas generales de funcionamiento) del sistema propuesto. Dependiendo de su uso, un OCD puede centrarse en comunicar las necesidades del usuario al desarrollador o las ideas del desarrollador al usuario y otras partes interesadas."

Este documento debería realizarse siempre en cualquier desarrollo, aunque fuera un sistema compuesto únicamente por software [Sheptunov 1999].

1.3 **Requisitos de sistema** (*System Requirements*)

En esta tarea se deben identificar aquellas características del sistema que son condiciones para su aceptación, dejando para el

⁵Concepto similar a los casos de uso [Jacobson *et al.* 1993, Booch *et al.* 1999].

⁶El término *concept* se define en [Webster 1990] como idea o noción general. El *concepto operacional* podría entenderse como las ideas generales de funcionamiento del sistema.

⁷Apartado *Description/Purpose* del *Data Item Description* correspondiente al *Operational Concept Description* (DI-IPSIC-81430).

1	Ámbito
1.1	Identificación
1.2	Descripción del sistema
1.3	Descripción del documento
2	Documentos referenciados
3	Sistema o situación actual
3.1	Historia, objetivos y ámbito
3.2	Políticas operacionales y restricciones
3.3	Descripción del sistema o situación actual
3.4	Usuarios o personal involucrado
3.5	Concepto de apoyo
4	Justificación y naturaleza de los cambios
4.1	Justificación para los cambios
4.2	Descripción de los cambios necesarios
4.3	Prioridades de los cambios
4.4	Cambios considerados pero no incluidos
4.5	Suposiciones y restricciones
5	Concepto para un sistema nuevo o modificado
5.1	Historia, objetivos y ámbito
5.2	Políticas operacionales y restricciones
5.3	Descripción del sistema nuevo o modificado
5.4	Usuarios o personal afectado
5.5	Concepto de apoyo
6	Escenarios operacionales
7	Resumen de impactos
7.1	Impactos organizacionales
7.2	Impactos durante el desarrollo
8	Análisis del sistema propuesto
8.1	Resumen de ventajas
8.2	Resumen de desventajas/limitaciones
8.3	Alternativas consideradas
9	Notas
A	Apéndices

Figura A.2: Estructura del OCD de la norma MIL-STD-498

diseño aquellas características que el comprador desea dejar en manos del desarrollador. Estos requisitos del sistema deben recogerse en el documento denominado *Especificación de Sistema/Subsistema* (*System/Subsystem Specification, SSS*), cuya estructura puede verse en la figura A.3. Si el sistema tiene que soportar interfaces con otros sistemas, los requisitos relativos a dichos interfaces pueden recogerse en el propio SSS o en uno o más documentos aparte denominados *Especificaciones de Requisitos de Interfaz* (*Interface Requirements Specifications, IRSs*).

2 Diseño del sistema (*System Design*)

En esta actividad se deben realizar las siguientes tareas:

2.1 Decisiones de diseño a nivel de sistema (*System-wide Design Decisions*)

En esta tarea se deben tomar decisiones sobre la selección y diseño de los componentes del sistema, que se desarrollarán posteriormente. Esta información debe recogerse en el documento denominado *Descripción de Diseño de Sistema/Subsistema* (*System/Subsystem Design Description, SSDD*), cuya estructura puede verse en la figura A.4. El diseño relativo a interfaces con otros sistemas puede incluirse en el propio SSDD o en uno o más documentos aparte denominados *Descripciones de Diseño de Interfaz* (*Interface Design Descriptions, IDD*s). El diseño relativo a bases de datos puede recogerse en el SSDD o en uno o más documentos aparte denominados *Descripciones de Diseño de Base de Datos* (*Database Design Descriptions, DBDD*s).

2.2 Diseño de la arquitectura del sistema (*System Architectural Design*)

En esta tarea se deben definir los componentes del sistema (hardware, software u operaciones manuales)⁸ y sus interfaces. También se debe establecer la rastreabilidad entre los requisitos del sistema y los componentes para determinar qué requisitos de sistema deberá cumplir cada componente.

3 Análisis de Requisitos Software (*Software Requirements Analysis*)

Esta actividad no se divide en tareas. En ella se deben definir los requisitos software para cada CSCI (*Computer Software Configuration*

⁸A los componentes hardware se les denomina en esta norma *Hardware Configuration Items* (HWCIs), y a los componentes software *Computer Software Configuration Items* (CSCIs). Como ya se comentó, la norma MIL-STD-498 no cubre el desarrollo de los HWCIs.

1	Ámbito
1.1	Identificación
1.2	Descripción del sistema
1.3	Descripción del documento
2	Documentos referenciados
3	Requisitos
3.1	Modos y estados requeridos
3.2	Requisitos de capacidades del sistema
3.2.x	(capacidad del sistema)
3.3	Requisitos de interfaces externas del sistema
3.3.1	Identificación y diagramas de interfaces
3.3.x	(interfaz)
3.4	Requisitos de interfaces internas del sistema
3.5	Requisitos de datos internos del sistema
3.6	Requisitos de adaptación
3.7	Requisitos de seguridad
3.8	Requisitos de privacidad
3.9	Requisitos de entorno del sistema
3.10	Requisitos de recursos informáticos
3.10.1	Requisitos de hardware
3.10.2	Requisitos de utilización de recursos hardware
3.10.3	Requisitos de software
3.10.4	Requisitos de comunicaciones
3.11	Factores de calidad del sistema
3.12	Restricciones de diseño y construcción
3.13	Requisitos de personal
3.14	Requisitos de formación
3.15	Requisitos de logística
3.16	Otros requisitos
3.17	Requisitos de empaquetado
3.18	Prioridad e importancia de los requisitos
4	Disposiciones de calificación
5	Rastreabilidad de requisitos
6	Notas
A	Apéndices

Figura A.3: Estructura del SSS de la norma MIL-STD-498

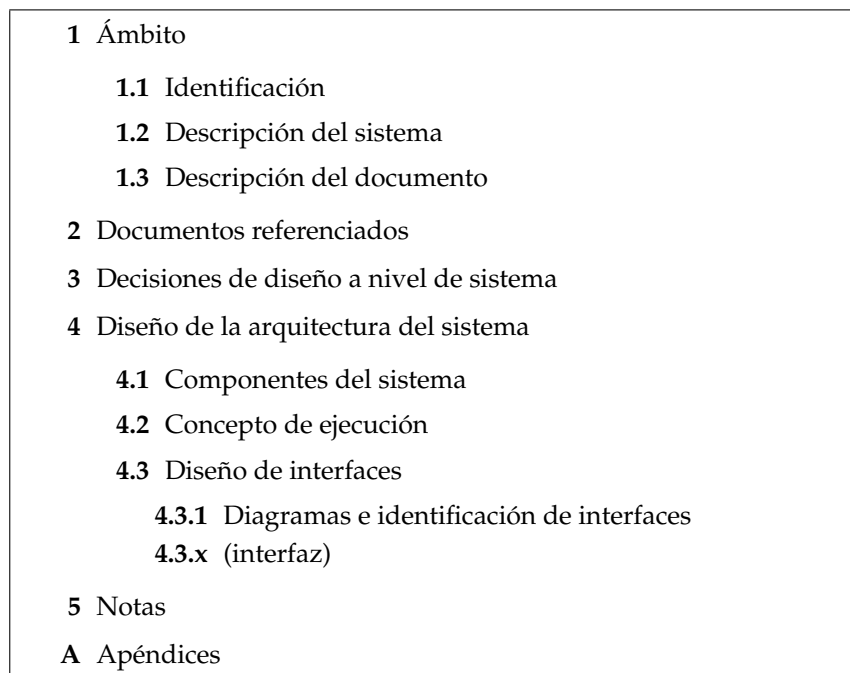


Figura A.4: Estructura del SSDD de la norma MIL-STD-498

Item) identificado durante el análisis de requisitos del sistema. Para cada elemento de desarrollo software o CSCI se elaborará una *Especificación de Requisitos Software* (*Software Requirements Specification*, SRS), cuya estructura puede verse en la figura A.5. En el caso de los requisitos relativos a las interfaces de cada CSCI, éstos pueden incluirse en el SRS o en uno o más documentos aparte denominados *Especificaciones de Requisitos de Interfaz* (*Interface Requirements Specification*, ISRs). También se debe establecer la rastreabilidad entre requisitos del sistema y requisitos software de cada CSCI.

En las figuras A.1, A.6 y A.7 se muestran posibles adaptaciones de la aplicación de la norma según se trate de sistemas hardware–software, sólo software con varios CSCIs o sólo software con un sólo CSCI. En este último caso se muestran dos posibles alternativas en las que sólo se elabora un documento de requisitos, ya que el sistema y el único CSCI coinciden.

Las actividades relativas a la validación de requisitos se llevan a cabo, junto con las de verificación, en evaluaciones y revisiones técnicas conjuntas (*Joint Technical Reviews*), que deben indicarse en el plan de desarrollo. Los criterios de aceptación de los distintos documentos, bastante generales, pueden consultarse en [DoD 1994, Apéndice D].

1	Ámbito
1.1	Identificación
1.2	Descripción del sistema
1.3	Descripción del documento
2	Documentos referenciados
3	Requisitos
3.1	Modos y estados requeridos
3.2	Requisitos de capacidades del CSCI
3.2.x	(requisito del CSCI)
3.3	Requisitos de interfaces externas del CSCI
3.3.1	Identificación y diagramas de interfaces
3.3.x	(interfaz)
3.4	Requisitos de interfaces internas del CSCI
3.5	Requisitos de datos internos del CSCI
3.6	Requisitos de adaptación
3.7	Requisitos de seguridad
3.8	Requisitos de privacidad
3.9	Requisitos de entorno del CSCI
3.10	Requisitos de recursos informáticos
3.10.1	Requisitos de hardware
3.10.2	Requisitos de utilización de recursos hardware
3.10.3	Requisitos de software
3.10.4	Requisitos de comunicaciones
3.11	Factores de calidad del software
3.12	Restricciones de diseño y construcción
3.13	Requisitos de personal
3.14	Requisitos de formación
3.15	Requisitos de logística
3.16	Otros requisitos
3.17	Requisitos de empaquetado
3.18	Prioridad e importancia de los requisitos
4	Disposiciones de calificación
5	Rastreabilidad de requisitos
6	Notas
A	Apéndices

Figura A.5: Estructura del SRS de la norma MIL-STD-498

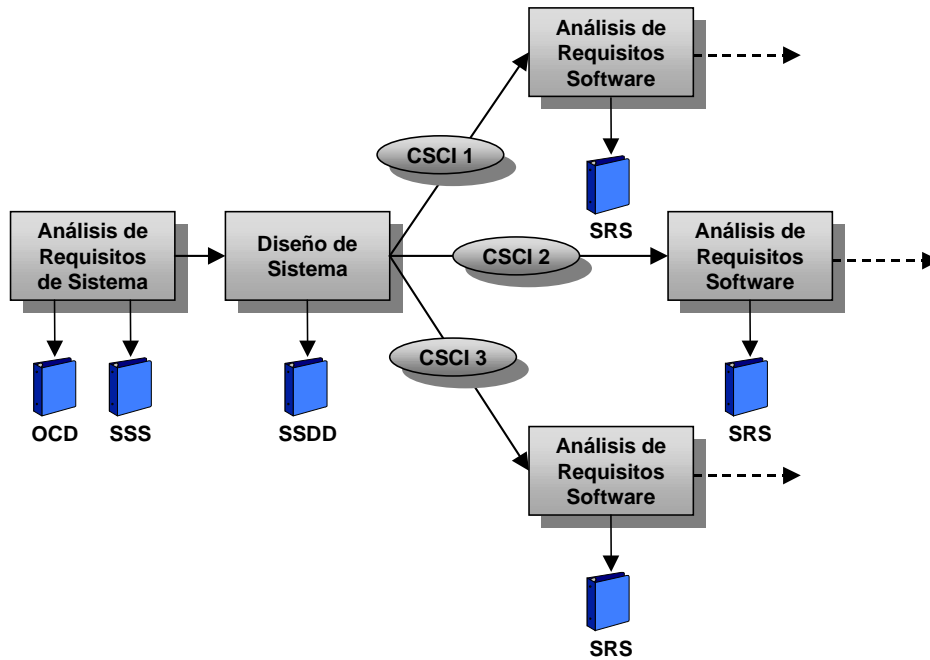


Figura A.6: MIL-STD-498: sistema sólo software con varios CSCIs

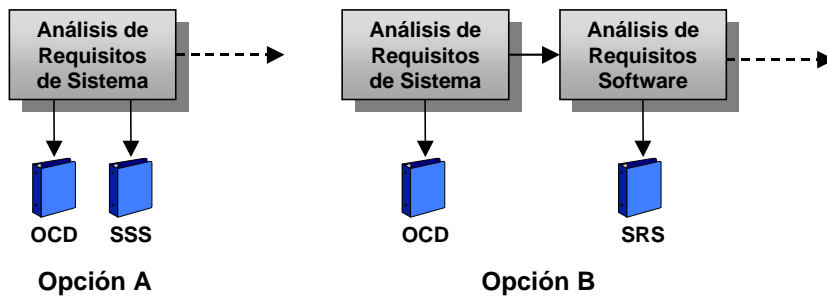


Figura A.7: MIL-STD-498: sistema sólo software con un sólo CSCI

A.2 Ingeniería de requisitos en las normas IEEE

Como se comentó en la sección anterior, las normas IEEE 12207.0, 12207.1 y 12207.2 son la versión norteamericana de la norma ISO/IEC 12207 [ISO/IEC 1995, Singh 1999, Esteban y Piattini 1995].

La norma IEEE 12207.0 [IEEE/EIA 1998c] coincide casi en su totalidad con la ISO/IEC 12207, mientras que la IEEE 12207.1 [IEEE/EIA 1998a] es una guía para la documentación y la [IEEE/EIA 1998b] es una guía para la implementación de ciclos de vida.

Dentro de los procesos del ciclo de vida del software (ver figura A.8), las actividades de ingeniería de requisitos corresponden básicamente al proceso de desarrollo, aunque todos los procesos de apoyo y de organización también afectan a la ingeniería de requisitos, sobre todo los de documentación, verificación, validación y resolución de problemas.

El esquema de actividades de ingeniería de requisitos (ver figura A.9) es muy similar al de la norma MIL-STD-498 comentada en la sección anterior. Su descripción es la siguiente:

- 1 Análisis de los requisitos del sistema** (*System Requirements Analysis*)
En esta actividad se deben especificar y evaluar⁹ los requisitos a nivel de sistema.

Para documentar estos requisitos se puede utilizar la norma IEEE-1233, *Guide for Developing System Requirements Specifications* [IEEE 1996], que en su anexo A propone la estructura que puede verse en la figura A.10 para el documento correspondiente a la especificación de requisitos de sistema.

- 2 Diseño de la arquitectura del sistema** (*System Architecture Design*)
En esta actividad se debe establecer la arquitectura a alto nivel del sistema, identificar los elementos de hardware, software y procesos manuales que compondrán el sistema, y evaluar la arquitectura resultante.

Dentro de las normas IEEE, no existe ninguna específica para documentar el diseño de la arquitectura del sistema, ya que las normas IEEE-1016 [IEEE 1987, IEEE 1993a] son para diseño a nivel de software y no de sistema.

⁹Dentro del contexto de las normas ISO/IEC e IEEE 12207, *evaluar* equivale a *verificar* en el sentido que se le dio en el capítulo 5.

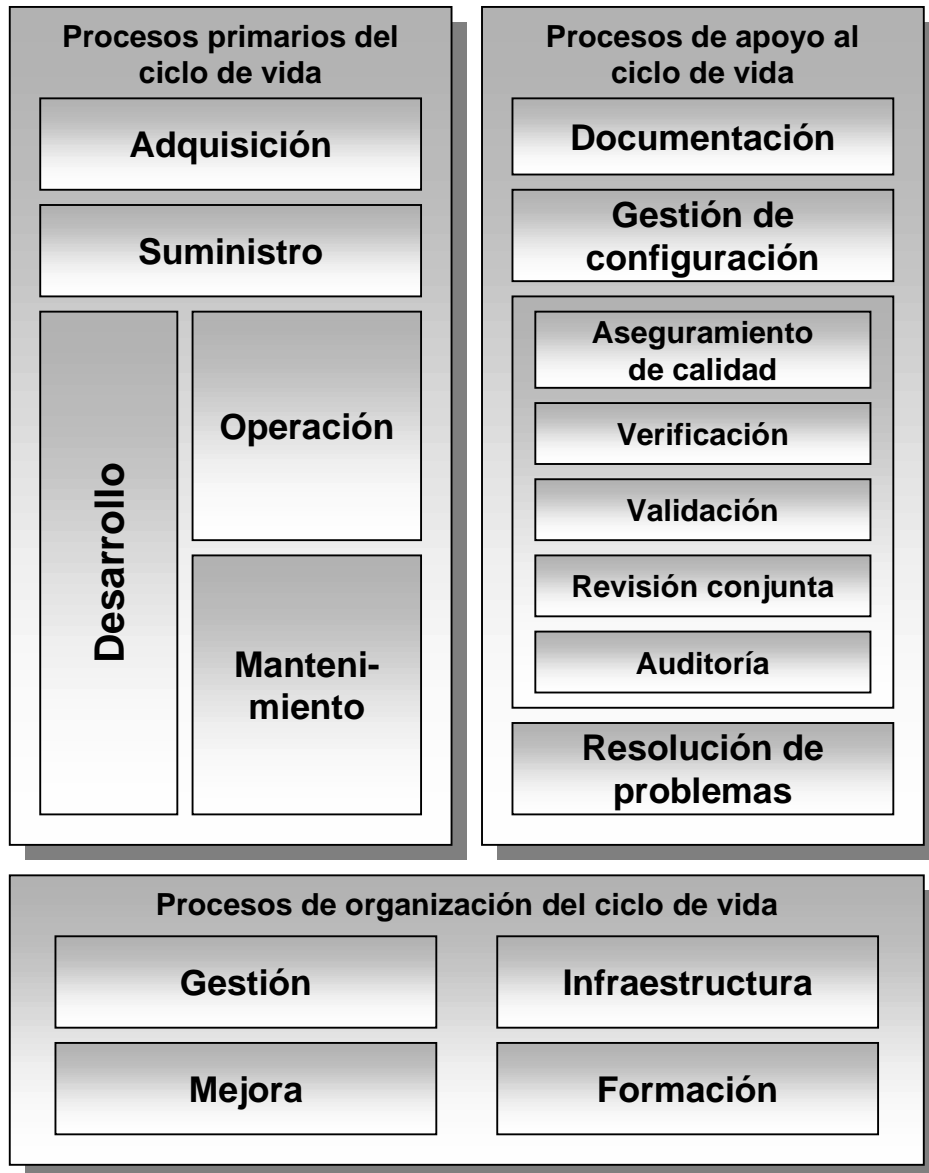


Figura A.8: Procesos del ciclo de vida del software según la norma ISO/IEC/IEEE 12207

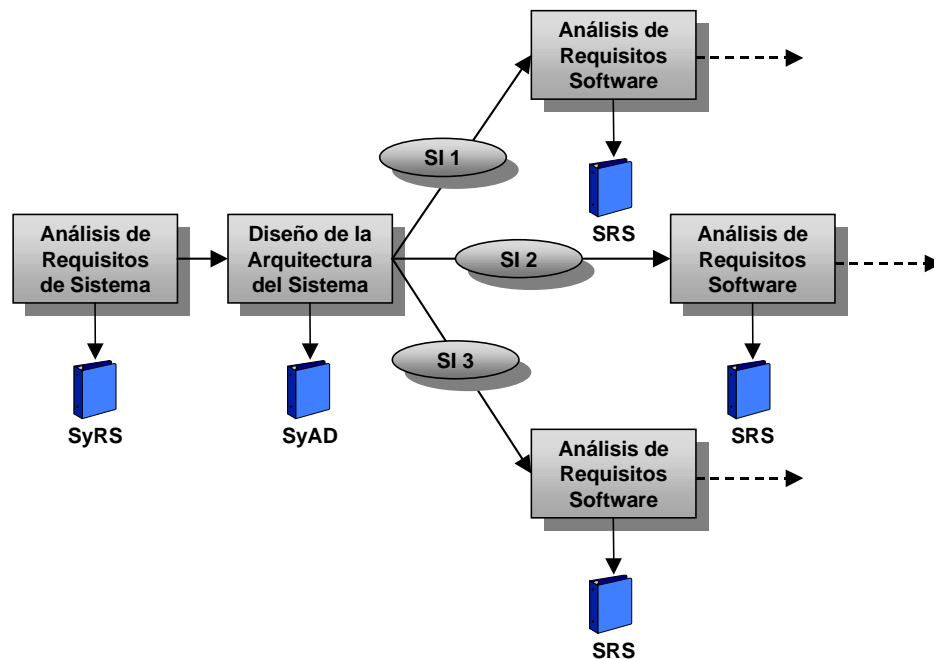


Figura A.9: IEEE 12207: sistema con varios elementos software

Una posibilidad planteada en [IEEE/EIA 1998a] es utilizar los documentos descritos en la J-STD-016 [EIA/IEEE 1996], prácticamente idénticos a los definidos en la MIL-STD-498, tal como se describe en [Moore *et al.* 1997].

3 Análisis de Requisitos Software (*Software Requirements Analysis*)

En esta actividad se deben establecer, evaluar y revisar los requisitos correspondiente a los elementos software (*software items*) identificados en la actividad anterior.

Para documentar estos requisitos se puede utilizar la norma IEEE-830, *Recommended Practice for Software Requirements Specifications* [IEEE 1993b], que propone la estructura que puede verse en la figura A.11 para el documento correspondiente a la especificación de los requisitos software. Para la organización del apartado dedicado a los requisitos, la norma propone varios formatos en su anexo A: organización por modos (estados) del sistema, por tipos de usuarios, por objetos, por características del sistema, por estímulos externos, por jerarquía funcional o por múltiples criterios.

1	Introducción
1.1	Propósito del sistema
1.2	Ámbito del sistema
1.3	Definiciones, acrónimos y abreviaturas
1.4	Referencias
1.5	Visión general del sistema
2	Descripción general del sistema
2.1	Contexto del sistema
2.2	Estados y modos del sistema
2.3	Capacidades principales del sistema
2.4	Condiciones principales del sistema
2.5	Restricciones principales del sistema
2.6	Características de los usuarios
2.7	Suposiciones y dependencias
2.8	Escenarios operacionales
3	Capacidades, condiciones y restricciones del sistema
3.1	Características físicas
3.1.1	Construcción
3.1.2	Durabilidad
3.1.3	Adaptabilidad
3.1.4	Condiciones del entorno
3.2	Características de rendimiento del sistema
3.3	Seguridad del sistema
3.4	Gestión de información
3.5	Operaciones del sistema
3.5.1	Factores humanos del sistema
3.5.2	Mantenibilidad del sistema
3.5.3	Fiabilidad del sistema
3.6	Normas y reglamentos
3.7	Actividades de calidad en el ciclo de vida del sistema
4	Interfaces del sistema

Figura A.10: Estructura de la SyRS de la norma IEEE-1233

1	Introducción
1.1	Propósito
1.2	Ámbito
1.3	Definiciones, acrónimos y abreviaturas
1.4	Referencias
1.5	Visión general
2	Descripción general
2.1	Perspectiva del producto
1.2	Funciones del producto
1.3	Características de los usuarios
1.4	Restricciones
1.5	Suposiciones y dependencias
3	Requisitos específicos
3.1	Requisitos de interfaces externos
3.1.1	Interfaces de usuario
3.1.2	Interfaces hardware
3.1.3	Interfaces software
3.1.4	Interfaces de comunicación
3.2	<i>Requisitos funcionales (varias posibilidades)</i>
3.3	Requisitos de rendimiento
3.4	Restricciones de diseño
3.5	Otros requisitos
	Apéndices
	Índice

Figura A.11: Estructura de la SRS de la norma IEEE-830

A.3 Ingeniería de requisitos en la norma ESA PSS-05

La norma para desarrollo de software de la Agencia Espacial Europea (*European Space Agency*, ESA) se denomina *ESA Software Engineering Standards* (ESA PSS-05-0 Issue 2) [Mazza *et al.* 1994]. Acompañando a la norma existen unas guías para su aplicación [Mazza *et al.* 1996].

Esta norma no permite una adaptación tan flexible como la MIL-STD-498, obligando a realizar una serie de fases y a obtener una serie de productos concretos. Tampoco abarca los requisitos a nivel de sistema, centrándose únicamente en requisitos software y exige claramente la necesidad de elaborar un modelo del sistema a desarrollar, en concreto un modelo orientado a procesos¹⁰, algo que no ocurre en normas como la MIL-STD-498 o las IEEE.

Esta norma se encuentra en periodo de revisión para permitir utilizar técnicas orientadas a objetos, aunque no se esperan cambios significativos en la primera fase, *Definición de Requisitos de Usuario* [Ruggier 1998].

Las fases de la PSS-05 dedicadas a la ingeniería de requisitos son las siguientes (ver figura A.12):

1 Definición de Requisitos de Usuario (*User Requirements Definition*)

Esta fase podría denominarse, según la propia norma, *Fase de Definición del Problema*, y es considerada como responsabilidad de los usuarios, aunque pueden necesitar ayuda por parte de los desarrolladores, que también participan en la revisión de los resultados¹¹.

Las actividades, relativas a la ingeniería de requisitos, a realizar en esta fase son:

1.1 Captura de Requisitos de Usuario (*Capture of User Requirements*)

En esta actividad se debe obtener información sobre las necesidades de los usuarios mediante las técnicas habituales de elicitación de requisitos (ver capítulo 3), incluyendo un estudio del sistema actual en el caso de que exista.

Esta información, al igual que la de las siguientes tareas, debe recogerse en el *Documento de Requisitos de Usuario* (*User Re-*

¹⁰Es decir, utilizando técnicas estructuradas, principalmente DFDs.

¹¹Obsérvese el enfoque tradicional en el que la definición de los requisitos-C se considera responsabilidad, principalmente, de los clientes y usuarios.

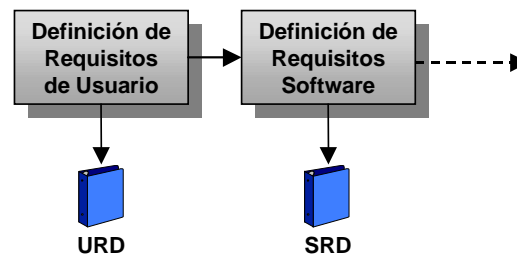


Figura A.12: Fases de ingeniería de requisitos en la norma ESA PSS-05

quirements Document, URD), basado en la norma IEEE 830-1993 [IEEE 1993b] y cuyo esquema puede verse en la figura A.13.

1.2 Determinación del Entorno Operacional (*Determination of Operational Environment*)

En esta actividad se debe describir el entorno en el que operará el software, estableciendo claramente el contexto del problema. Se deben describir también las interfaces que tendrá el sistema con otros sistemas que, si se considera oportuno, pueden describirse dentro del URD o en un *Documento de Control de Interfaz* (*Interface Control Document*, ICD) aparte.

1.3 Especificación de Requisitos de Usuario (*Specification of User Requirements*)

En esta actividad se debe organizar la información obtenida en las dos actividades previas y expresarla adecuadamente en el URD.

1.4 Revisión de Requisitos de Usuario (*User Requirements Review*)

En esta actividad se debe revisar formalmente el URD. Todo el proceso de elaboración del URD es iterativo, de forma que durante su desarrollo se pueden realizar revisiones internas previas a la revisión formal. En la revisión del URD participan usuarios, desarrolladores, gestores y personal de control de calidad. El objetivo principal de esta revisión es asegurar que los requisitos de usuario son claros y completos, presentado una descripción general de los procesos que los usuarios esperan que el sistema a desarrollar realice.

En otras palabras, se trata de una *validación* de los requisitos de usuario, en el sentido que se le dio al término validación en el capítulo 5, previa al análisis de dichos requisitos mediante la construcción de un modelo.

2 Definición de Requisitos Software (*Software Requirements Definition*)

Esta fase podría denominarse, según la propia norma, *Fase de Análisis del Problema*, y es considerada como responsabilidad de los desarrolladores, aunque pueden participar, si se considera necesario, usuarios, ingenieros de sistemas, ingenieros de hardware u otro personal.

Las actividades, relativas a la ingeniería de requisitos, a realizar en esta fase son:

2.1 Examen del URD (*Examination of the URD*)

Si los desarrolladores no han tomado parte en la revisión de los requisitos de usuario, deben examinar el URD para comprobar que es entendible y que permite continuar con el resto de las actividades de esta fase.

2.2 Construcción del Modelo Lógico (*Construction of the Logical Model*)

En esta actividad se debe construir un modelo lógico del sistema a desarrollar, que hace que los requisitos software se entiendan como un todo, no de forma individual. La norma propone una división funcional jerárquica y obliga a que se utilice un método y una notación ampliamente conocidos, es decir, no permite usar notaciones *ad hoc*.

2.3 Especificación de los Requisitos Software (*Specification of the Software Requirements*)

En esta actividad se deben especificar los requisitos software, es decir, deben expresarse de forma rigurosa en el *Documento de Requisitos Software* (*Software Requirements Document*, SRD), basado también en la norma IEEE 830-1993 [IEEE 1993b].

En [Mazza *et al.* 1996, pág. 44] se especifica que los requisitos funcionales deben corresponderse uno a uno con los *nodos* (*procesos*) del modelo lógico (normalmente basado en DFDs).

2.4 Revisión de los Requisitos Software (*Software Requirements Review*)

En esta actividad se debe revisar formalmente el SRD, que puede haber pasado revisiones internas previas. Esta revisión se considera una revisión técnica, en la que pueden participar usuarios, personal de operaciones, desarrolladores y gestores. El objetivo de esta revisión es asegurar que el URD especifica los requisitos de forma clara, completa y suficientemente detallada como para permitir que comience el proceso de diseño.

1	Introducción
1.1	Propósito
1.2	Ámbito
1.3	Definiciones, acrónimos y abreviaturas
1.4	Referencias
1.5	Estructura del documento
2	Descripción general
2.1	Perspectiva del producto
2.2	Capacidades generales
2.3	Restricciones generales
2.4	Características de los usuarios
2.5	Entorno operacional
2.6	Suposiciones y dependencias
3	Requisitos específicos
3.1	Requisitos de capacidades
3.2	Requisitos de restricciones
A	Apéndices

Figura A.13: Estructura del URD de la norma ESA PSS-05

- 1** Introducción
 - 1.1** Propósito
 - 1.2** Ámbito
 - 1.3** Definiciones, acrónimos y abreviaturas
 - 1.4** Referencias
 - 1.5** Estructura del documento
- 2** Descripción general
 - 2.1** Relación con proyectos actuales
 - 2.2** Relación con proyectos previos y posteriores
 - 2.3** Función y propósito
 - 2.4** Consideraciones de entorno
 - 2.5** Relación con otros sistemas
 - 2.6** Restricciones generales
 - 2.7** Descripción del modelo
- 3** Requisitos específicos
 - 3.1** Requisitos funcionales
 - 3.2** Requisitos de rendimiento
 - 3.3** Requisitos de interfaz
 - 3.4** Requisitos de operacionales
 - 3.5** Requisitos de recursos
 - 3.6** Requisitos de verificación
 - 3.7** Requisitos de pruebas de aceptación
 - 3.8** Requisitos de documentación
 - 3.9** Requisitos de privacidad
 - 3.10** Requisitos de portabilidad
 - 3.11** Requisitos de calidad
 - 3.12** Requisitos de fiabilidad
 - 3.13** Requisitos de mantenibilidad
 - 3.14** Requisitos de seguridad
- 4** Matriz de rastreabilidad de requisitos
- A** Apéndices

Figura A.14: Estructura del SRD de la norma ESA PSS-05

A.4 Ingeniería de requisitos en Métrica

La norma de desarrollo del Ministerio de Administraciones Públicas de España se denomina Métrica (*Metodología de Planificación y Desarrollo de Sistemas de Información*), de la que actualmente se encuentra vigente su versión 2.1 [MAP 1995] y cuya versión 3 [CSJ 2000] es de inminente aparición.

La estructura de los aspectos de ingeniería de requisitos en ambas versiones de Métrica es similar, tal como puede verse en las figuras A.15 y A.18. Esta estructura es también muy similar a la de otras normas comentadas en este apéndice como la MIL-STD-498 o la ISO 12207.

Básicamente, se comienza a nivel de sistema para identificar una serie de elementos software durante el *Plan de Sistemas de Información* (PSI), denominados *aplicaciones* (APL) en Métrica, que se desarrollan posteriormente por separado.

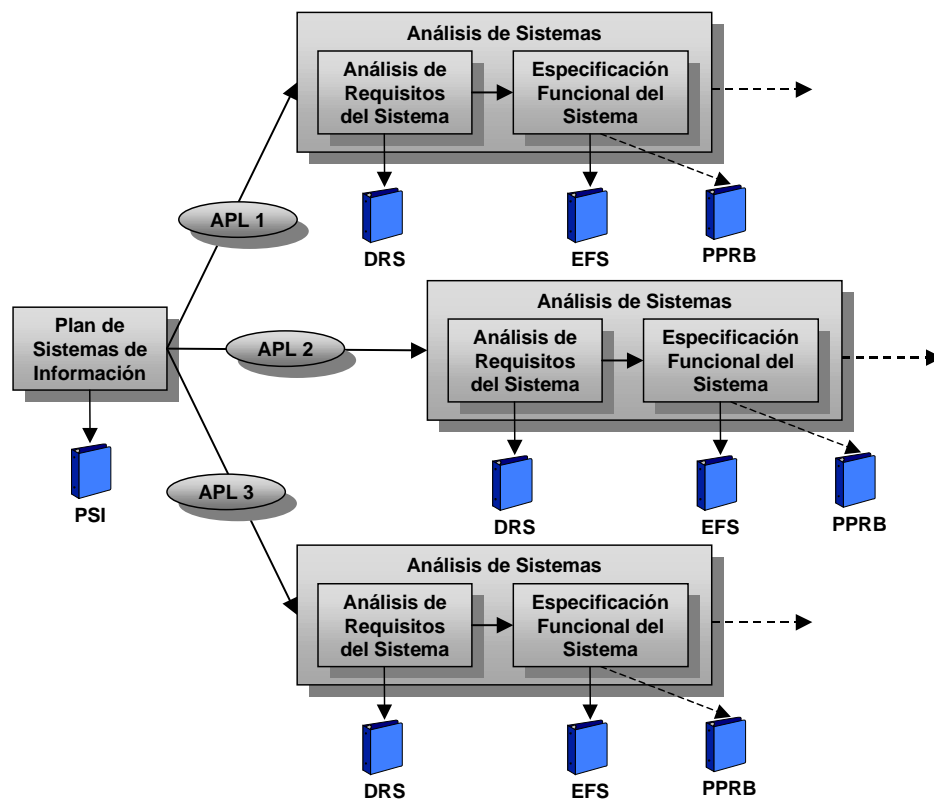


Figura A.15: Métrica 2.1: ingeniería de requisitos

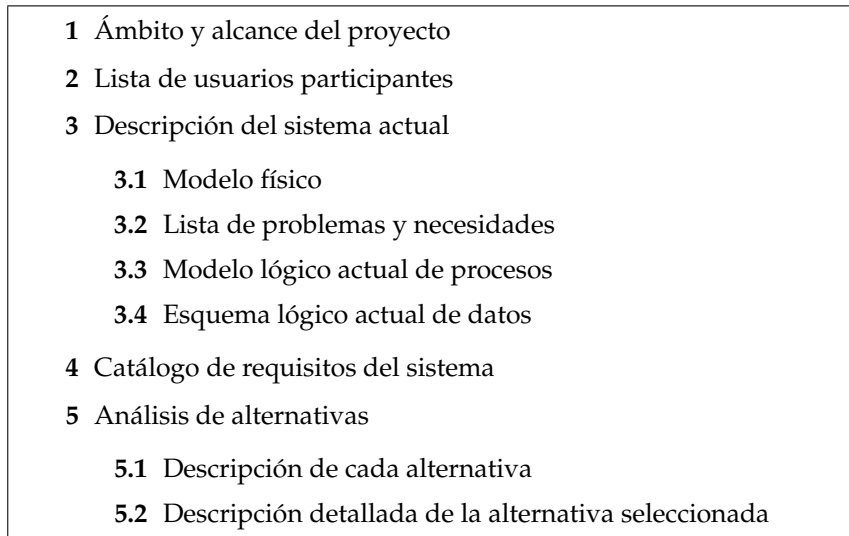


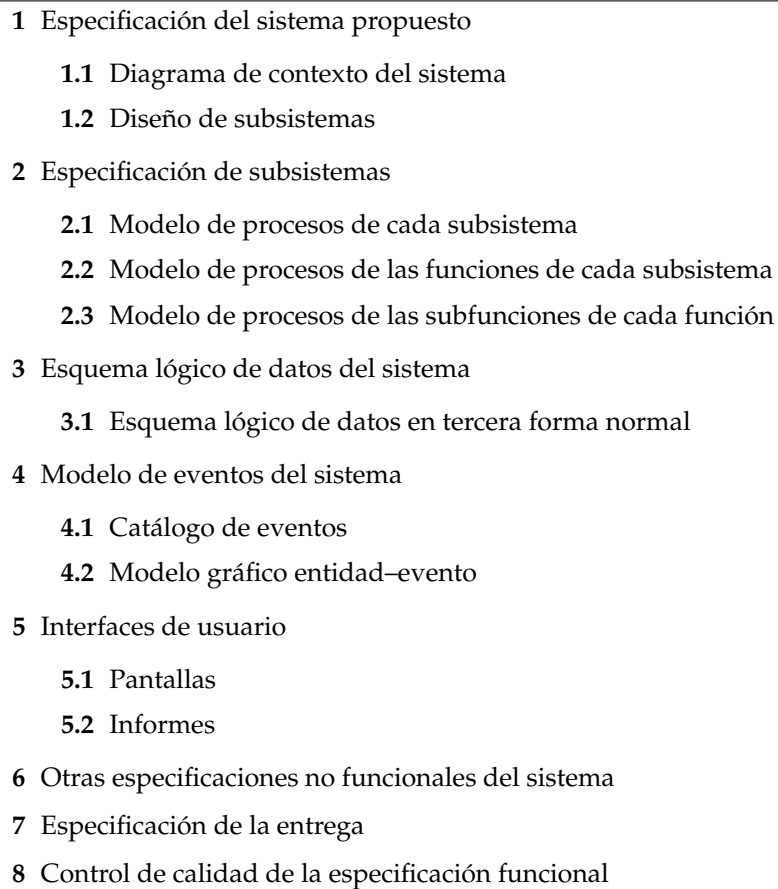
Figura A.16: Estructura del DRS de Métrica V2.1

En Métrica V2.1, las actividades relacionadas con la ingeniería de requisitos a nivel de software se encuadran dentro de la fase de *Análisis de Sistema* (AS), dentro de la cual se definen los dos siguientes módulos:

1 Análisis de Requisitos del Sistema

Este módulo, cuyo producto es el *Documento de Requisitos del Sistema* (DRS, ver figura A.16), se divide en 4 actividades en las que se debe:

- Estudiar el contexto en el que se va utilizar el software a desarrollar, normalmente mediante el estudio y modelado del sistema actual (actividad ARS 3, *Diseñar el Modelo y Esquema Lógico Actual*), e identificar a los usuarios participantes en el proceso de desarrollo y los objetivos del proyecto (actividad ARS 1, *Establecer el Ámbito y Alcance del Proyecto*).
- Preparar las sesiones de elicitación, obtener los requisitos del sistema a desarrollar y recogerlos en el *catálogo de requisitos* del DRS (actividad ARS 2, *Identificar y Definir Requisitos*).
- Estudiar distintas alternativas de solución (desarrollo a medida, adquisición o modificación de software de terceros, etc.) y seleccionar una de ellas (actividad ARS 4, *Estudiar Alternativas de Construcción*).

- 
- ```

graph TD
 1[1 Especificación del sistema propuesto] --> 1.1[1.1 Diagrama de contexto del sistema]
 1 --> 1.2[1.2 Diseño de subsistemas]
 2[2 Especificación de subsistemas] --> 2.1[2.1 Modelo de procesos de cada subsistema]
 2 --> 2.2[2.2 Modelo de procesos de las funciones de cada subsistema]
 2 --> 2.3[2.3 Modelo de procesos de las subfunciones de cada función]
 3[3 Esquema lógico de datos del sistema] --> 3.1[3.1 Esquema lógico de datos en tercera forma normal]
 4[4 Modelo de eventos del sistema] --> 4.1[4.1 Catálogo de eventos]
 4 --> 4.2[4.2 Modelo gráfico entidad–evento]
 5[5 Interfaces de usuario] --> 5.1[5.1 Pantallas]
 5 --> 5.2[5.2 Informes]
 6[6 Otras especificaciones no funcionales del sistema]
 7[7 Especificación de la entrega]
 8[8 Control de calidad de la especificación funcional]

```
- 1 Especificación del sistema propuesto
    - 1.1 Diagrama de contexto del sistema
    - 1.2 Diseño de subsistemas
  - 2 Especificación de subsistemas
    - 2.1 Modelo de procesos de cada subsistema
    - 2.2 Modelo de procesos de las funciones de cada subsistema
    - 2.3 Modelo de procesos de las subfunciones de cada función
  - 3 Esquema lógico de datos del sistema
    - 3.1 Esquema lógico de datos en tercera forma normal
  - 4 Modelo de eventos del sistema
    - 4.1 Catálogo de eventos
    - 4.2 Modelo gráfico entidad–evento
  - 5 Interfaces de usuario
    - 5.1 Pantallas
    - 5.2 Informes
  - 6 Otras especificaciones no funcionales del sistema
  - 7 Especificación de la entrega
  - 8 Control de calidad de la especificación funcional

Figura A.17: Estructura del EFS de Métrica V2.1

## 2 Especificación Funcional del Sistema

Este módulo, cuyo producto es el documento denominado *Especificación Funcional del Sistema* (EFS, ver figura A.17), se divide en 6 actividades en las que se debe:

- Construir un modelo conceptual del sistema a desarrollar mediante técnicas estructuradas. En concreto, se utilizan DFDs para el modelado de los aspectos funcionales (actividad EFS 1, *Construir el Modelo de Procesos del Nuevo Sistema*), diagramas de estructura de datos<sup>12</sup> para el modelado estático (actividad EFS 2, *Construir el Esquema Lógico de Datos del Nuevo Sistema*), y HVEs para el modelado dinámico (tarea EFS 3.1, *Construcción del Modelo Entidad–Evento*).

<sup>12</sup>Una variante muy simplificada de los diagramas entidad–interrelación.

- Especificar la interfaz de usuario del sistema mediante la construcción de prototipos que permitan a los usuarios tener una idea más concreta del producto que van a recibir (actividad EFS 4, *Definir Interfaces de Usuario*).
- Elicitar y especificar los requisitos no funcionales que no se hayan tenido en cuenta hasta el momento (actividad EFS 5, *Completar Especificaciones del Sistema*).
- Verificar y validar los modelos desarrollados con respecto a los requisitos elicitados (tarea EFS 6.3, *Verificación y Validación de la Especificación Funcional del Sistema*).
- Comenzar a desarrollar el *Plan de Pruebas del Sistema* (PPRB), que se completará en fases posteriores.

En el borrador de la versión 3 de Métrica [CSJ 2000], las actividades relacionadas con la ingeniería de requisitos no han sufrido grandes cambios comparadas con la versión anterior (ver figuras A.15 y A.18). Las diferencias más significativas son las siguientes:

- Se ha simplificado la estructura de Métrica V2.1 eliminando los *módulos*, de forma que la metodología se estructura en *procesos* → *actividades* → *tareas*, en lugar de hacerlo en *fases* → *módulos* → *actividades* → *tareas*.
- Se han cambiado los nombres de los procesos relacionados con la ingeniería de requisitos, de forma que el módulo *Análisis de Requisitos del Sistema* (ARS) pasa a ser el proceso *Estudio de Viabilidad del Sistema* (EVS) y el módulo *Especificación Funcional del Sistema* (EFS) pasa a ser el proceso *Análisis del Sistema de Información* (ASI).
- Se ha cambiado el orden de las actividades dentro del EVS con respecto al ARS, de forma que ahora se antepone el estudio de la situación actual (actividad AVS 2) a la definición de requisitos y al estudio de alternativas (actividades AVS 3–6).
- Se han introducido los casos de uso como técnica de elicitación y especificación de requisitos de forma similar a la propuesta en este trabajo, y se han incorporado técnicas orientadas a objetos para el modelado de los requisitos.
- Probablemente porque se trata aún de un borrador, no se incluyen definiciones detalladas de los documentos resultantes similares a las incluidas en la versión 2.1.

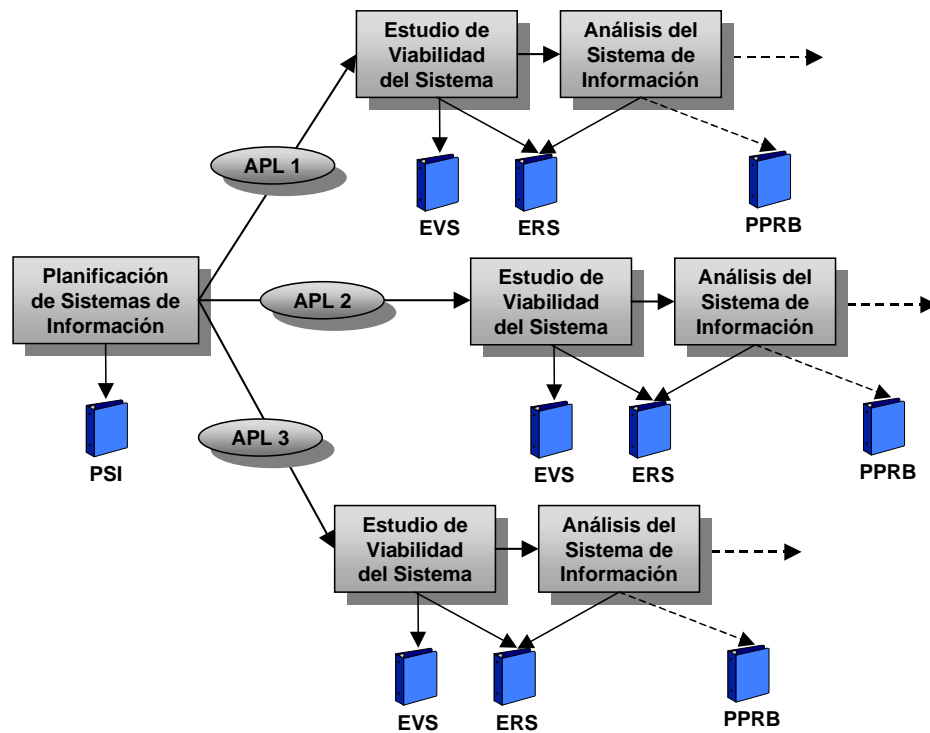


Figura A.18: Métrica 3: ingeniería de requisitos

## A.5 Conclusiones

En este apéndice se han estudiado los aspectos relativos a la ingeniería de requisitos en varias normas de desarrollo.

La estructura de las actividades que proponen es similar, comenzando a nivel de sistema para identificar y desarrollar posteriormente los distintos elementos software identificados.

En general, todas las normas estudiadas realizan una división entre requisitos-C y requisitos-D, aunque son las normas europeas (ESA-PSS-05 y Métrica) las que más inciden en esta diferencia.

Sin embargo, la nomenclatura empleada difiere bastante en cada caso. Así, las MIL-STD-498 incluye los requisitos-C en el denominado *concepto operacional*, la ISO 12207 no deja del todo claro dónde deben aparecer los requisitos-C, la ESA-PSS-05 los denomina *requisitos de usuario* y Métrica los incluye tanto en los productos resultantes de las actividades de elicitación (módulo ARS en la versión 2.1 y proceso EVS en la versión 3) como en los propiamente de análisis (módulo EFS en la versión 2.1 y proceso ASI en la versión 3).

Los requisitos-D suelen denominarse *requisitos software* en la mayoría de las normas consultadas, a menos que se establezcan a nivel de sistema, en cuyo caso se denominan *requisitos de sistema*.

Respecto a las técnicas propuestas, las normas militares norteamericanas y las IEEE no proponen ninguna técnica específica, mientras que en las normas ESA-PSS-05 y Métrica V2.1 es obligatorio el uso de técnicas estructuradas y en Métrica V3 pueden utilizarse tanto técnicas orientadas a objetos como estructuradas.

La propuesta de utilización de técnicas orientadas a objetos en la versión 3 de Métrica difiere bastante de la realizada en este trabajo. Las diferencias principales residen en el modelado de los aspectos dinámico-funcionales, ya que optan por el modelo de comunicación de paso de mensajes al utilizar los *diagramas de colaboración* y los *diagramas de secuencia* para modelar los casos de uso<sup>13</sup> y utilizan los *statechart* únicamente para la especificación del comportamiento de la interfaz de usuario.

## A.6 Bibliografía

[Booch *et al.* 1999] G. Booch, J. Rumbaugh, y I. Jacobson. *The Unified Modeling Language User Guide*. Addison-Wesley, 1999.

[CSJ 2000] CSJ. Metodología de Planificación, Desarrollo y Mantenimiento de Sistemas de Información. MÉTRICA Versión 3 (Borrador). Borrador, Consejo Superior de Informática, 2000. Disponible en <http://www.map.es/csi/pg5m42.htm>.

[DoD 1994] DoD. *Military Standard 498: Software Development and Documentation*. Department of Defense of the United States of America, 1994. Disponible en [http://www-library.itsi.disa.mil/mil\\_std/498\\_win3.exe](http://www-library.itsi.disa.mil/mil_std/498_win3.exe).

[Dorfman y Thayer 1990] M. Dorfman y R. H. Thayer, editores. *Standards, Guidelines, and Examples on System and Software Requirements Engineering*. IEEE Computer Society Press, 1990.

[EIA/IEEE 1996] EIA/IEEE. *Software Development and Documentation J-STD-016*. Standard, EIA, 1996. Este documento se incluye en [Dorf-

---

<sup>13</sup>Los inconvenientes de utilizar el modelo de comunicación de paso de mensajes se comentaron en el capítulo 4.

man y Thayer 1990] y en [IEEE 1997], ya que también es el estándar IEEE 1498.

[Esteban y Piattini 1995] J. L. Esteban y M. G. Piattini. Procesos del Ciclo de Vida del Software. *Novática*, (118), Noviembre–Diciembre 1995.

[Fairley y Thayer 1997] R. E. Fairley y R. H. Thayer. The Concept of Operations: The Bridge from Operational Requirements to Technical Specifications. En R. H. Thayer y M. Dorfman, editores, *Software Requirements Engineering*, páginas 73–83. IEEE Computer Society Press, 2ª edición, 1997.

[IEEE 1987] IEEE. IEEE Recommended Practice for Software Design Descriptions. IEEE/ANSI Standard 1016–1987, Institute of Electrical and Electronics Engineers, 1987.

[IEEE 1993a] IEEE. IEEE Guide to Software Design Descriptions. IEEE/ANSI Standard 1016.1–1993, Institute of Electrical and Electronics Engineers, 1993.

[IEEE 1993b] IEEE. IEEE Recommended Practice for Software Requirements Specifications. IEEE/ANSI Standard 830–1993, Institute of Electrical and Electronics Engineers, 1993.

[IEEE 1996] IEEE. IEEE Guide for Developing System Requirements Specifications. IEEE/ANSI Standard 1233–1996, Institute of Electrical and Electronics Engineers, 1996.

[IEEE 1997] IEEE. *IEEE Software Engineering Standards Collection*. Institute of Electrical and Electronics Engineers, 1997.

[IEEE/EIA 1998a] IEEE/EIA. *IEEE/EIA Guide for Information Technology: Software Life Cycle Processes — Life Cycle Data*, 1998.

[IEEE/EIA 1998b] IEEE/EIA. *IEEE/EIA Guide for Information Technology: Software Life Cycle Processes — Implementation Considerations*, 1998.

[IEEE/EIA 1998c] IEEE/EIA. Industry Implementation of International Standard for Information Technology ISO/IEC 12207 : Software Life Cycle Processes. IEEE/EIA Standard 12207.0–1996, Institute of Electrical and Electronics Engineers/Electronic Industries Alliance, 1998.

[ISO/IEC 1995] ISO/IEC. Information Technology–Software Life Cycle Processes. International Standard 12207 : 1995, International Organization for Standardization, 1995.



- [Jacobson *et al.* 1993] I. Jacobson, M. Christerson, P. Jonsson, y G. Övergaard. *Object–Oriented Software Engineering: A Use Case Driven Approach*. Addison–Wesley, 4ª edición, 1993.
- [MAP 1995] MAP. *Metodología de Planificación y Desarrollo de Sistemas de Información. MÉTRICA Versión 2.1*. Tecnos/Ministerio para las Administraciones Públicas, 1995.
- [Mazza *et al.* 1994] C. Mazza, J. Fairclough, B. Melton, D. de Pablo, A. Scheffer, y R. Stevens. *Software Engineering Standards*. Prentice–Hall, 1994. Disponible en <http://dxsting.cern.ch/sting/ESA.txt>.
- [Mazza *et al.* 1996] C. Mazza, J. Fairclough, B. Melton, D. de Pablo, A. Scheffer, R. Stevens, M. Jones, y G. Alvisi. *Software Engineering Guides*. Prentice–Hall, 1996.
- [Moore *et al.* 1997] J. W. Moore, P. R. DeWesse, y D. Rilling. *U. S. Software Lifecycle Process Standards*, 1997. Disponible en <http://www.stsc.hill.af.mil/crosstalk/1997/jul/lifecycle.asp>.
- [Ruggier 1998] M. Ruggier. *PSS–05 User Requirements Document Template*, 1998. Disponible en <http://framemaker.cern.ch/pss05>.
- [Sheptunov 1999] Bogdan Sheptunov. *Comunicación privada sobre la aplicación de la MIL–STD–498*. Tessart Corp., 1999.
- [Singh 1999] R. Singh. *An Introduction to International Standard ISO/IEC 12207*. Informe técnico, Federal Aviation Administration, 1999. Disponible en <http://www.abelia.com>.
- [Webster 1990] Webster. *Webster’s Dictionary of the English Language*. Portland House, 1990.



## Apéndice B

# Metodología para la elicitación de requisitos de sistemas de información

### B.1 Objetivo de la metodología

El objetivo de esta metodología es la definición de las tareas a realizar, los productos a obtener y las técnicas a emplear durante la actividad de *elicitación de requisitos* de la fase de *ingeniería de requisitos* del ciclo de vida de desarrollo del software.

En esta metodología se distinguen dos tipos de productos: los productos *entregables* y los productos *no entregables* o *internos*. Los productos *entregables* son aquellos que se entregan oficialmente al cliente como parte del desarrollo en fechas previamente acordadas, mientras que los *no entregables* son productos internos al desarrollo que no se entregan al cliente.

El único producto entregable definido en esta metodología es el *Documento de Requisitos del Sistema* (DRS), definido en la sección B.3.1, pág. 262.

La estructura de este documento es la siguiente: en la sección B.2 se describen las tareas recomendadas para obtener el DRS, en la sección B.3 se definen los productos entregables, en este caso el DRS, y por último, en la sección B.4 se describen algunas de las técnicas recomendadas para obtener los productos. También se incluye como apéndice un ejemplo de aplicación de esta metodología.

## **B.2 Tareas recomendadas**

Las tareas recomendadas para obtener los productos descritos en esta metodología son las siguientes:

**Tarea 1:** Obtener información sobre el dominio del problema y el sistema actual.

**Tarea 2:** Preparar y realizar las reuniones de elicitación/negociación.

**Tarea 3:** Identificar/revisar los objetivos del sistema.

**Tarea 4:** Identificar/revisar los requisitos de almacenamiento de información.

**Tarea 5:** Identificar/revisar los requisitos funcionales.

**Tarea 6:** Identificar/revisar los requisitos no funcionales.

**Tarea 7:** Priorizar objetivos y requisitos.

El orden recomendado de realización para estas tareas es: 1...7, aunque las tareas 4, 5, y 6 pueden realizarse simultáneamente o en cualquier orden que se considere oportuno. La tarea 1 es opcional y depende del conocimiento previo que tenga el equipo de desarrollo sobre el dominio del problema y el sistema actual.

En las siguientes secciones se describen cada una de las tareas mencionadas.

### **B.2.1 Tarea 1: Obtener información sobre el dominio del problema y el sistema actual**

#### **B.2.1.1 Objetivos**

- Conocer el dominio del problema.
- Conocer la situación actual.

### **B.2.1.2 Descripción**

Antes de mantener las reuniones con los clientes y usuarios e identificar los requisitos es fundamental conocer el dominio del problema y los contextos organizacional y operacional, es decir, la situación actual.

Enfrentarse a un desarrollo sin conocer las características principales ni el vocabulario propio de su dominio suele provocar que el producto final no sea el esperado por clientes ni usuarios.

Por otro lado, mantener reuniones con clientes y usuarios sin conocer las características de su actividad hará que probablemente no se entiendan sus necesidades y que su confianza inicial hacia el desarrollo se vea deteriorada enormemente.

Esta tarea es opcional, ya que puede que no sea necesario realizarla si el equipo de desarrollo tiene experiencia en el dominio del problema y el sistema actual es conocido.

### **B.2.1.3 Productos internos**

- Información recopilada: libros, artículos, folletos comerciales, desarrollos previos sobre el mismo dominio, etc.
- Modelos del sistema actual.

### **B.2.1.4 Productos entregables**

- Introducción, participantes en el proyecto, principalmente clientes y desarrolladores, y descripción del sistema actual como parte del DRS (ver secciones B.3.1.5–B.3.1.7, págs. 264–265).

### **B.2.1.5 Técnicas recomendadas**

- Obtener información de fuentes externas al negocio del cliente: folletos, informes sobre el sector, publicaciones, consultas con expertos, etc.

En el caso de que se trate de un dominio muy específico puede ser necesario recurrir a fuentes internas al propio negocio del cliente, en cuyo caso pueden utilizarse las técnicas auxiliares de elicitación de requisitos como el estudio de documentación, observación *in situ*,

cuestionarios, inmersión o *aprendizaje*, etc. (ver secciones B.4.1–B.4.3, págs. 267–268).

- Modelado del sistema actual [Laguna *et al.* 1999, García *et al.* 2000].

## **B.2.2 Tarea 2: Preparar y realizar las sesiones de elicitación/negociación**

### **B.2.2.1 Objetivos**

- Identificar a los usuarios participantes.
- Conocer las necesidades de clientes y usuarios.
- Resolver posibles conflictos.

### **B.2.2.2 Descripción**

Teniendo en cuenta la información recopilada en la tarea anterior, en esta tarea se deben preparar y realizar las reuniones con los clientes y usuarios participantes con objeto de obtener sus necesidades y resolver posibles conflictos que se hayan detectado en iteraciones previas del proceso.

Esta tarea es especialmente crítica y ha de realizarse con especial cuidado, ya que generalmente el equipo de desarrollo no conoce los detalles específicos de la organización para la que se va a desarrollar el sistema y, por otra parte, los clientes y posibles usuarios no saben qué necesita saber el equipo de desarrollo para llevar a cabo su labor.

### **B.2.2.3 Productos internos**

- Notas tomadas durante las reuniones, transcripciones o actas de reuniones, formularios, grabaciones en cinta o vídeo de las reuniones o cualquier otra documentación que se considere oportuna.

### **B.2.2.4 Productos entregables**

- Participantes en el proyecto, en concreto los usuarios participantes, como parte del DRS (ver sección B.3.1.6, pág. 264).

- Objetivos, requisitos o conflictos, que se hayan identificado claramente durante las sesiones de elicitación, como parte del DRS (ver secciones B.3.1.8–B.3.1.9 y B.3.1.17, págs. 265–265 y 267)

#### **B.2.2.5 Técnicas recomendadas**

- Técnicas de elicitación de requisitos (ver secciones B.4.1–B.4.3, págs. 267–268), incluyendo las plantillas de objetivos, requisitos y conflictos descritas en la sección B.4.5, pág. 268, que pueden usarse directamente durante las sesiones de elicitación.
- Técnicas de negociación como *WinWin* [Boehm *et al.* 1994].

### **B.2.3 Tarea 3: Identificar/revisar los objetivos del sistema**

#### **B.2.3.1 Objetivos**

- Identificar los objetivos que se esperan alcanzar mediante el sistema software a desarrollar.
- Revisar, en el caso de que haya conflictos, los objetivos previamente identificados.

#### **B.2.3.2 Descripción**

A partir de la información obtenida en la tarea anterior, en esta tarea se deben identificar qué objetivos se esperan alcanzar una vez que el sistema software a desarrollar se encuentre en explotación o revisarlos en función de los conflictos identificados. Puede que los objetivos hayan sido proporcionados antes de comenzar el desarrollo.

#### **B.2.3.3 Productos internos**

- No hay productos internos en esta tarea.

#### **B.2.3.4 Productos entregables**

- Objetivos del sistema como parte del DRS (ver sección B.3.1.8, pág. 265).

#### **B.2.3.5 Técnicas recomendadas**

- Análisis de factores críticos de éxito [MAP 1995] o alguna técnica similar de identificación de objetivos.
- Plantilla para especificar los objetivos del sistema

### **B.2.4 Tarea 4: Identificar/revisar los requisitos de almacenamiento de información**

#### **B.2.4.1 Objetivos**

- Identificar los requisitos de almacenamiento de información que deberá cumplir el sistema software a desarrollar.
- Revisar, en el caso de que haya conflictos, los requisitos de almacenamiento de información previamente identificados.

#### **B.2.4.2 Descripción**

A partir de la información obtenida en la tareas 1 y 2, y teniendo en cuenta los objetivos identificados en la tarea 3 y el resto de los requisitos, en esta tarea se debe identificar, o revisar si existen conflictos, qué información relevante para el cliente deberá gestionar y almacenar el sistema software a desarrollar.

Inicialmente se partirán de conceptos generales para posteriormente ir detallándolos hasta obtener todos los datos relevantes.

#### **B.2.4.3 Productos internos**

- No hay productos internos en esta tarea.

#### **B.2.4.4 Productos entregables**

- Requisitos de almacenamiento de información como parte del DRS (ver sección B.3.1.10, pág. 265).



#### **B.2.4.5 Técnicas recomendadas**

- Plantilla para requisitos de almacenamiento de información

### **B.2.5 Tarea 5: Identificar/revisar los requisitos funcionales**

#### **B.2.5.1 Objetivos**

- Identificar los actores del sistema del sistema software a desarrollar.
- Identificar los requisitos funcionales (casos de uso) que deberá cumplir el sistema software a desarrollar.
- Revisar, en el caso de que haya conflictos, los requisitos funcionales previamente identificados.

#### **B.2.5.2 Descripción**

A partir de la información obtenida en las tareas 1 y 2, y teniendo en cuenta los objetivos identificados en la tarea 3 y el resto de los requisitos, en esta tarea se debe identificar, o revisar si existen conflictos, qué debe hacer el sistema a desarrollar con la información identificada en la tarea anterior.

Inicialmente se identificarán los actores que interactuarán con el sistema, es decir aquellas personas u otros sistemas que serán los orígenes o destinos de la información que consumirá o producirá el sistema a desarrollar y que forman su entorno.

A continuación se identificarán los casos de uso asociados a los actores, los pasos de cada caso de uso y posteriormente se detallarán los casos de uso con las posibles excepciones hasta definir todas las situaciones posibles.

#### **B.2.5.3 Productos internos**

- No hay productos internos en esta tarea.

#### **B.2.5.4 Productos entregables**

- Requisitos funcionales como parte del DRS (ver sección B.3.1.11, pág. 265).

#### B.2.5.5 Técnicas recomendadas

- Casos de uso (ver sección B.4.4, pág. 268).
- Plantilla para actores
- Plantilla para los requisitos funcionales

### B.2.6 Tarea 6: Identificar/revisar los requisitos no funcionales

#### B.2.6.1 Objetivos

- Identificar los requisitos no funcionales del sistema software a desarrollar.

#### B.2.6.2 Descripción

A partir de la información obtenida en las tareas 1 y 2, y teniendo en cuenta los objetivos identificados en la tarea 3 y el resto de los requisitos, en esta tarea se deben identificar, o revisar si existen conflictos, los requisitos no funcionales, normalmente de carácter técnico o legal.

Algunos tipos de requisitos que se suelen incluir en esta sección son los siguientes:

##### **Requisitos de comunicaciones del sistema**

Son requisitos de carácter técnico relativos a las comunicaciones que deberá soportar el sistema software a desarrollar. Por ejemplo: *el sistema deberá utilizar el protocolo TCP/IP para las comunicaciones con otros sistemas.*

##### **Requisitos de interfaz de usuario**

Este tipo de requisitos especifica las características que deberá tener el sistema en su comunicación con el usuario. Por ejemplo: *la interfaz de usuario del sistema deberá ser consistente con los estándares definidos en IBM's Common User Access.*

Se debe ser cuidadoso con este tipo de requisitos, ya que en esta fase de desarrollo todavía no se conocen bien las dificultades que pueden surgir a la hora de diseñar e implementar las interfaces, por esto no es conveniente entrar en detalles demasiado específicos.

### **Requisitos de fiabilidad**

Los requisitos de fiabilidad deben establecer los factores que se requieren para la fiabilidad del software en tiempo de explotación. La fiabilidad mide la probabilidad del sistema de producir una respuesta satisfactoria a las demandas del usuario. Por ejemplo: *la tasa de fallos del sistema no podrá ser superior a 2 fallos por semana.*

### **Requisitos de entorno de desarrollo**

Este tipo de requisitos especifican si el sistema debe desarrollarse con un producto específico. Por ejemplo: *el sistema deberá desarrollarse con Oracle 7 como servidor y clientes Visual Basic 4.*

### **Requisitos de portabilidad**

Los requisitos de portabilidad definen qué características deberá tener el software para que sea fácil utilizarlo en otra máquina o bajo otro sistema operativo. Por ejemplo: *el sistema deberá funcionar en los sistemas operativos Windows 95, Windows 98 y Windows NT 4.0, siendo además posible el acceso al sistema a través de Internet usando cualquier navegador compatible con HTML 3.0.*

#### **B.2.6.3 Productos internos**

- No hay productos internos en esta tarea.

#### **B.2.6.4 Productos entregables**

- Requisitos no funcionales del sistema como parte del DRS (ver sección B.3.1.15, pág. 266).

#### **B.2.6.5 Técnicas recomendadas**

- Plantilla para requisitos no funcionales

## **B.3 Productos entregables**

El único producto entregable que se contempla en esta metodología es el *Documento de Requisitos del Sistema (DRS)*.

### B.3.1 Documento de requisitos del sistema

La estructura del DRS puede verse en la figura B.1. En las siguientes secciones se describe con detalle cada sección del DRS.

|                                                                                           |
|-------------------------------------------------------------------------------------------|
| Portada                                                                                   |
| Lista de cambios                                                                          |
| Índice                                                                                    |
| Lista de figuras                                                                          |
| Lista de tablas                                                                           |
| 1 Introducción                                                                            |
| 2 Participantes en el proyecto                                                            |
| 3 Descripción del sistema actual <i>[opcional]</i>                                        |
| 4 Objetivos del sistema                                                                   |
| 5 Catálogo de requisitos del sistema                                                      |
| 5.1 Requisitos de almacenamiento de información                                           |
| 5.2 Requisitos funcionales                                                                |
| 5.2.1 Diagramas de casos de uso                                                           |
| 5.2.2 Definición de actores                                                               |
| 5.2.3 Casos de uso del sistema                                                            |
| 5.3 Requisitos no funcionales                                                             |
| 6 Matriz de rastreabilidad objetivos/requisitos                                           |
| 7 Conflictos pendientes de resolución <i>[opcional, pueden ir en un documento aparte]</i> |
| 8 Glosario de términos <i>[opcional]</i>                                                  |
| Apéndices <i>[opcionales]</i>                                                             |

Figura B.1: Estructura del Documento de Requisitos del Sistema

#### B.3.1.1 Portada

La portada del DRS debe tener el formato que puede verse en la figura B.2. Los elementos que deben aparecer son los siguientes:

- **Nombre del proyecto:** el nombre del proyecto al que pertenece el DRS.

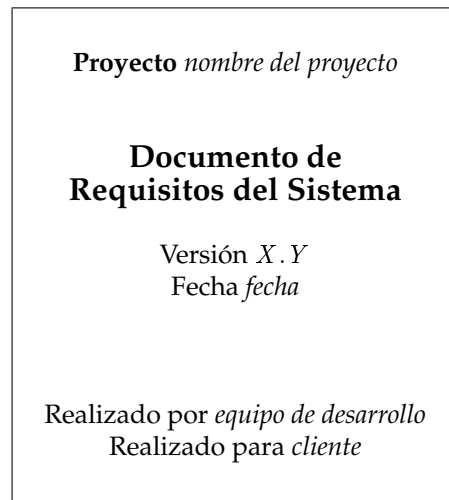


Figura B.2: Portada del Documento de Requisitos del Sistema

- **Versión:** la versión del DRS que se entrega al cliente. La versión se compone de dos números *X* e *Y*. El primero indica la versión, y se debe incrementar cada vez que se hace una nueva entrega formal al cliente. Cuando se incrementa el primer número, el segundo debe volver a comenzar en cero. El segundo número indica cambios dentro de la misma versión aún no entregada, y se debe incrementar cada vez que se publica una versión con cambios respecto a la última que se publicó y que no se vaya a entregar formalmente todavía. Este tipo de versiones pueden ser internas al equipo de desarrollo o ser entregadas al cliente a título orientativo.
- **Fecha:** fecha de la publicación de la versión.
- **Equipo de desarrollo:** nombre de la empresa o equipo de desarrollo.
- **Cliente:** nombre del cliente, normalmente otra empresa.

#### B.3.1.2 Lista de cambios

El documento debe incluir una lista de cambios en la que se especifiquen, para cada versión del documento, los cambios producidos en el mismo con un formato similar al que puede verse en la figura B.3. Para cada cambio realizado se debe incluir el número de orden, la fecha, una descripción y los autores.

| Núm.     | Fecha     | Descripción             | Autores   |
|----------|-----------|-------------------------|-----------|
| 0        | $fecha_0$ | Versión x.y             | $autor_0$ |
| 1        | $fecha_1$ | $descripción\ cambio_1$ | $autor_1$ |
| $\vdots$ | $\vdots$  | $\vdots$                | $\vdots$  |
| $n$      | $fecha_n$ | $descripción\ cambio_n$ | $autor_n$ |

Figura B.3: Lista de cambios del Documento de Requisitos del Sistema

### B.3.1.3 Índice

El índice del DRS debe indicar la página en la que comienza cada sección, subsección o apartado del documento. En la medida de lo posible, se sangrarán las entradas del índice para ayudar a comprender la estructura del documento.

### B.3.1.4 Listas de figuras y tablas

El DRS deberá incluir listas de las figuras y tablas que aparezcan en el mismo. Dichas listas serán dos índices que indicarán el número, la descripción y la página en que aparece cada figura o tabla del DRS.

### B.3.1.5 Introducción

Esta sección debe contener una descripción breve de las principales características del sistema software que se va a desarrollar, la situación actual que genera la necesidad del nuevo desarrollo, la problemática que se acomete, y cualquier otra consideración que sitúe al posible lector en el contexto oportuno para comprender el resto del documento.

### B.3.1.6 Participantes en el proyecto

Esta sección debe contener una lista con todos los participantes en el proyecto, tanto desarrolladores como clientes y usuarios. Para cada participante se deberá indicar su nombre, el papel que desempeña en el proyecto, la organización a la que pertenece y cualquier otra información adicional que se considere oportuna.

#### **B.3.1.7 Descripción del sistema actual**

Esta sección debe contener una descripción del sistema actual en el caso de que se haya acometido su estudio. Para describir el sistema actual puede utilizarse cualquier técnica que se considere oportuno, por ejemplo las descritas en [Laguna *et al.* 1999] (*Diagrama Documentos–Tarea*, DDT) o en [García *et al.* 2000] (*Diagramas de Actividad*, también descritos en [Booch *et al.* 1999]).

#### **B.3.1.8 Objetivos del sistema**

Esta sección debe contener una lista con los objetivos que se esperan alcanzar cuando el sistema software a desarrollar esté en explotación, especificados mediante la plantilla para objetivos.

#### **B.3.1.9 Catálogo de requisitos del sistema**

Esta sección se divide en las siguientes subsecciones en las que se describen los requisitos del sistema. Cada uno de los grandes grupos de requisitos, de almacenamiento de información, funcionales y no funcionales, podrán dividirse para ayudar a la legibilidad del documento, por ejemplo dividiendo cada subsección en requisitos asociados a un determinado objetivo, requisitos con características comunes, etc.

#### **B.3.1.10 Requisitos de almacenamiento de información**

Esta subsección debe contener la lista de requisitos de almacenamiento de información que se hayan identificado, utilizando para especificarlos la plantilla para requisitos de almacenamiento de información.

#### **B.3.1.11 Requisitos funcionales**

Esta subsección debe contener la lista de requisitos funcionales que se hayan identificado, dividiéndose en los siguientes apartados que se describen a continuación.

**B.3.1.12 Diagrama de casos de uso**

Este apartado debe contener los diagramas de casos de uso del sistema que se hayan realizado.

**B.3.1.13 Definición de los actores**

Este apartado debe contener una lista con los actores que se hayan identificado, especificados mediante la plantilla para actores de casos de uso.

**B.3.1.14 Casos de uso del sistema**

Este apartado debe contener los casos de uso que se hayan identificado, especificados mediante la plantilla para requisitos funcionales.

**B.3.1.15 Requisitos no funcionales**

Esta subsección debe contener la lista los requisitos no funcionales del sistema que se hayan identificado, especificados mediante la plantilla para requisitos no funcionales.

**B.3.1.16 Matriz de rastreabilidad objetivos/requisitos**

Esta sección debe contener una matriz *objetivo-requisito*, de forma que para cada objetivo se pueda conocer con qué requisitos está asociado. El formato de la matriz de rastreabilidad puede verse en la figura B.4.

|        | OBJ-01 | OBJ-02 | ... | OBJ- <i>n</i> |
|--------|--------|--------|-----|---------------|
| RI-01  | •      | •      |     |               |
| RI-02  |        | •      |     |               |
| ...    |        |        |     |               |
| RF-01  | •      |        |     |               |
| RF-02  | •      | •      |     |               |
| ...    |        |        |     |               |
| RNF-01 |        |        |     | •             |
| RNF-02 |        | •      |     |               |
| ...    |        |        |     |               |

Figura B.4: Matriz de rastreabilidad del Documento de Requisitos del Sistema



### **B.3.1.17 Conflictos pendientes de resolución**

Esta sección, que se incluirá en el caso de que no se opte por registrar los conflictos en un documento aparte, deberá contener los conflictos identificados durante el proceso y que aún están pendientes de resolución, descritos mediante la plantilla para conflictos.

### **B.3.1.18 Glosario de términos**

Esta sección, que se incluirá si se considera oportuno, deberá contener una lista ordenada alfabéticamente de los términos específicos del dominio del problema, acrónimos y abreviaturas que aparezcan en el documento y que se considere que su significado deba ser aclarado. Cada término deberá acompañarse de su significado.

### **B.3.1.19 Apéndices**

Los apéndices se usarán para proporcionar información adicional a la documentación obligatoria del documento. Sólo deben aparecer si se consideran oportunos y se identificarán con letras ordenadas alfabéticamente: A, B, C, etc.

## **B.4 Técnicas**

A continuación, se describen algunas de las técnicas que se proponen en esta metodología para obtener los productos de las tareas que se han descrito.

### **B.4.1 Entrevistas**

**Nota:** Esta sección coincide con la sección 3.4.1, pág. 63, por lo que se ha omitido para evitar repeticiones innecesarias.

### **B.4.2 Joint Application Development**

**Nota:** Esta sección coincide con la sección 3.4.2, pág. 66, por lo que se ha omitido para evitar repeticiones innecesarias.

### B.4.3 Brainstorming

**Nota:** Esta sección coincide con la sección 3.4.3, pág. 70, por lo que se ha omitido para evitar repeticiones innecesarias.

### B.4.4 Casos de uso

**Nota:** Esta sección coincide con la sección 3.4.4, pág. 72, por lo que se ha omitido para evitar repeticiones innecesarias.

### B.4.5 Plantillas y patrones de objetivos y requisitos

**Nota:** Esta sección coincide con la sección 3.6, pág. 84, por lo que se ha omitido para evitar repeticiones innecesarias.

## B.5 Ejemplo: gestión de un vídeo-club

En este apéndice se ofrecen algunos ejemplos de aplicación de las técnicas propuestas en esta metodología suponiendo el caso de la gestión de pequeño vídeo-club.

Dado que se trata de un ejemplo ficticio se han simplificado las plantillas eliminando los campos relativos a versión, autores, fuentes, importancia, urgencia y estado de desarrollo.

El ejemplo no es una especificación de requisitos completa, se incluye sólo a modo de ejemplo.

### B.5.1 Objetivos del sistema

|                    |                                                                                                                                 |
|--------------------|---------------------------------------------------------------------------------------------------------------------------------|
| <b>OBJ-01</b>      | <b>Gestionar las cintas y películas</b>                                                                                         |
| <b>Descripción</b> | El sistema deberá gestionar las cintas y películas disponibles en el vídeo club: adquisiciones, retiradas, disponibilidad, etc. |
| <b>Estabilidad</b> | alta                                                                                                                            |
| <b>Comentarios</b> | ninguno                                                                                                                         |

|                    |                                                                                                                                              |
|--------------------|----------------------------------------------------------------------------------------------------------------------------------------------|
| <b>OBJ-02</b>      | <b>Gestionar los socios</b>                                                                                                                  |
| <b>Descripción</b> | El sistema deberá gestionar las socios del vídeo-club: altas, bajas, modificaciones de datos, sanciones, personas autorizadas, cuentas, etc. |
| <b>Estabilidad</b> | alta                                                                                                                                         |
| <b>Comentarios</b> | ninguno                                                                                                                                      |

|                    |                                                                                                                                         |
|--------------------|-----------------------------------------------------------------------------------------------------------------------------------------|
| <b>OBJ-02</b>      | <b>Gestionar los alquileres</b>                                                                                                         |
| <b>Descripción</b> | El sistema deberá gestionar los alquileres de cintas: entregas, devoluciones, devoluciones tardías, reclamaciones, disponibilidad, etc. |
| <b>Estabilidad</b> | alta                                                                                                                                    |
| <b>Comentarios</b> | ninguno                                                                                                                                 |

## B.5.2 Requisitos de almacenamiento de información

|                             |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |
|-----------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>RI-01</b>                | <b>Información sobre películas</b>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          |
| <b>Objetivos asociados</b>  | <ul style="list-style-type: none"> <li>• OBJ-01 <i>Gestionar las películas y cintas</i></li> </ul>                                                                                                                                                                                                                                                                                                                                                                                                                          |
| <b>Requisitos asociados</b> | <ul style="list-style-type: none"> <li>• RF-04 <i>Alta de película</i></li> <li>• RF-05 <i>Alta de cinta de vídeo</i></li> <li>• RF-08 <i>Baja de cinta de vídeo</i></li> <li>• RF-10 <i>Consulta de película</i></li> <li>• RF-13 <i>Consulta de películas alquiladas un día determinado</i></li> </ul>                                                                                                                                                                                                                    |
| <b>Descripción</b>          | El sistema deberá almacenar la información correspondiente a las películas del vídeo-club. En concreto:                                                                                                                                                                                                                                                                                                                                                                                                                     |
| <b>Datos específicos</b>    | <ul style="list-style-type: none"> <li>• Título de la película</li> <li>• Cintas de la película alquiladas en cada momento</li> <li>• Cintas de la película disponibles para ser alquiladas en cada momento</li> <li>• Tipo de la película: infantil, acción, ciencia-ficción o adultos</li> <li>• Duración de la película, en horas y minutos</li> <li>• Actores principales de la película</li> <li>• Director de la película</li> <li>• Productora de la película</li> <li>• Año de producción de la película</li> </ul> |
| <b>Intervalo temporal</b>   | pasado y presente                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |
| <b>Estabilidad</b>          | alta                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        |
| <b>Comentarios</b>          | ninguno                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |

|                             |                                                                                                                                                                                                                                                                                                                                                                                                          |
|-----------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>RI-02</b>                | <b>Información sobre socios</b>                                                                                                                                                                                                                                                                                                                                                                          |
| <b>Objetivos asociados</b>  | <ul style="list-style-type: none"> <li>• OBJ-02 <i>Gestionar los socios</i></li> </ul>                                                                                                                                                                                                                                                                                                                   |
| <b>Requisitos asociados</b> | <ul style="list-style-type: none"> <li>• RF-01 <i>Alta de socio</i></li> <li>• RF-02 <i>Baja de socio</i></li> <li>• RF-03 <i>Modificación de datos de un socio</i></li> <li>• RF-11 <i>Consulta de un socio</i></li> <li>• RF-12 <i>Consulta de socios con pagos pendientes</i></li> <li>• RF-12 <i>Consulta de los socios más rentables</i></li> <li>• RF-15 <i>Identificación de socio</i></li> </ul> |
| <b>Descripción</b>          | El sistema deberá almacenar la información correspondiente a los socios del vídeo-club. En concreto:                                                                                                                                                                                                                                                                                                     |
| <b>Datos específicos</b>    | <ul style="list-style-type: none"> <li>• Número de socio, que deberá ser único para cada socio</li> <li>• Número del documento nacional de identidad</li> <li>• Nombre y apellidos</li> <li>• Fecha de nacimiento</li> <li>• Sexo</li> <li>• Fecha de alta como socio</li> <li>• Dirección</li> <li>• Teléfonos</li> <li>• Películas alquiladas en un momento dado</li> </ul>                            |
| <b>Intervalo temporal</b>   | sólo presente                                                                                                                                                                                                                                                                                                                                                                                            |
| <b>Estabilidad</b>          | alta                                                                                                                                                                                                                                                                                                                                                                                                     |
| <b>Comentarios</b>          | ninguno                                                                                                                                                                                                                                                                                                                                                                                                  |

|                             |                                                                                                                                                                                                                                                                                                                                                                                                                                             |
|-----------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>RI-03</b>                | <b>Información sobre cuentas de socios</b>                                                                                                                                                                                                                                                                                                                                                                                                  |
| <b>Objetivos asociados</b>  | <ul style="list-style-type: none"> <li>• OBJ-02 <i>Gestionar los socios</i></li> </ul>                                                                                                                                                                                                                                                                                                                                                      |
| <b>Requisitos asociados</b> | <ul style="list-style-type: none"> <li>• RF-01 <i>Alta de socio</i></li> <li>• RF-02 <i>Baja de socio</i></li> <li>• RF-05 <i>Alquiler de cinta de vídeo</i></li> <li>• RF-08 <i>Devolución de cintas de vídeo</i></li> <li>• RF-09 <i>Ingreso a cuenta</i></li> <li>• RF-11 <i>Consulta de un socio</i></li> <li>• RF-12 <i>Consulta de socios con pagos pendientes</i></li> </ul>                                                         |
| <b>Descripción</b>          | El sistema deberá almacenar la información correspondiente a las cuentas de los socios del vídeo-club. En concreto:                                                                                                                                                                                                                                                                                                                         |
| <b>Datos específicos</b>    | <ul style="list-style-type: none"> <li>• Saldo de la cuenta en cada momento</li> <li>• Ingresos realizados en la cuenta, indicando fecha y cantidad</li> <li>• Cargos realizados en la cuenta, indicando fecha, motivo y cantidad</li> <li>• Pagos pendientes, indicando motivo que podrá ser alquiler no pagado o multa; en el caso de alquiler no pagado se debe indicar también la película alquilada y la fecha del alquiler</li> </ul> |
| <b>Intervalo temporal</b>   | sólo presente                                                                                                                                                                                                                                                                                                                                                                                                                               |
| <b>Estabilidad</b>          | alta                                                                                                                                                                                                                                                                                                                                                                                                                                        |
| <b>Comentarios</b>          | Un socio puede hacer ingresos a cuenta, por ejemplo para enviar a sus hijos por películas sin que éstos tengan que llevar dinero                                                                                                                                                                                                                                                                                                            |

## B.5.3 Requisitos funcionales

### B.5.3.1 Diagramas de casos de uso

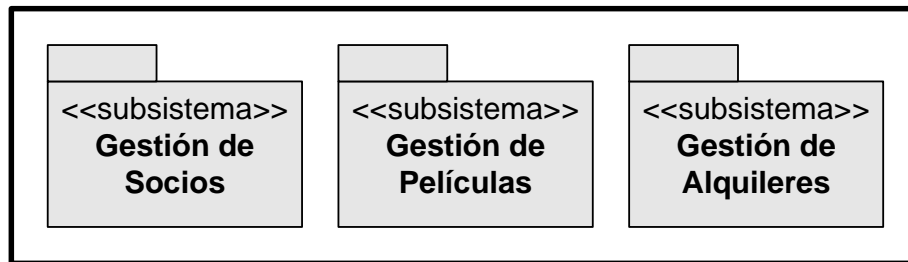


Figura B.5: Diagrama de subsistemas

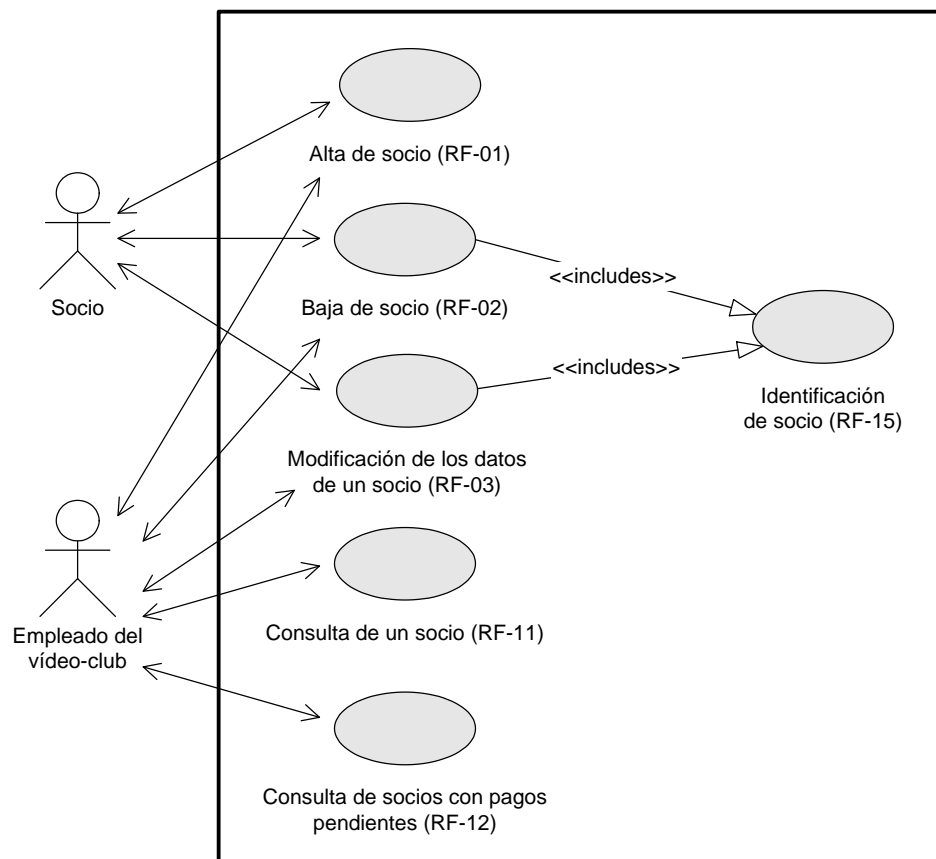


Figura B.6: Diagrama de casos de uso del subsistema Gestión de socios

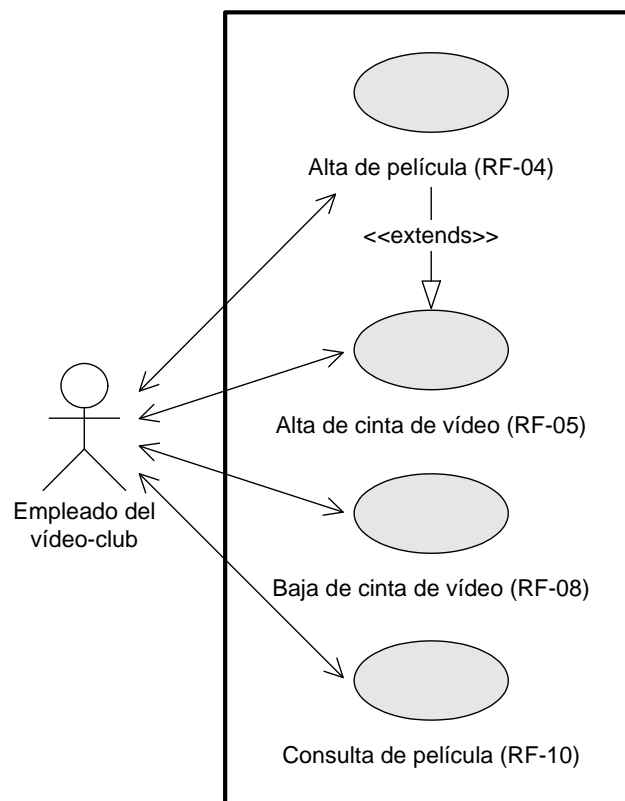


Figura B.7: Diagrama de casos de uso del subsistema Gestión de películas

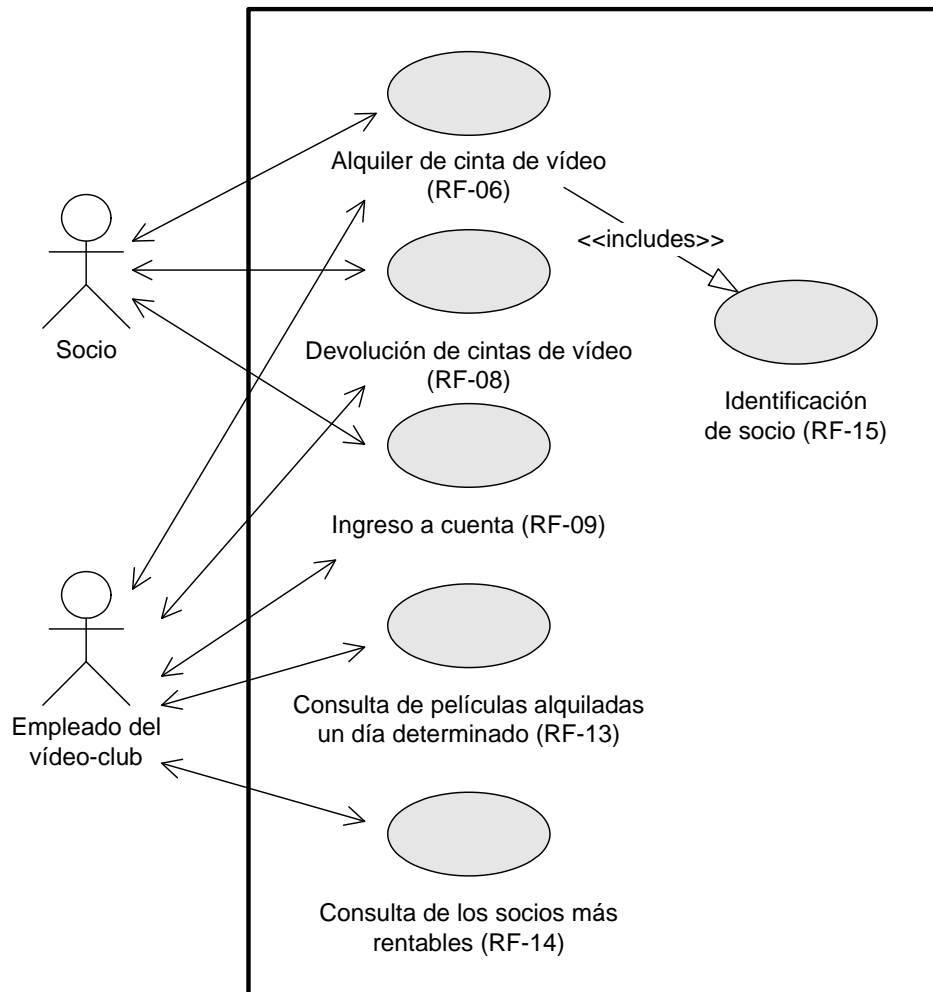


Figura B.8: Diagrama de casos de uso del subsistema Gestión de alquileres

**B.5.3.2 Definición de actores**

|                    |                                                   |
|--------------------|---------------------------------------------------|
| <b>ACT-01</b>      | <b>Socio</b>                                      |
| <b>Descripción</b> | Este actor representa a los socios del vídeo-club |
| <b>Comentarios</b> | ninguno                                           |

|                    |                                                      |
|--------------------|------------------------------------------------------|
| <b>ACT-02</b>      | <b>Empleado del vídeo-club</b>                       |
| <b>Descripción</b> | Este actor representa a los empleados del vídeo-club |
| <b>Comentarios</b> | ninguno                                              |

**B.5.3.3 Casos de uso del sistema**

(ver páginas siguientes)



**Casos de uso del subsistema Gestión de socios**

|                             |                                                                                                                              |                                                                                                                                                                                                                               |
|-----------------------------|------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>RF-01</b>                | <b>Alta de socio</b>                                                                                                         |                                                                                                                                                                                                                               |
| <b>Objetivos asociados</b>  | • OBJ-02 <i>Gestionar las socios</i>                                                                                         |                                                                                                                                                                                                                               |
| <b>Requisitos asociados</b> | • RI-02 <i>Información sobre socios</i>                                                                                      |                                                                                                                                                                                                                               |
| <b>Descripción</b>          | El sistema deberá comportarse tal como se describe en el siguiente caso de uso cuando alguien solicite su ingreso como socio |                                                                                                                                                                                                                               |
| <b>Precondición</b>         | El solicitante no es un socio del vídeo-club y tiene su documentación disponible                                             |                                                                                                                                                                                                                               |
| <b>Secuencia normal</b>     | <b>Paso</b>                                                                                                                  | <b>Acción</b>                                                                                                                                                                                                                 |
|                             | 1                                                                                                                            | El empleado del vídeo-club solicita al sistema comenzar el proceso de alta de un nuevo socio                                                                                                                                  |
|                             | 2                                                                                                                            | El sistema solicita los siguientes datos del nuevo socio: nº del DNI, nombre, apellidos, fecha de nacimiento, sexo, dirección y teléfonos de contacto                                                                         |
|                             | 3                                                                                                                            | El empleado del vídeo-club solicita los datos requeridos y la documentación al nuevo socio                                                                                                                                    |
|                             | 4                                                                                                                            | El empleado del vídeo-club comprueba que los datos del nuevo socio coinciden con los de la documentación aportada                                                                                                             |
|                             | 5                                                                                                                            | El empleado del vídeo-club proporciona los datos requeridos y solicita al sistema que los almacene                                                                                                                            |
|                             | 6                                                                                                                            | El sistema almacena los datos proporcionados, imprime el carnet de socio e informa al empleado del vídeo-club de que el proceso ha terminado con éxito                                                                        |
|                             | 7                                                                                                                            | El empleado del vídeo-club entrega el carnet al nuevo socio                                                                                                                                                                   |
| <b>Postcondición</b>        | El solicitante es socio del vídeo-club y el saldo de su cuenta es 0                                                          |                                                                                                                                                                                                                               |
| <b>Excepciones</b>          | <b>Paso</b>                                                                                                                  | <b>Acción</b>                                                                                                                                                                                                                 |
|                             | 4                                                                                                                            | Si la documentación aportada no es correcta, el empleado del vídeo-club cancela la operación, a continuación este caso de uso termina                                                                                         |
|                             | 5                                                                                                                            | Si el sistema detecta que el nuevo socio ya es socio del vídeo-club, el sistema informa de la situación al empleado del vídeo-club permitiéndole modificar los datos proporcionados, a continuación este caso de uso continúa |
|                             | 5                                                                                                                            | Si el empleado del vídeo-club solicita cancelar la operación, el sistema cancela la operación, a continuación este caso de uso termina                                                                                        |
| <b>Rendimiento</b>          | <b>Paso</b>                                                                                                                  | <b>Cota de tiempo</b>                                                                                                                                                                                                         |
|                             | 4                                                                                                                            | 5 segundos                                                                                                                                                                                                                    |
| <b>Frecuencia esperada</b>  | 10 veces/día                                                                                                                 |                                                                                                                                                                                                                               |
| <b>Estabilidad</b>          | alta                                                                                                                         |                                                                                                                                                                                                                               |
| <b>Comentarios</b>          | La frecuencia será mucho mayor durante los dos primeros meses, probablemente 100 veces/día                                   |                                                                                                                                                                                                                               |

|                             |                                                                                                                                          |                                                                                                                                                                            |
|-----------------------------|------------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>RF-02</b>                | <b>Baja de socio</b>                                                                                                                     |                                                                                                                                                                            |
| <b>Objetivos asociados</b>  | • OBJ-02 <i>Gestionar las socios</i>                                                                                                     |                                                                                                                                                                            |
| <b>Requisitos asociados</b> | • RI-02 <i>Información sobre socios</i>                                                                                                  |                                                                                                                                                                            |
| <b>Descripción</b>          | El sistema deberá comportarse tal como se describe en el siguiente caso de uso cuando un socio solicite su baja                          |                                                                                                                                                                            |
| <b>Precondición</b>         | El solicitante es un socio del vídeo-club y tiene su documentación disponible                                                            |                                                                                                                                                                            |
| <b>Secuencia normal</b>     | <b>Paso</b>                                                                                                                              | <b>Acción</b>                                                                                                                                                              |
|                             | 1                                                                                                                                        | El empleado del vídeo-club solicita al sistema comenzar el proceso de baja de un socio                                                                                     |
|                             | 2                                                                                                                                        | Se realiza el caso de uso RF-15 (Identificación de socio)                                                                                                                  |
|                             | 3                                                                                                                                        | El empleado del vídeo-club solicita al sistema que elimine la información correspondiente al socio                                                                         |
|                             | 4                                                                                                                                        | El sistema elimina los datos correspondientes al socio e informa al empleado del vídeo-club de que el proceso ha terminado con éxito                                       |
|                             | 4                                                                                                                                        | El empleado del vídeo-club inhabilita el carnet al socio que se acaba de dar de baja                                                                                       |
| <b>Postcondición</b>        | El solicitante no es socio del vídeo-club                                                                                                |                                                                                                                                                                            |
| <b>Excepciones</b>          | <b>Paso</b>                                                                                                                              | <b>Acción</b>                                                                                                                                                              |
|                             | 3                                                                                                                                        | Si el socio tiene pagos pendientes, el sistema el sistema comunica la situación al empleado del vídeo-club y cancela la operación, a continuación este caso de uso termina |
|                             | 3                                                                                                                                        | Si el empleado del vídeo-club solicita cancelar la operación, el sistema el sistema cancela la operación, a continuación este caso de uso termina                          |
| <b>Rendimiento</b>          | <b>Paso</b>                                                                                                                              | <b>Cota de tiempo</b>                                                                                                                                                      |
|                             | 6                                                                                                                                        | 1 segundo                                                                                                                                                                  |
| <b>Frecuencia esperada</b>  | 1 vez/mes                                                                                                                                |                                                                                                                                                                            |
| <b>Estabilidad</b>          | alta                                                                                                                                     |                                                                                                                                                                            |
| <b>Comentarios</b>          | Si el socio que desea darse de baja tiene un pago pendiente, puede hacer un ingreso por su importe y repetir el proceso de darse de baja |                                                                                                                                                                            |

|                             |                                                                                                                                      |                                                                                                                                                                            |
|-----------------------------|--------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>RF-03</b>                | <b>Modificación de los datos de un socio</b>                                                                                         |                                                                                                                                                                            |
| <b>Objetivos asociados</b>  | • OBJ-02 <i>Gestionar las socios</i>                                                                                                 |                                                                                                                                                                            |
| <b>Requisitos asociados</b> | • RI-02 <i>Información sobre socios</i>                                                                                              |                                                                                                                                                                            |
| <b>Descripción</b>          | El sistema deberá comportarse tal como se describe en el siguiente caso de uso cuando un socio solicite la modificación de sus datos |                                                                                                                                                                            |
| <b>Precondición</b>         | El solicitante es un socio del vídeo-club y tiene su documentación disponible                                                        |                                                                                                                                                                            |
| <b>Secuencia normal</b>     | <b>Paso</b>                                                                                                                          | <b>Acción</b>                                                                                                                                                              |
|                             | 1                                                                                                                                    | El empleado del vídeo-club solicita al sistema comenzar el proceso de modificación de los datos de un de un socio                                                          |
|                             | 2                                                                                                                                    | Se realiza el caso de uso RF-15 (Identificación de socio)                                                                                                                  |
|                             | 3                                                                                                                                    | El sistema muestra los siguientes datos correspondientes al socio a modificar: n° del DNI, nombre, apellidos, fecha de nacimiento, sexo, dirección y teléfonos de contacto |
|                             | 4                                                                                                                                    | El sistema permite al empleado del vídeo-club modificar los siguientes datos: dirección y teléfonos de contacto                                                            |
|                             | 5                                                                                                                                    | El empleado del vídeo-club modifica los datos que el sistema le permite y solicita al sistema que los almacene                                                             |
|                             | 6                                                                                                                                    | El sistema modifica los datos correspondientes al socio e informa al empleado del vídeo-club de que el proceso ha terminado con éxito                                      |
|                             | 7                                                                                                                                    | Si algún dato modificado aparece en el carnet de socio, el sistema imprime un nuevo carnet de socio                                                                        |
|                             | 8                                                                                                                                    | Si fue necesario imprimir un nuevo carnet de socio, el empleado del vídeo-club entrega el nuevo carnet al socio e inhabilita el antiguo                                    |
| <b>Postcondición</b>        | La información del socio está actualizada                                                                                            |                                                                                                                                                                            |
| <b>Excepciones</b>          | <b>Paso</b>                                                                                                                          | <b>Acción</b>                                                                                                                                                              |
|                             | 5                                                                                                                                    | Si el empleado del vídeo-club solicita cancelar la operación, el sistema el sistema cancela la operación, a continuación este caso de uso termina                          |
| <b>Rendimiento</b>          | <b>Paso</b>                                                                                                                          | <b>Cota de tiempo</b>                                                                                                                                                      |
|                             | 6                                                                                                                                    | 1 segundo                                                                                                                                                                  |
| <b>Frecuencia esperada</b>  | 1 vez/mes                                                                                                                            |                                                                                                                                                                            |
| <b>Comentarios</b>          | ninguno                                                                                                                              |                                                                                                                                                                            |

|                             |                                                                                                                                        |                                                                                                                                                                                          |
|-----------------------------|----------------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>RF-11</b>                | <b>Consulta de un socio</b>                                                                                                            |                                                                                                                                                                                          |
| <b>Objetivos asociados</b>  | • OBJ-02 <i>Gestionar las socios</i>                                                                                                   |                                                                                                                                                                                          |
| <b>Requisitos asociados</b> | • RI-02 <i>Información sobre socios</i>                                                                                                |                                                                                                                                                                                          |
| <b>Descripción</b>          | El sistema deberá comportarse tal como se describe en el siguiente caso de uso cuando el empleado del vídeo-club lo considere oportuno |                                                                                                                                                                                          |
| <b>Precondición</b>         | Ninguna                                                                                                                                |                                                                                                                                                                                          |
| <b>Secuencia normal</b>     | <b>Paso</b>                                                                                                                            | <b>Acción</b>                                                                                                                                                                            |
|                             | 1                                                                                                                                      | El empleado del vídeo-club solicita al sistema comenzar el proceso de consulta de los datos de un socio                                                                                  |
|                             | 2                                                                                                                                      | El sistema solicita que se identifique al socio                                                                                                                                          |
|                             | 3                                                                                                                                      | El empleado del vídeo-club proporciona los datos de identificación al sistema                                                                                                            |
|                             | 4                                                                                                                                      | El sistema muestra la siguiente información asociada al socio: nombre, apellidos, dirección, números de teléfono, alquileres pendientes y saldo de su cuenta                             |
|                             | 5                                                                                                                                      | Si el empleado del vídeo-club solicita la impresión de los datos, el sistema imprime los datos del socio                                                                                 |
| <b>Postcondición</b>        | Ninguna                                                                                                                                |                                                                                                                                                                                          |
| <b>Excepciones</b>          | <b>Paso</b>                                                                                                                            | <b>Acción</b>                                                                                                                                                                            |
|                             | 3                                                                                                                                      | Si el empleado del vídeo-club solicita cancelar la operación, el sistema cancela la operación, a continuación este caso de uso termina                                                   |
|                             | 5                                                                                                                                      | Si el sistema no tiene registrado ningún socio con la identificación proporcionada, el sistema comunica al empleado del vídeo-club la situación, a continuación este caso de uso termina |
| <b>Rendimiento</b>          | <b>Paso</b>                                                                                                                            | <b>Cota de tiempo</b>                                                                                                                                                                    |
|                             | 4                                                                                                                                      | 1 segundo                                                                                                                                                                                |
| <b>Frecuencia esperada</b>  | 5 veces/día                                                                                                                            |                                                                                                                                                                                          |
| <b>Comentarios</b>          | El formato de visualización de los datos está pendiente de definición                                                                  |                                                                                                                                                                                          |

|                             |                                                                                                                                        |                                                                                                                                                                                              |
|-----------------------------|----------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>RF-12</b>                | <b>Consulta de socios con pagos pendientes</b>                                                                                         |                                                                                                                                                                                              |
| <b>Objetivos asociados</b>  | • OBJ-02 <i>Gestionar las socios</i>                                                                                                   |                                                                                                                                                                                              |
| <b>Requisitos asociados</b> | • RI-02 <i>Información sobre socios</i><br>• RI-03 <i>Información sobre cuentas de socios</i>                                          |                                                                                                                                                                                              |
| <b>Descripción</b>          | El sistema deberá comportarse tal como se describe en el siguiente caso de uso cuando el empleado del vídeo-club lo considere oportuno |                                                                                                                                                                                              |
| <b>Precondición</b>         | Ninguna                                                                                                                                |                                                                                                                                                                                              |
| <b>Secuencia normal</b>     | <b>Paso</b>                                                                                                                            | <b>Acción</b>                                                                                                                                                                                |
|                             | 1                                                                                                                                      | El empleado del vídeo-club solicita al sistema comenzar el proceso de consulta de los socios con pagos pendientes                                                                            |
|                             | 2                                                                                                                                      | El sistema muestra una lista ordenada por cantidad pendiente con la siguiente información por cada socio: nombre, apellidos, cantidad total pendiente y detalle de las cantidades pendientes |
|                             | 3                                                                                                                                      | Si el empleado del vídeo-club solicita la impresión de los datos, el sistema imprime la lista                                                                                                |
| <b>Postcondición</b>        | Ninguna                                                                                                                                |                                                                                                                                                                                              |
| <b>Excepciones</b>          | <b>Paso</b>                                                                                                                            | <b>Acción</b>                                                                                                                                                                                |
|                             | –                                                                                                                                      | –                                                                                                                                                                                            |
| <b>Rendimiento</b>          | <b>Paso</b>                                                                                                                            | <b>Cota de tiempo</b>                                                                                                                                                                        |
|                             | 2                                                                                                                                      | 5 segundos                                                                                                                                                                                   |
| <b>Frecuencia esperada</b>  | 1 vez/semana                                                                                                                           |                                                                                                                                                                                              |
| <b>Comentarios</b>          | ninguno                                                                                                                                |                                                                                                                                                                                              |

|                             |                                                                                                                                                                                                                                                                                                                         |                                                                                                                                                                                                                                                                |
|-----------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>RF-15</b>                | <b>Identificación de socio</b>                                                                                                                                                                                                                                                                                          |                                                                                                                                                                                                                                                                |
| <b>Objetivos asociados</b>  | • OBJ-02 <i>Gestionar las socios</i>                                                                                                                                                                                                                                                                                    |                                                                                                                                                                                                                                                                |
| <b>Requisitos asociados</b> | • RI-02 <i>Información sobre socios</i>                                                                                                                                                                                                                                                                                 |                                                                                                                                                                                                                                                                |
| <b>Descripción</b>          | El sistema deberá comportarse tal como se describe en el siguiente caso de uso durante la realización de los casos de uso: <ul style="list-style-type: none"> <li>• RF-02 <i>Baja de socio</i></li> <li>• RF-03 <i>Modificación de datos de un socio</i></li> <li>• RF-06 <i>Alquiler de cintas de vídeo</i></li> </ul> |                                                                                                                                                                                                                                                                |
| <b>Precondición</b>         | El socio tiene su documentación disponible                                                                                                                                                                                                                                                                              |                                                                                                                                                                                                                                                                |
| <b>Secuencia normal</b>     | <b>Paso</b>                                                                                                                                                                                                                                                                                                             | <b>Acción</b>                                                                                                                                                                                                                                                  |
|                             | 1                                                                                                                                                                                                                                                                                                                       | El sistema solicita que se identifique al socio                                                                                                                                                                                                                |
|                             | 2                                                                                                                                                                                                                                                                                                                       | El empleado del vídeo-club solicita el carnet de socio                                                                                                                                                                                                         |
|                             | 3                                                                                                                                                                                                                                                                                                                       | El empleado del vídeo-club proporciona los datos de identificación al sistema                                                                                                                                                                                  |
|                             | 4                                                                                                                                                                                                                                                                                                                       | El sistema muestra los números de teléfonos que el socio proporcionó cuando se dio de alta                                                                                                                                                                     |
|                             | 5                                                                                                                                                                                                                                                                                                                       | El empleado del vídeo-club solicita al socio que le confirme alguno de los números de teléfono registrados en el sistema                                                                                                                                       |
|                             | 6                                                                                                                                                                                                                                                                                                                       | El empleado del vídeo-club confirma la identidad del socio al sistema                                                                                                                                                                                          |
| <b>Postcondición</b>        | Ninguna                                                                                                                                                                                                                                                                                                                 |                                                                                                                                                                                                                                                                |
| <b>Excepciones</b>          | <b>Paso</b>                                                                                                                                                                                                                                                                                                             | <b>Acción</b>                                                                                                                                                                                                                                                  |
|                             | 3                                                                                                                                                                                                                                                                                                                       | Si el sistema detecta que el supuesto socio no es socio del vídeo-club, el sistema comunica al empleado del vídeo-club la situación, a continuación este caso de uso aborta                                                                                    |
|                             | 5                                                                                                                                                                                                                                                                                                                       | Si el socio no conoce ningún número de teléfono registrado en el sistema y no puede demostrar su identidad, el empleado del vídeo-club retiene el carnet de socio y cancela la operación, a continuación este caso de uso aborta                               |
|                             | 5                                                                                                                                                                                                                                                                                                                       | Si el socio no conoce ningún número de teléfono registrado pero puede demostrar su identidad por otros medios, el empleado del vídeo-club le recuerda los números de teléfonos que proporcionó cuando se dio de alta, a continuación este caso de uso continúa |
| <b>Rendimiento</b>          | <b>Paso</b>                                                                                                                                                                                                                                                                                                             | <b>Cota de tiempo</b>                                                                                                                                                                                                                                          |
|                             | –                                                                                                                                                                                                                                                                                                                       | –                                                                                                                                                                                                                                                              |
| <b>Frecuencia esperada</b>  | 50 veces/día                                                                                                                                                                                                                                                                                                            |                                                                                                                                                                                                                                                                |
| <b>Comentarios</b>          | ninguno                                                                                                                                                                                                                                                                                                                 |                                                                                                                                                                                                                                                                |

**Casos de uso del subsistema Gestión de películas**

|                             |                                                                                                                                     |                                                                                                                                                                                                                    |
|-----------------------------|-------------------------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>RF-04</b>                | <b>Alta de película</b>                                                                                                             |                                                                                                                                                                                                                    |
| <b>Objetivos asociados</b>  | • OBJ-01 <i>Gestionar las cintas y películas</i>                                                                                    |                                                                                                                                                                                                                    |
| <b>Requisitos asociados</b> | • RI-01 <i>Información sobre películas</i><br>• RF-05 <i>Alta de cinta de vídeo</i>                                                 |                                                                                                                                                                                                                    |
| <b>Descripción</b>          | El sistema deberá comportarse tal como se describe en el siguiente caso de uso cuando se adquiriera una cinta de una película nueva |                                                                                                                                                                                                                    |
| <b>Precondición</b>         | La película no está registrada en el sistema                                                                                        |                                                                                                                                                                                                                    |
| <b>Secuencia normal</b>     | <b>Paso</b>                                                                                                                         | <b>Acción</b>                                                                                                                                                                                                      |
|                             | 1                                                                                                                                   | El empleado del vídeo-club solicita al sistema comenzar el proceso de alta de película                                                                                                                             |
|                             | 2                                                                                                                                   | El sistema solicita los siguientes datos de la nueva película: título, tipo de película, duración, actores principales, director, productora y año de producción                                                   |
|                             | 3                                                                                                                                   | El empleado del vídeo-club proporciona los datos requeridos y solicita al sistema que los almacene                                                                                                                 |
|                             | 4                                                                                                                                   | El sistema almacena los datos proporcionados e informa al empleado del vídeo-club de que el proceso ha terminado con éxito                                                                                         |
| <b>Postcondición</b>        | El sistema ha almacenado la información correspondiente a la nueva película                                                         |                                                                                                                                                                                                                    |
| <b>Excepciones</b>          | <b>Paso</b>                                                                                                                         | <b>Acción</b>                                                                                                                                                                                                      |
|                             | 4                                                                                                                                   | Si el sistema detecta que la película ya está registrada, el sistema informa de la situación al empleado del vídeo-club permitiéndole modificar los datos proporcionados, a continuación este caso de uso continúa |
|                             | 4                                                                                                                                   | Si el empleado del vídeo-club solicita cancelar la operación, el sistema cancela la operación, a continuación este caso de uso se cancela                                                                          |
| <b>Rendimiento</b>          | <b>Paso</b>                                                                                                                         | <b>Cota de tiempo</b>                                                                                                                                                                                              |
|                             | 4                                                                                                                                   | 1 segundo                                                                                                                                                                                                          |
| <b>Frecuencia esperada</b>  | 1 vez/día                                                                                                                           |                                                                                                                                                                                                                    |
| <b>Comentarios</b>          | ninguno                                                                                                                             |                                                                                                                                                                                                                    |

|                             |                                                                                                                                  |                                                                                                                                                                                            |
|-----------------------------|----------------------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>RF-05</b>                | <b>Alta de cinta de vídeo</b>                                                                                                    |                                                                                                                                                                                            |
| <b>Objetivos asociados</b>  | • OBJ-01 <i>Gestionar las cintas y películas</i>                                                                                 |                                                                                                                                                                                            |
| <b>Requisitos asociados</b> | • RI-01 <i>Información sobre películas</i>                                                                                       |                                                                                                                                                                                            |
| <b>Descripción</b>          | El sistema deberá comportarse tal como se describe en el siguiente caso de uso cuando se adquieran nuevas cintas de una película |                                                                                                                                                                                            |
| <b>Precondición</b>         | Ninguna                                                                                                                          |                                                                                                                                                                                            |
| <b>Secuencia normal</b>     | <b>Paso</b>                                                                                                                      | <b>Acción</b>                                                                                                                                                                              |
|                             | 1                                                                                                                                | El empleado del vídeo-club solicita al sistema comenzar el proceso de alta de cinta                                                                                                        |
|                             | 2                                                                                                                                | El sistema solicita que se identifique la película que contiene la cinta                                                                                                                   |
|                             | 3                                                                                                                                | El empleado del vídeo-club identifica la película                                                                                                                                          |
|                             | 4                                                                                                                                | Si la película no está registrada, se realiza el caso de uso RF-04 (Alta de película)                                                                                                      |
|                             | 5                                                                                                                                | El sistema solicita el número de cintas de la película a dar de alta                                                                                                                       |
|                             | 6                                                                                                                                | El empleado del vídeo-club proporciona el número de cintas y solicita al sistema que almacene la información                                                                               |
|                             | 7                                                                                                                                | El sistema almacena los datos proporcionados, imprime la etiquetas de identificación de cintas autoadhesivas e informa al empleado del vídeo-club de que el proceso ha terminado con éxito |
|                             | 8                                                                                                                                | El empleado del vídeo-club pega las etiquetas en las cintas y las coloca en las estanterías                                                                                                |
| <b>Postcondición</b>        | Las cintas están registradas en el sistema                                                                                       |                                                                                                                                                                                            |
| <b>Excepciones</b>          | <b>Paso</b>                                                                                                                      | <b>Acción</b>                                                                                                                                                                              |
|                             | 6                                                                                                                                | Si el empleado del vídeo-club solicita cancelar la operación, el sistema cancela la operación, a continuación este caso de uso termina                                                     |
| <b>Rendimiento</b>          | <b>Paso</b>                                                                                                                      | <b>Cota de tiempo</b>                                                                                                                                                                      |
|                             | 7                                                                                                                                | 1 segundo                                                                                                                                                                                  |
| <b>Frecuencia esperada</b>  | 1 vez/día                                                                                                                        |                                                                                                                                                                                            |
| <b>Comentarios</b>          | ninguno                                                                                                                          |                                                                                                                                                                                            |



|                             |                                                                                                                                        |                                                                                                                                                                                           |
|-----------------------------|----------------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>RF-08</b>                | <b>Baja de cinta de vídeo</b>                                                                                                          |                                                                                                                                                                                           |
| <b>Objetivos asociados</b>  | • OBJ-01 <i>Gestionar las cintas y películas</i>                                                                                       |                                                                                                                                                                                           |
| <b>Requisitos asociados</b> | • RI-01 <i>Información sobre películas</i>                                                                                             |                                                                                                                                                                                           |
| <b>Descripción</b>          | El sistema deberá comportarse tal como se describe en el siguiente caso de uso cuando el empleado del vídeo-club lo considere oportuno |                                                                                                                                                                                           |
| <b>Precondición</b>         | La cinta está registrada en el sistema                                                                                                 |                                                                                                                                                                                           |
| <b>Secuencia normal</b>     | <b>Paso</b>                                                                                                                            | <b>Acción</b>                                                                                                                                                                             |
|                             | 1                                                                                                                                      | El empleado del vídeo-club solicita al sistema comenzar el proceso de baja de cinta de vídeo                                                                                              |
|                             | 2                                                                                                                                      | El sistema solicita que se identifique la cinta a dar de baja                                                                                                                             |
|                             | 3                                                                                                                                      | El empleado del vídeo-club identifica la cinta a eliminar y solicita al sistema que la dé de baja                                                                                         |
|                             | 4                                                                                                                                      | El sistema registra la baja de la cinta e informa al empleado del vídeo-club de que el proceso ha terminado con éxito                                                                     |
|                             | 5                                                                                                                                      | El empleado del vídeo-club elimina la cinta de las estanterías                                                                                                                            |
| <b>Postcondición</b>        | La cinta no está registrada en el sistema                                                                                              |                                                                                                                                                                                           |
| <b>Excepciones</b>          | <b>Paso</b>                                                                                                                            | <b>Acción</b>                                                                                                                                                                             |
|                             | 3                                                                                                                                      | Si el sistema no tiene registrada ninguna cinta con la identificación proporcionada, el sistema comunica al empleado del vídeo-club la situación, a continuación este caso de uso termina |
|                             | 3                                                                                                                                      | Si el empleado del vídeo-club solicita cancelar la operación, el sistema cancela la operación, a continuación este caso de uso termina                                                    |
| <b>Rendimiento</b>          | <b>Paso</b>                                                                                                                            | <b>Cota de tiempo</b>                                                                                                                                                                     |
|                             | 4                                                                                                                                      | 1 segundo                                                                                                                                                                                 |
| <b>Frecuencia esperada</b>  | 1 vez/mes                                                                                                                              |                                                                                                                                                                                           |
| <b>Comentarios</b>          | ninguno                                                                                                                                |                                                                                                                                                                                           |

|                             |                                                                                                                                        |                                                                                                                                                                                      |
|-----------------------------|----------------------------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>RF-10</b>                | <b>Consulta de una película</b>                                                                                                        |                                                                                                                                                                                      |
| <b>Objetivos asociados</b>  | • OBJ-01 <i>Gestionar las cintas y películas</i>                                                                                       |                                                                                                                                                                                      |
| <b>Requisitos asociados</b> | • RI-01 <i>Información sobre películas</i>                                                                                             |                                                                                                                                                                                      |
| <b>Descripción</b>          | El sistema deberá comportarse tal como se describe en el siguiente caso de uso cuando el empleado del vídeo-club lo considere oportuno |                                                                                                                                                                                      |
| <b>Precondición</b>         | Ninguna                                                                                                                                |                                                                                                                                                                                      |
| <b>Secuencia normal</b>     | <b>Paso</b>                                                                                                                            | <b>Acción</b>                                                                                                                                                                        |
|                             | 1                                                                                                                                      | El empleado del vídeo-club solicita al sistema comenzar el proceso de consulta de los datos de una película                                                                          |
|                             | 2                                                                                                                                      | El sistema solicita que se identifique la película a consultar                                                                                                                       |
|                             | 3                                                                                                                                      | El empleado del vídeo-club identifica la película a consultar                                                                                                                        |
|                             | 4                                                                                                                                      | El sistema muestra los siguientes datos correspondientes a la película: título, tema, año de producción, actores principales, nombre de la productora y número de cintas disponibles |
|                             | 5                                                                                                                                      | Si el empleado del vídeo-club solicita la impresión de los datos, el sistema imprime los datos de la película                                                                        |
| <b>Postcondición</b>        | La información correspondiente a la película consultada no ha cambiado                                                                 |                                                                                                                                                                                      |
| <b>Excepciones</b>          | <b>Paso</b>                                                                                                                            | <b>Acción</b>                                                                                                                                                                        |
|                             | 3                                                                                                                                      | Si el empleado del vídeo-club solicita cancelar la operación, el sistema cancela la operación, a continuación este caso de uso termina                                               |
| <b>Rendimiento</b>          | <b>Paso</b>                                                                                                                            | <b>Cota de tiempo</b>                                                                                                                                                                |
|                             | 4                                                                                                                                      | 1 segundo                                                                                                                                                                            |
| <b>Frecuencia esperada</b>  | 1 vez/día                                                                                                                              |                                                                                                                                                                                      |
| <b>Comentarios</b>          | ninguno                                                                                                                                |                                                                                                                                                                                      |

**Casos de uso del subsistema Gestión de alquileres**

|                             |                                                                                                                                            |                                                                                                                                                                                                |
|-----------------------------|--------------------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>RF-06</b>                | <b>Alquiler de cintas de vídeo</b>                                                                                                         |                                                                                                                                                                                                |
| <b>Objetivos asociados</b>  | • OBJ-03 <i>Gestionar los alquileres</i>                                                                                                   |                                                                                                                                                                                                |
| <b>Requisitos asociados</b> | • RI-02 <i>Información sobre socios</i><br>• RI-03 <i>Información sobre cuentas de socios</i>                                              |                                                                                                                                                                                                |
| <b>Descripción</b>          | El sistema deberá comportarse tal como se describe en el siguiente caso de uso cuando un socio solicite alquilar una o más cintas de vídeo |                                                                                                                                                                                                |
| <b>Precondición</b>         | Ninguna de las cintas a alquilar está registradas como alquiladas                                                                          |                                                                                                                                                                                                |
| <b>Secuencia normal</b>     | <b>Paso</b>                                                                                                                                | <b>Acción</b>                                                                                                                                                                                  |
|                             | 1                                                                                                                                          | El empleado del vídeo-club solicita al sistema comenzar el proceso de alquiler de cintas de vídeo                                                                                              |
|                             | 2                                                                                                                                          | Se realiza el caso de uso RF-15 (Identificación de socio)                                                                                                                                      |
|                             | 2                                                                                                                                          | El sistema solicita que se identifiquen las cintas que desean alquilar                                                                                                                         |
|                             | 3                                                                                                                                          | El empleado del vídeo-club identifica las cintas y solicita al sistema que registre el alquiler                                                                                                |
|                             | 4                                                                                                                                          | El sistema almacena la información de los alquileres y comunica al empleado del vídeo-club que el proceso de registro ha terminado con éxito                                                   |
|                             | 5                                                                                                                                          | Si el socio decide pagar al contado, el sistema imprime el ticket con el importe correspondiente y registra el pago como un ingreso en la cuenta del socio                                     |
|                             | 6                                                                                                                                          | Si el socio decide pagar a cuenta, el sistema registra el cargo en la cuenta del socio                                                                                                         |
| <b>Postcondición</b>        | Las cintas a alquilar están registradas como alquiladas y la cuenta del socio está actualizada                                             |                                                                                                                                                                                                |
| <b>Excepciones</b>          | <b>Paso</b>                                                                                                                                | <b>Acción</b>                                                                                                                                                                                  |
|                             | 3                                                                                                                                          | Si alguna de las cintas está registrada como alquilada, el sistema comunicar la situación al empleado del vídeo-club y excluir la cinta del alquiler, a continuación este caso de uso continúa |
|                             | 3                                                                                                                                          | Si el empleado del vídeo-club solicita cancelar la operación, el sistema cancela la operación, a continuación este caso de uso termina                                                         |
| <b>Rendimiento</b>          | <b>Paso</b>                                                                                                                                | <b>Cota de tiempo</b>                                                                                                                                                                          |
|                             | 4                                                                                                                                          | 1 segundo                                                                                                                                                                                      |
| <b>Frecuencia esperada</b>  | 50 veces/día                                                                                                                               |                                                                                                                                                                                                |
| <b>Comentarios</b>          | ninguno                                                                                                                                    |                                                                                                                                                                                                |

|                             |                                                                                                                                            |                                                                                                                                                                                                |
|-----------------------------|--------------------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>RF-07</b>                | <b>Devolución de cintas de vídeo</b>                                                                                                       |                                                                                                                                                                                                |
| <b>Objetivos asociados</b>  | • OBJ-03 <i>Gestionar los alquileres</i>                                                                                                   |                                                                                                                                                                                                |
| <b>Requisitos asociados</b> | • RI-02 <i>Información sobre socios</i><br>• RI-03 <i>Información sobre cuentas de socios</i>                                              |                                                                                                                                                                                                |
| <b>Descripción</b>          | El sistema deberá comportarse tal como se describe en el siguiente caso de uso cuando un socio solicite devolver una o más cintas de vídeo |                                                                                                                                                                                                |
| <b>Precondición</b>         | Todas las cintas a devolver están registradas como alquiladas                                                                              |                                                                                                                                                                                                |
| <b>Secuencia normal</b>     | <b>Paso</b>                                                                                                                                | <b>Acción</b>                                                                                                                                                                                  |
|                             | 1                                                                                                                                          | El empleado del vídeo-club solicita al sistema comenzar el proceso de devolución de cintas de vídeo                                                                                            |
|                             | 2                                                                                                                                          | El sistema solicita que se identifiquen las cintas que se desean devolver                                                                                                                      |
|                             | 3                                                                                                                                          | El empleado del vídeo-club identifica las cintas y solicita al sistema que registre su devolución                                                                                              |
|                             | 4                                                                                                                                          | El sistema registra los devoluciones                                                                                                                                                           |
|                             | 5                                                                                                                                          | Si alguna cinta ha sido devuelta fuera de plazo, el sistema registra la multa correspondiente como un cargo en la cuenta del socio                                                             |
|                             | 6                                                                                                                                          | Si el socio decide pagar al contado, el sistema imprime el ticket con el importe correspondiente y registra el pago como un ingreso en la cuenta del socio                                     |
|                             | 7                                                                                                                                          | Si el socio decide pagar a cuenta, el sistema registra el cargo en la cuenta del socio                                                                                                         |
| <b>Postcondición</b>        | Las cintas a alquilar están registradas como alquiladas y la cuenta del socio está actualizada                                             |                                                                                                                                                                                                |
| <b>Excepciones</b>          | <b>Paso</b>                                                                                                                                | <b>Acción</b>                                                                                                                                                                                  |
|                             | 4                                                                                                                                          | Si alguna de las cintas está registrada como alquilada, el sistema comunicar la situación al empleado del vídeo-club y excluir la cinta del alquiler, a continuación este caso de uso continúa |
| <b>Rendimiento</b>          | <b>Paso</b>                                                                                                                                | <b>Cota de tiempo</b>                                                                                                                                                                          |
|                             | 4                                                                                                                                          | 1 segundo                                                                                                                                                                                      |
| <b>Frecuencia esperada</b>  | 50 veces/día                                                                                                                               |                                                                                                                                                                                                |
| <b>Comentarios</b>          | ninguno                                                                                                                                    |                                                                                                                                                                                                |

|                             |                                                                                                                                       |                                                                                                                                        |
|-----------------------------|---------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------|
| <b>RF-09</b>                | <b>Ingreso a cuenta</b>                                                                                                               |                                                                                                                                        |
| <b>Objetivos asociados</b>  | • OBJ-03 <i>Gestionar los alquileres</i>                                                                                              |                                                                                                                                        |
| <b>Requisitos asociados</b> | • RI-02 <i>Información sobre socios</i><br>• RI-03 <i>Información sobre cuentas de socios</i>                                         |                                                                                                                                        |
| <b>Descripción</b>          | El sistema deberá comportarse tal como se describe en el siguiente caso de uso cuando un socio solicite hacer un ingreso en su cuenta |                                                                                                                                        |
| <b>Precondición</b>         | El socio tiene disponible su carnet                                                                                                   |                                                                                                                                        |
| <b>Secuencia normal</b>     | <b>Paso</b>                                                                                                                           | <b>Acción</b>                                                                                                                          |
|                             | 1                                                                                                                                     | El empleado del vídeo-club solicita al sistema comenzar el proceso de ingreso en cuenta                                                |
|                             | 2                                                                                                                                     | El sistema solicita que se identifique al socio y se indique la cantidad a ingresar                                                    |
|                             | 3                                                                                                                                     | El empleado del vídeo-club proporciona al sistema la identificación del socio y la cantidad a ingresar                                 |
|                             | 4                                                                                                                                     | El sistema registra el ingreso e informa del nuevo saldo                                                                               |
|                             | 3                                                                                                                                     | El empleado del vídeo-club comunica al socio su nuevo saldo                                                                            |
| <b>Postcondición</b>        | El saldo de la cuenta del socio está actualizado                                                                                      |                                                                                                                                        |
| <b>Excepciones</b>          | <b>Paso</b>                                                                                                                           | <b>Acción</b>                                                                                                                          |
|                             | 3                                                                                                                                     | Si el empleado del vídeo-club solicita cancelar la operación, el sistema cancela la operación, a continuación este caso de uso termina |
| <b>Rendimiento</b>          | <b>Paso</b>                                                                                                                           | <b>Cota de tiempo</b>                                                                                                                  |
|                             | 4                                                                                                                                     | 1 segundo                                                                                                                              |
| <b>Frecuencia esperada</b>  | 5 veces/día                                                                                                                           |                                                                                                                                        |
| <b>Comentarios</b>          | Mientras no se implemente se puede hacer que todos los pagos sean al contado                                                          |                                                                                                                                        |

|                             |                                                                                                                                        |                                                                                                                                                                          |
|-----------------------------|----------------------------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>RF-13</b>                | <b>Consulta de las películas alquiladas un día determinado</b>                                                                         |                                                                                                                                                                          |
| <b>Objetivos asociados</b>  | • OBJ-03 <i>Gestionar los alquileres</i>                                                                                               |                                                                                                                                                                          |
| <b>Requisitos asociados</b> | • RI-01 <i>Información sobre películas</i>                                                                                             |                                                                                                                                                                          |
| <b>Descripción</b>          | El sistema deberá comportarse tal como se describe en el siguiente caso de uso cuando el empleado del vídeo-club lo considere oportuno |                                                                                                                                                                          |
| <b>Precondición</b>         | Ninguna                                                                                                                                |                                                                                                                                                                          |
| <b>Secuencia normal</b>     | <b>Paso</b>                                                                                                                            | <b>Acción</b>                                                                                                                                                            |
|                             | 1                                                                                                                                      | El empleado del vídeo-club solicita al sistema comenzar el proceso de consulta de las películas alquiladas un día determinado                                            |
|                             | 2                                                                                                                                      | El sistema solicita la fecha del día que se quiere consultar, proponiendo la del día actual                                                                              |
|                             | 3                                                                                                                                      | El empleado del vídeo-club proporciona la fecha del día determinado al sistema                                                                                           |
|                             | 4                                                                                                                                      | El sistema muestra una lista ordenada por número de alquileres con la siguiente información: título y tema de cada película y número de alquileres en el día determinado |
|                             | 5                                                                                                                                      | Si el empleado del vídeo-club solicita la impresión de los datos, el sistema imprime la lista                                                                            |
| <b>Postcondición</b>        | La información sobre las películas no ha cambiado                                                                                      |                                                                                                                                                                          |
| <b>Excepciones</b>          | <b>Paso</b>                                                                                                                            | <b>Acción</b>                                                                                                                                                            |
|                             | 3                                                                                                                                      | Si el empleado del vídeo-club solicita cancelar la operación, el sistema cancela la operación, a continuación este caso de uso termina                                   |
| <b>Rendimiento</b>          | <b>Paso</b>                                                                                                                            | <b>Cota de tiempo</b>                                                                                                                                                    |
|                             | 4                                                                                                                                      | 5 segundos                                                                                                                                                               |
| <b>Frecuencia esperada</b>  | 1 vez/día                                                                                                                              |                                                                                                                                                                          |
| <b>Importancia</b>          | importante                                                                                                                             |                                                                                                                                                                          |
| <b>Urgencia</b>             | hay presión                                                                                                                            |                                                                                                                                                                          |
| <b>Comentarios</b>          | ninguno                                                                                                                                |                                                                                                                                                                          |

|                             |                                                                                                                                        |                                                                                                                                                                                                                 |
|-----------------------------|----------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>RF-14</b>                | <b>Consulta de los socios más rentables</b>                                                                                            |                                                                                                                                                                                                                 |
| <b>Objetivos asociados</b>  | • OBJ-03 <i>Gestionar los alquileres</i>                                                                                               |                                                                                                                                                                                                                 |
| <b>Requisitos asociados</b> | • RI-01 <i>Información sobre películas</i>                                                                                             |                                                                                                                                                                                                                 |
| <b>Descripción</b>          | El sistema deberá comportarse tal como se describe en el siguiente caso de uso cuando el empleado del vídeo-club lo considere oportuno |                                                                                                                                                                                                                 |
| <b>Precondición</b>         | Ninguna                                                                                                                                |                                                                                                                                                                                                                 |
| <b>Secuencia normal</b>     | <b>Paso</b>                                                                                                                            | <b>Acción</b>                                                                                                                                                                                                   |
|                             | 1                                                                                                                                      | El empleado del vídeo-club solicita al sistema comenzar el proceso de consulta de los socios más rentables                                                                                                      |
|                             | 2                                                                                                                                      | El sistema solicita el periodo de selección: última semana, último mes, último año o siempre                                                                                                                    |
|                             | 3                                                                                                                                      | El empleado del vídeo-club proporciona el periodo de selección al sistema                                                                                                                                       |
|                             | 4                                                                                                                                      | El sistema muestra una lista ordenada por cantidad de alquileres realizados con la siguiente información: número de socio, nombre, apellidos, teléfono y número de alquileres realizados en el periodo indicado |
|                             | 5                                                                                                                                      | Si el empleado del vídeo-club solicita la impresión de los datos, el sistema imprime la lista                                                                                                                   |
| <b>Postcondición</b>        | La información sobre los socios no ha cambiado                                                                                         |                                                                                                                                                                                                                 |
| <b>Excepciones</b>          | <b>Paso</b>                                                                                                                            | <b>Acción</b>                                                                                                                                                                                                   |
|                             | 3                                                                                                                                      | Si el empleado del vídeo-club solicita cancelar la operación, el sistema cancela la operación, a continuación este caso de uso termina                                                                          |
| <b>Rendimiento</b>          | <b>Paso</b>                                                                                                                            | <b>Cota de tiempo</b>                                                                                                                                                                                           |
|                             | 4                                                                                                                                      | 5 segundos                                                                                                                                                                                                      |
| <b>Frecuencia esperada</b>  | 1 vez/día                                                                                                                              |                                                                                                                                                                                                                 |
| <b>Comentarios</b>          | Si el periodo es <i>siempre</i> , el tiempo de respuesta puede ser muy alto                                                            |                                                                                                                                                                                                                 |

### B.5.4 Requisitos no funcionales

|                             |                                                                                                                |
|-----------------------------|----------------------------------------------------------------------------------------------------------------|
| <b>RNF-01</b>               | <b>Copias de seguridad</b>                                                                                     |
| <b>Objetivos asociados</b>  | –                                                                                                              |
| <b>Requisitos asociados</b> | –                                                                                                              |
| <b>Descripción</b>          | El sistema deberá incorporar algún mecanismo que permita realizar copias de seguridad de los datos almacenados |
| <b>Comentarios</b>          | ninguno                                                                                                        |

|                             |                                                                                                                                                                        |
|-----------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>RNF-02</b>               | <b>Entorno de explotación</b>                                                                                                                                          |
| <b>Objetivos asociados</b>  | –                                                                                                                                                                      |
| <b>Requisitos asociados</b> | –                                                                                                                                                                      |
| <b>Descripción</b>          | El sistema deberá funcionar en un entorno de 2 PC's Pentium con 16 Mbytes de RAM y 2 GBytes de disco duro conectados en red con sistema operativo Microsoft Windows 98 |
| <b>Comentarios</b>          | ninguno                                                                                                                                                                |

|                             |                                                                                       |
|-----------------------------|---------------------------------------------------------------------------------------|
| <b>RNF-03</b>               | <b>Portabilidad</b>                                                                   |
| <b>Objetivos asociados</b>  | –                                                                                     |
| <b>Requisitos asociados</b> | –                                                                                     |
| <b>Descripción</b>          | El sistema deberá ser fácilmente portable al sistema operativo Microsoft Windows 2000 |
| <b>Comentarios</b>          | ninguno                                                                               |

## B.6 Bibliografía

- [Boehm *et al.* 1994] B. W. Boehm, P. Bose, E. Horowitz, y M.-J. Lee. Software Requirements as Negotiated Win Conditions. En *Proceedings of the First International Conference on Requirements Engineering*, 1994. Disponible en <http://sunset.usc.edu/TechRpts/Papers/NGPM-Requirements93.ps>.
- [Booch *et al.* 1999] G. Booch, J. Rumbaugh, y I. Jacobson. *The Unified Modeling Language User Guide*. Addison-Wesley, 1999.
- [García *et al.* 2000] J. García, M. J. Ortín, B. Moros, y J. Nicolás. Modelado de Casos de Uso y Conceptual a partir del Modelado del Negocio. En *Actas de las V Jornadas de Trabajo Menhir*, Granada, 2000.
- [Laguna *et al.* 1999] M. A. Laguna, J. M. Marqués, y F. J. García. Una Herramienta para la Captura de Requisitos de Usuario. En *Actas de las JISBD'99*, Cáceres, 1999.



- [MAP 1995] MAP. *Metodología de Planificación y Desarrollo de Sistemas de Información. MÉTRICA Versión 2.1*. Tecnos/Ministerio para las Administraciones Públicas, 1995.



## Apéndice C

# Metodología para el análisis de requisitos de sistemas de información

### C.1 Objetivo de la metodología

El objetivo de esta metodología es la definición de las tareas a realizar, los productos a obtener y las técnicas a emplear durante la actividad de *análisis de requisitos* de la fase de *ingeniería de requisitos* del ciclo de vida de desarrollo del software.

En esta metodología se distinguen dos tipos de productos: los productos *entregables* y los productos *no entregables* o *internos*. Los productos *entregables* son aquellos que se entregan oficialmente al cliente como parte del desarrollo en fechas previamente acordadas, mientras que los *no entregables* son productos internos al desarrollo que no se entregan al cliente.

El único producto entregable definido en esta metodología es el *Documento de Análisis del Sistema* (DAS), definido en la sección C.3.1, pág. 298.

La estructura de este documento es la siguiente: en la sección C.2 se describen las tareas recomendadas para obtener el DAS, en la sección C.3 se definen los productos entregables, en este caso el DAS, y por último, en la sección C.4 se describen las técnicas recomendadas para obtener los productos. También se incluye como apéndice un ejemplo de aplicación de esta metodología.

## **C.2 Tareas recomendadas**

Las tareas recomendadas para obtener los productos descritos en esta metodología son las siguientes:

**Tarea 1:** Analizar los requisitos de almacenamiento de información

**Tarea 2:** Analizar los requisitos funcionales

**Tarea 3:** Analizar los requisitos no funcionales

**Tarea 4:** Desarrollar prototipos

El orden recomendado de realización para estas tareas es: 1...4, aunque las tareas 1, 2 y 3 pueden realizarse simultáneamente y el tipo de sistema puede determinar un orden u otro. La tarea 4 es opcional, dependiendo del ciclo de vida adoptado en el desarrollo.

En las siguientes secciones se describen cada una de las tareas mencionadas.

### **C.2.1 Tarea 1: Analizar los requisitos de almacenamiento de información**

#### **C.2.1.1 Objetivos**

- Identificar conflictos en los requisitos de almacenamiento de información
- Incrementar el conocimiento del problema
- Establecer las bases para el diseño de la persistencia del estado del sistema

#### **C.2.1.2 Descripción**

En esta tarea se debe obtener el modelo estático del sistema a partir de los requisitos de almacenamiento de información obtenidos en la actividad de elicitación de requisitos, identificando los posibles conflictos que pudieran surgir. Se deben identificar los tipos de objetos que forman el sistema, así como sus asociaciones, agregaciones y clasificaciones.

### **C.2.1.3 Productos internos**

No hay productos internos en esta tarea.

### **C.2.1.4 Productos entregables**

- Diagramas de tipos como parte del DAS (ver sección C.3.1.6, pág. 302)
- Catálogo de tipos y asociaciones como parte del DAS (ver sección C.3.1.7, pág. 302)
- Conflictos pendientes de resolución como parte del DAS (ver sección C.3.1.13, pág. 304), como parte del Documento de Requisitos del Sistema (DRS) o como parte de un documento específico de conflictos

### **C.2.1.5 Técnicas**

- Diagrama de tipos
- Plantillas para tipos
- Plantillas para asociaciones
- Plantillas para conflictos

## **C.2.2 Tarea 2: Analizar los requisitos funcionales**

### **C.2.2.1 Objetivos**

- Identificar conflictos en los requisitos funcionales
- Incrementar el conocimiento del problema
- Establecer las bases para el diseño de la conducta del sistema

### **C.2.2.2 Descripción**

En esta tarea se debe obtener el modelo de comportamiento del sistema a partir de los requisitos funcionales obtenidos en la actividad de elicitación de requisitos, identificando los posibles conflictos que pudieran surgir. Se deben especificar tanto las operaciones del sistema como los estados y transiciones de los tipos de objetos identificados en la tarea anterior que tengan un dinamismo significativo.

### **C.2.2.3 Productos internos**

No hay productos internos en esta tarea.

### **C.2.2.4 Productos entregables**

- Catálogo de tipos y asociaciones como parte del DAS (ver sección C.3.1.7, pág. 302)
- Catálogo de operaciones del sistema como parte del DAS (ver sección C.3.1.10, pág. 303)
- Conflictos pendientes de resolución como parte del DAS (ver sección C.3.1.13, pág. 304), como parte del DRS o como parte de un documento específico de conflictos

### **C.2.2.5 Técnicas**

- Plantilla para operaciones del sistema
- Diagrama de estados
- Plantillas para conflictos

## **C.2.3 Tarea 3: Analizar los requisitos no funcionales**

### **C.2.3.1 Objetivos**

- Identificar conflictos en los requisitos no funcionales
- Incrementar el conocimiento del problema

- Establecer las bases para el diseño de los aspectos no funcionales del sistema

#### **C.2.3.2 Descripción**

En esta tarea se deben identificar los posibles conflictos en los requisitos no funcionales.

#### **C.2.3.3 Productos internos**

No hay productos internos en esta tarea.

#### **C.2.3.4 Productos entregables**

- Conflictos pendientes de resolución como parte del DAS (ver sección C.3.1.13, pág. 304), como parte del DRS o como parte de un documento específico de conflictos

#### **C.2.3.5 Técnicas**

- Lectura detenida y aplicación de criterios fruto de la experiencia
- Plantillas para conflictos

### **C.2.4 Tarea 4: Desarrollar prototipos**

#### **C.2.4.1 Objetivos**

- Definir la interfaz de usuario
- Incrementar el conocimiento del problema
- Comprobar propiedades del sistema

#### **C.2.4.2 Descripción**

En esta tarea se deben desarrollar prototipos que permitan al cliente tener una idea más clara del sistema a desarrollar e identificar nuevos requisitos o conflictos que hayan permanecido *ocultos* hasta el momento.

#### **C.2.4.3 Productos internos**

- Prototipo del sistema

#### **C.2.4.4 Productos entregables**

- Interfaces de usuario del catálogo de operaciones del sistema como parte del DAS (ver sección C.3.1.10, pág. 303)

#### **C.2.4.5 Técnicas**

- Prototipado de sistemas software (ver sección C.4.7, pág. 306).

### **C.3 Productos entregables**

El único producto entregable que se contempla en esta metodología es el *Documento de Análisis del Sistema* (DAS).

#### **C.3.1 Documento de análisis del sistema**

La estructura del DAS puede verse en la figura C.1. En las siguientes secciones se describe con detalle cada sección del DAS.

##### **C.3.1.1 Portada**

La portada del DAS debe tener el formato que puede verse en la figura C.2. Los elementos que deben aparecer son los siguientes:

- **Nombre del proyecto:** el nombre del proyecto al que pertenece el DAS.
- **Versión:** la versión del DAS que se entrega al cliente. La versión se compone de dos números *X* e *Y*. El primero indica la versión, y se debe incrementar cada vez que se hace una nueva entrega formal al cliente. Cuando se incremente el primer número, el segundo debe



|                                                                                           |
|-------------------------------------------------------------------------------------------|
| Portada                                                                                   |
| Lista de cambios                                                                          |
| Índice                                                                                    |
| Lista de figuras                                                                          |
| Lista de tablas                                                                           |
| 1 Introducción                                                                            |
| 2 Diagramas de tipos                                                                      |
| 3 Catálogo de tipos y asociaciones del sistema                                            |
| 3.1 Tipo X                                                                                |
| 3.1.1 Descripción tipo X                                                                  |
| 3.1.2 Atributos tipo X                                                                    |
| 3.1.3 Enlaces tipo X                                                                      |
| 3.1.4 Invariante tipo X                                                                   |
| 3.1.5 Diagrama de estados tipo X <i>[opcional]</i>                                        |
| 3.2 Asociación Y entre A,B,...,C                                                          |
| 3.2.1 Descripción asociación Y(A,B,...,C)                                                 |
| 3.2.2 Roles asociación Y(A,B,...,C)                                                       |
| 3.2.3 Atributos asociación Y(A,B,...,C) <i>[opcional]</i>                                 |
| 3.2.4 Enlaces asociación Y(A,B,...,C) <i>[opcional]</i>                                   |
| 3.2.5 Invariante asociación Y(A,B,...,C) <i>[opcional]</i>                                |
| 3.2.6 Diagrama de estados asociación Y(A,B,...,C) <i>[opcional]</i>                       |
| 4 Catálogo de operaciones del sistema                                                     |
| 4.1 Operación Z                                                                           |
| 4.1.1 Descripción operación Z                                                             |
| 4.1.2 Diagrama de acción conjunta de la operación Z <i>[opcional]</i>                     |
| 4.1.3 Diagrama de secuencia de la operación Z <i>[opcional]</i>                           |
| 4.1.4 Interfaz de usuario de la operación Z <i>[opcional]</i>                             |
| 5 Matriz de rastreabilidad requisitos/modelos                                             |
| 6 Conflictos pendientes de resolución <i>[opcional, pueden ir en un documento aparte]</i> |
| 7 Glosario de términos <i>[opcional]</i>                                                  |
| Apéndices <i>[opcional]</i>                                                               |

Figura C.1: Estructura del Documento de Análisis del Sistema

volver a comenzar en cero. El segundo número indica cambios dentro de la misma versión aún no entregada, y se debe incrementar cada vez que se imprima una versión con cambios respecto a la última que se imprimió y que no se vaya a entregar formalmente todavía. Este tipo de versiones pueden ser internas al equipo de desarrollo o ser entregadas al cliente a título orientativo.

- **Fecha:** fecha de la publicación de la versión.
- **Equipo de desarrollo:** nombre de la empresa o equipo de desarrollo.
- **Cliente:** nombre del cliente, normalmente otra empresa.

#### **C.3.1.2 Lista de cambios**

El documento debe incluir una lista de cambios en la que se especifiquen, para cada versión del documento, los cambios producidos en el mismo con un formato similar al que puede verse en la figura C.3. Para cada cambio realizado se debe incluir el número de orden, la fecha, una descripción y los autores.

#### **C.3.1.3 Índice**

El índice del DAS debe indicar la página en la que comienza cada sección, subsección o apartado del documento. En la medida de lo posible, se sangrarán las entradas del índice para ayudar a comprender la estructura del documento.

#### **C.3.1.4 Listas de figuras y tablas**

El DAS deberá incluir listas de las figuras y tablas que aparezcan en el mismo. Dichas listas serán dos índices que indicarán el número, la descripción y la página en que aparece cada figura o tabla del DAS.

#### **C.3.1.5 Introducción**

Esta sección debe contener una descripción breve de las principales características del sistema software que se va a desarrollar, la situación actual que genera la necesidad del nuevo desarrollo, la problemática que se

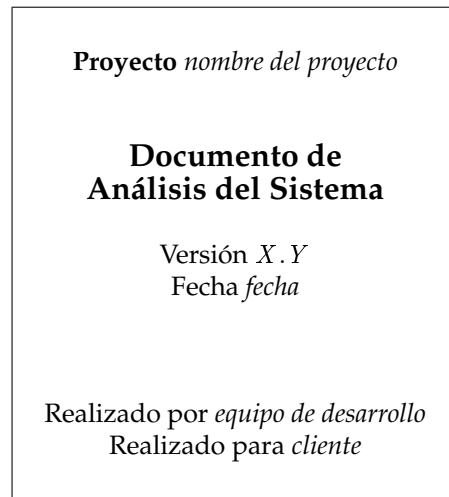


Figura C.2: Portada del Documento de Análisis del Sistema

| Núm. | Fecha     | Descripción                                         | Autor/es  |
|------|-----------|-----------------------------------------------------|-----------|
| 0    | $fecha_0$ | Versión x.y                                         | $autor_0$ |
| 1    | $fecha_1$ | <i>descripción cambio</i> <sub>1</sub>              | $autor_1$ |
| ⋮    | ⋮         | ⋮                                                   | ⋮         |
| $n$  | $fecha_n$ | <i>descripción cambio</i> <sub><math>n</math></sub> | $autor_n$ |

Figura C.3: Lista de cambios del Documento de Análisis del Sistema

acomete, y cualquier otra consideración que sitúe al posible lector en el contexto oportuno para comprender el resto del documento.

#### C.3.1.6 Diagramas de tipos

Este apartado debe contener los diagramas de tipos del modelo del sistema construido durante el análisis. Para sistemas complejos puede ser conveniente agrupar los diagramas en *paquetes* o *subsistemas*, tal como se describe en [Booch *et al.* 1999].

#### C.3.1.7 Catálogo de tipos y asociaciones

Esta sección se divide en los siguientes apartados en los que se describen los tipos y asociaciones identificados durante el análisis. Si en los diagramas de tipos se han agrupado los tipos y asociaciones en paquetes o subsistemas, el catálogo de tipos y asociaciones deberá reflejar también dicha división.

Tanto los tipos como las asociaciones deben estar ordenadas alfabéticamente por su nombre dentro de cada paquete o subsistema.

#### C.3.1.8 Tipo X

Este apartado, que se repite para cada tipo X identificada en el análisis, debe tener el siguiente contenido:

- **Descripción:** descripción del tipo mediante la plantilla de tipo
- **Atributos:** descripción, ordenada alfabéticamente, de los atributos del tipo mediante las plantillas de atributo
- **Enlaces:** descripción, ordenada alfabéticamente, de los enlaces del tipo mediante las plantillas de enlace
- **Invariante:** descripción de la invariante del tipo mediante la plantilla de invariante
- **Diagrama de estados:** diagrama de estados correspondiente al tipo, si se ha identificado alguno

### C.3.1.9 Asociación Y entre A, B, ..., C

Este apartado, que se repite para cada asociación Y entre los tipos A, B, ..., C identificada en el análisis, debe tener el siguiente contenido, muy similar al de los tipos:

- **Descripción:** descripción de la asociación mediante la plantilla de asociación
- **Roles:** descripción, ordenada alfabéticamente, de los roles que juegan los tipos participantes en la asociación mediante la plantilla de rol
- El resto del contenido de este apartado es opcional y su descripción coincide con el de la descripción de tipos:
  - **Atributos**
  - **Enlaces**
  - **Invariante**
  - **Diagrama de estados**

### C.3.1.10 Catálogo de operaciones del sistema

Esta sección se divide en los siguientes apartados en los que se describen las operaciones del sistema identificadas durante el análisis.

### C.3.1.11 Operación Z

Este apartado, que se repite para cada operación Z identificada en el análisis, debe tener el siguiente contenido:

- **Descripción:** descripción de la operación mediante la plantilla de operación de sistema
- **Diagrama de acción conjunta:** diagrama de acción conjunta correspondiente a la operación, si se considera oportuno
- **Diagrama de secuencia:** diagrama de secuencia correspondiente a la operación, si se considera oportuno

- **Interfaz de usuario:** interfaz de usuario asociada a la operación, en el caso de que se haya construido algún prototipo (ver sección C.4.7, pág. 306). Se deberán mostrar la/s pantalla/s y/o informes asociados a la operación así como su descripción.

### C.3.1.12 Matriz de rastreabilidad requisitos/modelo

Este apartado debe contener una matriz *requisito/elemento de modelo*, de forma que para cada requisito identificado en el *Documento de Requisitos del Sistema* se pueda conocer con qué tipos, asociaciones y operaciones del sistema está asociado. El formato de la matriz de rastreabilidad puede verse en la figura C.4.

Dado que los requisitos no funcionales no suelen recogerse en el modelo del sistema, si se considera oportuno pueden eliminarse de la matriz de rastreabilidad.

|                    | RI-01 | RI-02 | ... | RF-01 | RF-02 | ... | RNF-01 | RNF-02 | ... |
|--------------------|-------|-------|-----|-------|-------|-----|--------|--------|-----|
| <b>Tipo</b> $X_1$  | •     | •     |     |       |       |     |        |        |     |
| <b>Tipo</b> $X_2$  |       | •     |     |       |       |     |        |        |     |
| ...                |       |       |     |       |       |     |        |        |     |
| <b>Asoc.</b> $Y_1$ | •     |       |     |       |       |     |        |        |     |
| <b>Asoc.</b> $Y_2$ | •     | •     |     |       |       |     |        |        |     |
| ...                |       |       |     |       |       |     |        |        |     |
| <b>OpS.</b> $Z_1$  |       |       |     | •     |       |     |        |        |     |
| <b>OpS.</b> $Z_2$  |       |       |     |       | •     |     |        |        |     |
| ...                |       |       |     |       |       |     |        |        |     |

Figura C.4: Matriz de rastreabilidad del Documento de Análisis del Sistema

### C.3.1.13 Conflictos pendientes de resolución

Esta sección, que se incluirá en el caso de que no se opte por registrar los conflictos en un documento aparte, deberá contener los conflictos identificados durante el proceso y que aún están pendientes de resolución, descritos mediante la plantilla para conflictos.

### C.3.1.14 Glosario de términos

Esta sección, que se incluirá si se considera oportuno, deberá contener una lista ordenada alfabéticamente de los términos específicos del dominio del

problema, acrónimos y abreviaturas que aparezcan en el documento y que se considere que su significado deba ser aclarado. Cada término deberá acompañarse de su significado.

#### **C.3.1.15 Apéndices**

Los apéndices se usarán para proporcionar información adicional a la documentación obligatoria del documento. Sólo deben aparecer si se consideran oportunos y se identificarán con letras ordenadas alfabéticamente: A, B, C, etc.

## **C.4 Técnicas**

A continuación, se describen algunas de las técnicas que se proponen en esta metodología para obtener los productos de las tareas que se han descrito.

### **C.4.1 Diagrama de tipos**

Los diagramas de tipos permiten expresar gráficamente aspectos estáticos del modelo de un sistema. En esta metodología se propone que los diagramas de tipos usen la notación UML [Booch *et al.* 1999].

### **C.4.2 Plantillas para especificación de tipos y asociaciones**

**Nota:** Esta sección coincide con la sección 4.7.1, pág. 164, por lo que se ha omitido para evitar repeticiones innecesarias.

### **C.4.3 Diagrama de estados**

Los diagramas de estados permiten expresar aspectos de comportamiento a nivel de una clase de objetos. En esta metodología se propone que los diagramas de estados se realicen utilizando la notación UML [Booch *et al.* 1999] con la semántica descrita en [D'Souza y Wills 1999].

#### **C.4.4 Diagrama de acción conjunta**

Los diagramas de acción conjunta permiten representar de manera visual las clases de los objetos que intervienen en una determinada operación del sistema. En esta metodología se propone que los diagramas de acción conjunta usen la notación descrita en [D'Souza y Wills 1999].

#### **C.4.5 Diagrama de secuencia**

Los diagramas de secuencia permiten representar visualmente el intercambio de estímulos y respuestas entre los actores y el sistema durante el transcurso de la realización de una operación del sistema. En esta metodología se propone que los diagramas de secuencia se realicen utilizando la notación UML [Booch *et al.* 1999], aunque aplicados solamente a las interacciones entre los actores y el sistema y no entre los objetos que componen el sistema.

#### **C.4.6 Plantillas para especificación de operaciones del sistema**

**Nota:** Esta sección coincide con la sección 4.7.2, pág. 170, por lo que se ha omitido para evitar repeticiones innecesarias.

#### **C.4.7 Construcción de prototipos**

Esta metodología no establece ninguna restricción para la elaboración de prototipos. Se podrán utilizar herramientas de desarrollo rápido, prototipos en papel o cualquier otra técnica que se considere oportuna.

### **C.5 Ejemplo: gestión de un vídeo-club**

En este apéndice se ofrecen algunos ejemplos de aplicación de las técnicas propuestas en esta norma. Para ello se ha continuado con el ejemplo de la gestión de un pequeño vídeo-club que se utilizó en la *Metodología para la Elicitación de Requisitos de Sistemas Software*, aunque por motivos de



espacio se ha optado por no incluir todas las operaciones del sistema correspondientes a los casos de uso identificados en el ejemplo citado, sino solamente un subconjunto significativo.

Aunque no están definidos en la actual versión de OCL [Rat 1997], se han asumido los tipos básicos `Date` y `Time` para representar fechas y horas respectivamente. Se ha asumido también que se ha definido la operación menor o igual sobre ambos tipos (`<=`) y que `Date.today` y `Time.now` devuelven la fecha y hora actual respectivamente, de forma similar a la propuesta de [Warmer y Kleppe 1999].

En las especificaciones en OCL se utilizarán las expresiones `isOrderedBy` sobre secuencias, `new` y `dead` sobre tipos, la palabra reservada `respuesta`, `result` se utilizará castellanizada como `resultado` y se permitirá el uso de tipos sin nombre, tal como se describió en la sección 4.8.2, pág. 176.

En las operaciones del sistema especificadas se ha asumido que se ha desarrollado un prototipo de la interfaz de usuario del sistema.

### C.5.1 Diagramas de tipos

Los diagramas de tipos correspondientes al modelo, y realizados utilizando la notación UML [Booch *et al.* 1999], pueden verse en las figuras C.6 y C.7. Para mayor claridad se han dividido en dos *paquetes* o *subsistemas* que pueden verse en la figura C.5.

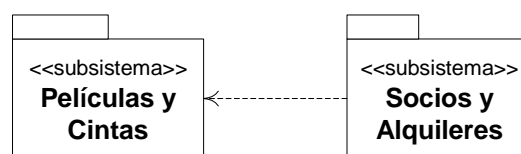


Figura C.5: Diagrama de *subsistemas*

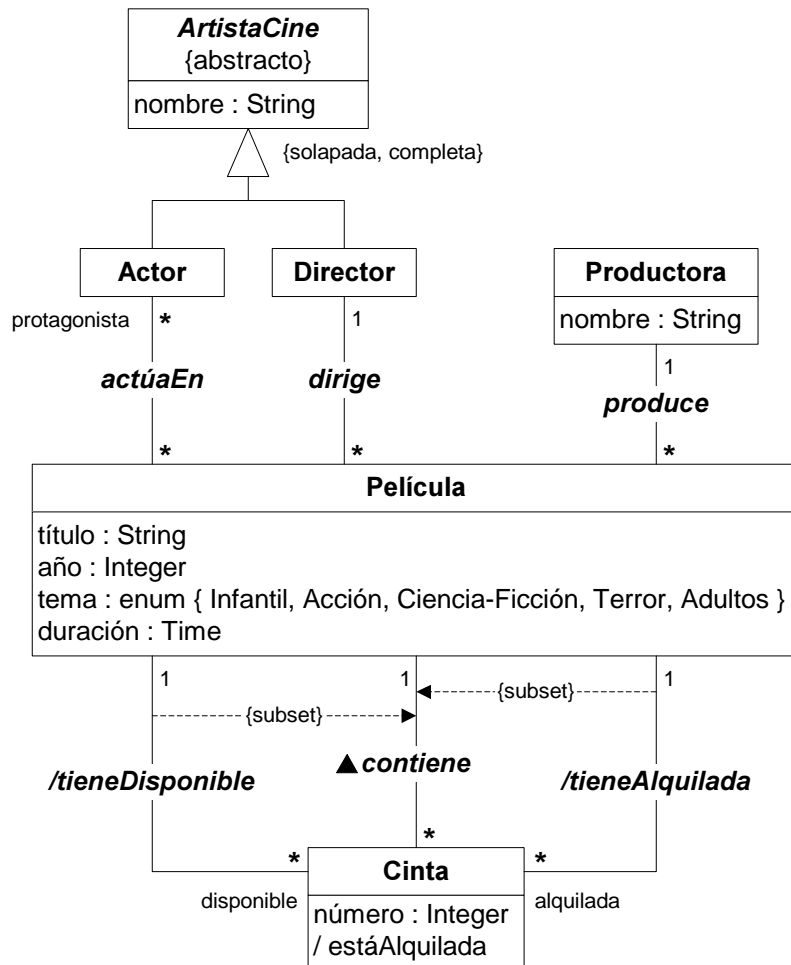


Figura C.6: Diagrama de tipos del subsistema *Películas y Cintas*



## C.5.2 Tipos y asociaciones

### C.5.2.1 Subsistema Películas y Cintas

Este subsistema contiene los tipos y asociaciones relacionadas con las películas y cintas del vídeo-club.

#### C.5.2.2 Tipo Actor

##### Descripción tipo Actor

|                             |                                                                                         |                 |              |
|-----------------------------|-----------------------------------------------------------------------------------------|-----------------|--------------|
| <b>Tipo</b>                 | <b>Actor</b>                                                                            |                 |              |
| <b>Requisitos asociados</b> | • RI-01 Información sobre películas                                                     |                 |              |
| <b>Descripción</b>          | Este tipo concreto representa los actores protagonistas de las películas del vídeo-club |                 |              |
| <b>Supertipos</b>           | ArtistaCine                                                                             |                 |              |
| <b>Subtipos</b>             | –                                                                                       |                 |              |
| <b>Componentes</b>          | <b>Nombre</b>                                                                           | <b>Tipo OCL</b> | <b>Mult.</b> |
|                             | –                                                                                       | –               | –            |
| <b>Comentarios</b>          | Ninguno                                                                                 |                 |              |

##### Atributos tipo Actor

No se han identificado atributos para este tipo.

##### Enlaces tipo Actor

|                        |                                                           |
|------------------------|-----------------------------------------------------------|
| <b>Enlace variable</b> | <b>Actor::película</b>                                    |
| <b>Descripción</b>     | Conjunto de películas en las que el actor es protagonista |
| <b>Tipo OCL</b>        | Set( Película )                                           |
| <b>Asociación</b>      | actúaEn(Actor,Película)                                   |
| <b>Comentarios</b>     | Ninguno                                                   |

##### Invariante tipo Actor

No se han identificado restricciones para este tipo.

### C.5.2.3 Tipo ArtistaCine

#### Descripción tipo ArtistaCine

|                             |                                                                                              |                 |              |
|-----------------------------|----------------------------------------------------------------------------------------------|-----------------|--------------|
| <b>Tipo</b>                 | <b>ArtistaCine</b>                                                                           |                 |              |
| <b>Requisitos asociados</b> | • RI-01 Información sobre películas                                                          |                 |              |
| <b>Descripción</b>          | Este tipo abstracto representa los actores protagonistas de las películas del vídeo-club     |                 |              |
| <b>Supertipos</b>           | –                                                                                            |                 |              |
| <b>Subtipos</b>             | <b>Actor, Director</b> (solapadas, completas)                                                |                 |              |
| <b>Componentes</b>          | <b>Nombre</b>                                                                                | <b>Tipo OCL</b> | <b>Mult.</b> |
|                             | –                                                                                            | –               | –            |
| <b>Comentarios</b>          | Puede haber artistas de cine que sean actores y directores a la vez, por ejemplo Woody Allen |                 |              |

#### Atributos tipo ArtistaCine

|                           |                            |
|---------------------------|----------------------------|
| <b>Atributo constante</b> | <b>ArtistaCine::nombre</b> |
| <b>Descripción</b>        | Nombre del artista de cine |
| <b>Tipo OCL</b>           | String                     |
| <b>Comentarios</b>        | Ninguno                    |

#### Enlaces tipo ArtistaCine

Este tipo no participa en ninguna asociación.

#### Invariante tipo ArtistaCine

|                    |                                                                                                                    |
|--------------------|--------------------------------------------------------------------------------------------------------------------|
| <b>Invariante</b>  | <b>ArtistaCine</b>                                                                                                 |
| <b>Descripción</b> | • No puede haber dos artistas de cine con el mismo nombre (nombre es <i>clave</i> )                                |
| <b>Expresión</b>   | <code>ArtistaCine.allInstances-&gt;forall( a1, a2  <br/>(a1 &lt;&gt; a2) = (a1.nombre &lt;&gt; a2.nombre) )</code> |
| <b>Comentarios</b> | Ninguno                                                                                                            |

### C.5.2.4 Tipo Cinta

#### Descripción tipo Cinta

|                             |                                                                              |                 |              |
|-----------------------------|------------------------------------------------------------------------------|-----------------|--------------|
| <b>Tipo</b>                 | <b>Cinta</b>                                                                 |                 |              |
| <b>Requisitos asociados</b> | • RI-01 Información sobre películas                                          |                 |              |
| <b>Descripción</b>          | Este tipo concreto representa las cintas actualmente en poder del vídeo-club |                 |              |
| <b>Supertipos</b>           | –                                                                            |                 |              |
| <b>Subtipos</b>             | –                                                                            |                 |              |
| <b>Componentes</b>          | <b>Nombre</b>                                                                | <b>Tipo OCL</b> | <b>Mult.</b> |
|                             | –                                                                            | –               | –            |
| <b>Comentarios</b>          | Ninguno                                                                      |                 |              |

#### Atributos tipo Cinta

|                          |                                               |
|--------------------------|-----------------------------------------------|
| <b>Atributo derivado</b> | <b>Cinta::estáAlquilada</b>                   |
| <b>Descripción</b>       | Indica si la cinta está alquilada actualmente |
| <b>Expresión</b>         | <code>self.alquilerActual-&gt;notEmpty</code> |
| <b>Comentarios</b>       | Ninguno                                       |

|                           |                                                                                                                            |
|---------------------------|----------------------------------------------------------------------------------------------------------------------------|
| <b>Atributo constante</b> | <b>Cinta::número</b>                                                                                                       |
| <b>Descripción</b>        | Número identificativo de la cinta                                                                                          |
| <b>Tipo OCL</b>           | Integer                                                                                                                    |
| <b>Comentarios</b>        | Se debe generar automáticamente; es imprescindible para que el empleado del vídeo-club localice las cintas en los estantes |

#### Enlaces tipo Cinta

|                        |                                                      |
|------------------------|------------------------------------------------------|
| <b>Enlace variable</b> | <b>Cinta::alquiler</b>                               |
| <b>Descripción</b>     | Alquileres de los que ha sido o es objeto la cinta   |
| <b>Tipo OCL</b>        | <code>Sequence( Alquiler )</code>                    |
| <b>Asociación</b>      | <b>esObjetoDe(Cinta, Alquiler)</b>                   |
| <b>Comentarios</b>     | Los alquileres están ordenados por fecha de alquiler |

|                        |                                                                 |
|------------------------|-----------------------------------------------------------------|
| <b>Enlace derivado</b> | <b>Cinta::alquilerActual</b>                                    |
| <b>Descripción</b>     | Alquiler en curso de la cinta                                   |
| <b>Expresión</b>       | <code>self.alquiler-&gt;select( a   not a.estáDevuelto )</code> |
| <b>Asociación</b>      | <b>esObjetoActualmenteDe(Cinta, Alquiler)</b>                   |
| <b>Comentarios</b>     | Ninguno                                                         |

|                        |                                  |
|------------------------|----------------------------------|
| <b>Enlace variable</b> | <b>Cinta::película</b>           |
| <b>Descripción</b>     | Película contenida en la cinta   |
| <b>Tipo OCL</b>        | Película                         |
| <b>Asociación</b>      | <b>contiene(Cinta, Película)</b> |
| <b>Comentarios</b>     | Ninguno                          |

|                        |                                                |
|------------------------|------------------------------------------------|
| <b>Enlace derivado</b> | <b>Cinta::socio</b>                            |
| <b>Descripción</b>     | Socio que tiene alquilada la cinta actualmente |
| <b>Expresión</b>       | <code>self.alquilerActual.socio</code>         |
| <b>Asociación</b>      | <b>tieneAlquilada(Socio, Cinta)</b>            |
| <b>Comentarios</b>     | Ninguno                                        |

### Invariante tipo Cinta

|                    |                                                                                                                                                                                                                                                                                                              |
|--------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Invariante</b>  | <b>Cinta</b>                                                                                                                                                                                                                                                                                                 |
| <b>Descripción</b> | <ul style="list-style-type: none"> <li>• No puede haber dos cintas con el mismo número (número es <i>clave</i>)</li> <li>• Los alquileres deben estar ordenados por fecha de alquiler</li> <li>• No hay alquileres duplicados</li> <li>• La cinta sólo puede tener pendiente un alquiler a la vez</li> </ul> |
| <b>Expresión</b>   | <pre>Cinta.allInstances-&gt;forall( c1, c2       ( c1 &lt;&gt; c2 ) = ( c1.número &lt;&gt; c2.número ) ) and self.alquiler-&gt;isOrderedBy( &lt;=, fechaAlquiler ) and self.alquiler-&gt;size = self.alquiler-&gt;asSet-&gt;size and self.alquilerActual-&gt;size &lt;= 1</pre>                              |
| <b>Comentarios</b> | Ninguno                                                                                                                                                                                                                                                                                                      |

**C.5.2.5 Tipo Director****Descripción tipo Director**

|                             |                                                                              |                 |              |
|-----------------------------|------------------------------------------------------------------------------|-----------------|--------------|
| <b>Tipo</b>                 | <b>Director</b>                                                              |                 |              |
| <b>Requisitos asociados</b> | • RI-01 Información sobre películas                                          |                 |              |
| <b>Descripción</b>          | Este tipo concreto representa los directores de las películas del vídeo-club |                 |              |
| <b>Supertipos</b>           | ArtistaCine                                                                  |                 |              |
| <b>Subtipos</b>             | –                                                                            |                 |              |
| <b>Componentes</b>          | <b>Nombre</b>                                                                | <b>Tipo OCL</b> | <b>Mult.</b> |
|                             | –                                                                            | –               | –            |
| <b>Comentarios</b>          | Ninguno                                                                      |                 |              |

**Atributos tipo Director**

No se han identificado atributos para este tipo.

**Enlaces tipo Director**

|                        |                                                 |
|------------------------|-------------------------------------------------|
| <b>Enlace variable</b> | <b>Director::película</b>                       |
| <b>Descripción</b>     | Conjunto de películas dirigidas por el director |
| <b>Tipo OCL</b>        | <code>Set( Película )</code>                    |
| <b>Asociación</b>      | <b>dirige(Director,Película)</b>                |
| <b>Comentarios</b>     | Ninguno                                         |

**Invariante tipo Director**

No se han identificado restricciones para este tipo.



### C.5.2.6 Tipo Película

#### Descripción tipo Película

|                             |                                                                                                 |                 |              |
|-----------------------------|-------------------------------------------------------------------------------------------------|-----------------|--------------|
| <b>Tipo</b>                 | <b>Película</b>                                                                                 |                 |              |
| <b>Requisitos asociados</b> | ● RI-01 Información sobre películas                                                             |                 |              |
| <b>Descripción</b>          | Este tipo concreto representa las películas que han estado o están disponibles en el vídeo-club |                 |              |
| <b>Supertipos</b>           | –                                                                                               |                 |              |
| <b>Subtipos</b>             | –                                                                                               |                 |              |
| <b>Componentes</b>          | <b>Nombre</b>                                                                                   | <b>Tipo OCL</b> | <b>Mult.</b> |
|                             | –                                                                                               | –               | –            |
| <b>Comentarios</b>          | Ninguno                                                                                         |                 |              |

#### Atributos tipo Película

|                           |                                      |
|---------------------------|--------------------------------------|
| <b>Atributo constante</b> | <b>Película::año</b>                 |
| <b>Descripción</b>        | Año en el que se estrenó la película |
| <b>Tipo OCL</b>           | Integer                              |
| <b>Comentarios</b>        | Ninguno                              |

|                           |                           |
|---------------------------|---------------------------|
| <b>Atributo constante</b> | <b>Película::duración</b> |
| <b>Descripción</b>        | Duración de la película   |
| <b>Tipo OCL</b>           | Time                      |
| <b>Comentarios</b>        | Ninguno                   |

|                           |                                                           |
|---------------------------|-----------------------------------------------------------|
| <b>Atributo constante</b> | <b>Película::tema</b>                                     |
| <b>Descripción</b>        | Tema de la película                                       |
| <b>Tipo OCL</b>           | enum {Infantil, Acción, Ciencia-Ficción, Terror, Adultos} |
| <b>Comentarios</b>        | Ninguno                                                   |

|                           |                         |
|---------------------------|-------------------------|
| <b>Atributo constante</b> | <b>Película::título</b> |
| <b>Descripción</b>        | Título de la película   |
| <b>Tipo OCL</b>           | String                  |
| <b>Comentarios</b>        | Ninguno                 |

#### Enlaces tipo Película

|                        |                                                           |
|------------------------|-----------------------------------------------------------|
| <b>Enlace derivado</b> | <b>Película::alquilada</b>                                |
| <b>Descripción</b>     | Cintas de la película actualmente alquiladas              |
| <b>Expresión</b>       | <code>self.cinta-&gt;select( c   c.estáAlquilada )</code> |
| <b>Asociación</b>      | <b>tieneAlquilada(Película, Cinta)</b>                    |
| <b>Comentarios</b>     | Ninguno                                                   |

|                        |                                                                    |
|------------------------|--------------------------------------------------------------------|
| <b>Enlace variable</b> | <b>Película::cinta</b>                                             |
| <b>Descripción</b>     | Cintas de la película de las que dispone actualmente el vídeo-club |
| <b>Tipo OCL</b>        | Set( Cinta )                                                       |
| <b>Asociación</b>      | <b>contiene(Cinta, Película)</b>                                   |
| <b>Comentarios</b>     | Ninguno                                                            |

|                         |                                   |
|-------------------------|-----------------------------------|
| <b>Enlace constante</b> | <b>Película::director</b>         |
| <b>Descripción</b>      | Director de la película           |
| <b>Tipo OCL</b>         | Director                          |
| <b>Asociación</b>       | <b>dirige(Director, Película)</b> |
| <b>Comentarios</b>      | Ninguno                           |

|                        |                                                               |
|------------------------|---------------------------------------------------------------|
| <b>Enlace derivado</b> | <b>Película::disponible</b>                                   |
| <b>Descripción</b>     | Cintas de la película actualmente disponibles (no alquiladas) |
| <b>Expresión</b>       | <b>self.cinta-&gt;select( c   not c.estáAlquilada )</b>       |
| <b>Asociación</b>      | <b>tieneDisponible(Película, Cinta)</b>                       |
| <b>Comentarios</b>     | Ninguno                                                       |

|                         |                                     |
|-------------------------|-------------------------------------|
| <b>Enlace constante</b> | <b>Película::productora</b>         |
| <b>Descripción</b>      | Productora de la película           |
| <b>Tipo OCL</b>         | Productora                          |
| <b>Asociación</b>       | <b>produce(Productor, Película)</b> |
| <b>Comentarios</b>      | Ninguno                             |

|                         |                                                  |
|-------------------------|--------------------------------------------------|
| <b>Enlace constante</b> | <b>Película::protagonista</b>                    |
| <b>Descripción</b>      | Conjunto de actores protagonistas de la película |
| <b>Tipo OCL</b>         | Set( Actor )                                     |
| <b>Asociación</b>       | <b>actúaEn(Actor, Película)</b>                  |
| <b>Comentarios</b>      | Ninguno                                          |

### Invariante tipo Película

|                    |                                                                                                                                            |
|--------------------|--------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Invariante</b>  | <b>Película</b>                                                                                                                            |
| <b>Descripción</b> | • Las cintas alquiladas y no alquiladas forman una partición de todas las cintas de la película                                            |
| <b>Expresión</b>   | <b>self.cinta = self.disponible-&gt;union( self.alquilada )<br/>and<br/>self.disponible-&gt;intersection( self.alquilada )-&gt;isEmpty</b> |
| <b>Comentarios</b> | Ninguno                                                                                                                                    |

### C.5.2.7 Tipo Productora

#### Descripción tipo Productora

|                             |                                                                               |                 |              |
|-----------------------------|-------------------------------------------------------------------------------|-----------------|--------------|
| <b>Tipo</b>                 | <b>Productora</b>                                                             |                 |              |
| <b>Requisitos asociados</b> | • RI-01 Información sobre películas                                           |                 |              |
| <b>Descripción</b>          | Este tipo concreto representa las productoras de las películas del vídeo-club |                 |              |
| <b>Supertipos</b>           | –                                                                             |                 |              |
| <b>Subtipos</b>             | –                                                                             |                 |              |
| <b>Componentes</b>          | <b>Nombre</b>                                                                 | <b>Tipo OCL</b> | <b>Mult.</b> |
|                             | –                                                                             | –               | –            |
| <b>Comentarios</b>          | Ninguno                                                                       |                 |              |

#### Atributos tipo Productora

|                           |                           |
|---------------------------|---------------------------|
| <b>Atributo constante</b> | <b>Productora::nombre</b> |
| <b>Descripción</b>        | Nombre de la productora   |
| <b>Tipo OCL</b>           | String                    |
| <b>Comentarios</b>        | Ninguno                   |

#### Enlaces tipo Productora

|                        |                                                    |
|------------------------|----------------------------------------------------|
| <b>Enlace variable</b> | <b>Productora::película</b>                        |
| <b>Descripción</b>     | Conjunto de películas producidas por la productora |
| <b>Tipo OCL</b>        | Set( Película )                                    |
| <b>Asociación</b>      | <b>produce(Productora,Película)</b>                |
| <b>Comentarios</b>     | Ninguno                                            |

#### Invariante tipo Productora

|                    |                                                                                      |
|--------------------|--------------------------------------------------------------------------------------|
| <b>Invariante</b>  | <b>Productora</b>                                                                    |
| <b>Descripción</b> | • No puede haber dos productoras con el mismo nombre (nombre es clave)               |
| <b>Expresión</b>   | Productora.allInstances->forall( a1, a2  <br>(a1 <> a2) = (a1.nombre <> a2.nombre) ) |
| <b>Comentarios</b> | Ninguno                                                                              |

**C.5.2.8 Asociación actúaEn entre Actor y Película****Descripción asociación actúaEn(Actor, Película)**

|                             |                                                                                     |
|-----------------------------|-------------------------------------------------------------------------------------|
| <b>Asociación</b>           | <b>actúaEn entre Actor y Película</b>                                               |
| <b>Requisitos asociados</b> | • RI-01 Información sobre películas                                                 |
| <b>Descripción</b>          | Esta asociación representa el hecho de que un actor es protagonista en una película |
| <b>Comentarios</b>          | Ninguno                                                                             |

**Roles asociación actúaEn(Actor, Película)**

|                      |                                                                 |
|----------------------|-----------------------------------------------------------------|
| <b>Rol</b>           | <b>Actor juega rol protagonista en actúaEn(Actor, Película)</b> |
| <b>Descripción</b>   | Actor protagonista de una película                              |
| <b>Tipo OCL</b>      | Set( Actor )                                                    |
| <b>Multiplicidad</b> | 0..n                                                            |
| <b>Comentarios</b>   | Ninguno                                                         |

|                      |                                                                |
|----------------------|----------------------------------------------------------------|
| <b>Rol</b>           | <b>Película juega rol película en actúaEn(Actor, Película)</b> |
| <b>Descripción</b>   | Películas en la que un actor es protagonista                   |
| <b>Tipo OCL</b>      | Set( Película )                                                |
| <b>Multiplicidad</b> | 0..n                                                           |
| <b>Comentarios</b>   | Ninguno                                                        |

**C.5.2.9 Asociación contiene entre Cinta y Película****Descripción asociación contiene(Cinta, Película)**

|                             |                                                                                     |
|-----------------------------|-------------------------------------------------------------------------------------|
| <b>Asociación</b>           | <b>contiene entre Cinta y Película</b>                                              |
| <b>Requisitos asociados</b> | • RI-01 Información sobre películas                                                 |
| <b>Descripción</b>          | Esta asociación representa el hecho de que una cinta de vídeo contiene una película |
| <b>Comentarios</b>          | Ninguno                                                                             |

**Roles asociación contiene(Cinta, Película)**

|                      |                                                           |
|----------------------|-----------------------------------------------------------|
| <b>Rol</b>           | <b>Cinta juega rol cinta en contiene(Cinta, Película)</b> |
| <b>Descripción</b>   | Cinta que contiene una película                           |
| <b>Tipo OCL</b>      | Set( Cinta )                                              |
| <b>Multiplicidad</b> | 0..n                                                      |
| <b>Comentarios</b>   | Ninguno                                                   |

|                      |                                                                 |
|----------------------|-----------------------------------------------------------------|
| <b>Rol</b>           | <b>Película juega rol película en contiene(Cinta, Película)</b> |
| <b>Descripción</b>   | Cinta que contiene una película                                 |
| <b>Tipo OCL</b>      | Película                                                        |
| <b>Multiplicidad</b> | 0..n                                                            |
| <b>Comentarios</b>   | Ninguno                                                         |

**C.5.2.10 Asociación dirige entre Director y Película****Descripción asociación dirige(Director, Película)**

|                             |                                                                            |
|-----------------------------|----------------------------------------------------------------------------|
| <b>Asociación</b>           | <b>dirige entre Director y Película</b>                                    |
| <b>Requisitos asociados</b> | • RI-01 Información sobre películas                                        |
| <b>Descripción</b>          | Esta asociación representa el hecho de que un director dirige una película |
| <b>Comentarios</b>          | Ninguno                                                                    |

**Roles asociación dirige(Director, Película)**

|                      |                                                                  |
|----------------------|------------------------------------------------------------------|
| <b>Rol</b>           | <b>Director juega rol director en dirige(Director, Película)</b> |
| <b>Descripción</b>   | Director de una película                                         |
| <b>Tipo OCL</b>      | Director                                                         |
| <b>Multiplicidad</b> | 1                                                                |
| <b>Comentarios</b>   | Ninguno                                                          |

|                      |                                                                  |
|----------------------|------------------------------------------------------------------|
| <b>Rol</b>           | <b>Película juega rol película en dirige(Director, Película)</b> |
| <b>Descripción</b>   | Películas dirigidas por un director                              |
| <b>Tipo OCL</b>      | Set( Película )                                                  |
| <b>Multiplicidad</b> | 0..n                                                             |
| <b>Comentarios</b>   | Ninguno                                                          |

**C.5.2.11 Asociación produce entre Productora y Película****Descripción asociación produce(Productora, Película)**

|                             |                                                                                |
|-----------------------------|--------------------------------------------------------------------------------|
| <b>Asociación</b>           | <b>produce entre Productora y Película</b>                                     |
| <b>Requisitos asociados</b> | • RI-01 Información sobre películas                                            |
| <b>Descripción</b>          | Esta asociación representa el hecho de que una productora produce una película |
| <b>Comentarios</b>          | Ninguno                                                                        |

**Roles asociación produce(Productora, Película)**

|                      |                                                                     |
|----------------------|---------------------------------------------------------------------|
| <b>Rol</b>           | <b>Película juega rol película en produce(Productora, Película)</b> |
| <b>Descripción</b>   | Películas producidas por una productora                             |
| <b>Tipo OCL</b>      | Set( Película )                                                     |
| <b>Multiplicidad</b> | 0..n                                                                |
| <b>Comentarios</b>   | Ninguno                                                             |

|                      |                                                                       |
|----------------------|-----------------------------------------------------------------------|
| <b>Rol</b>           | <b>Director juega rol Productora en produce(Productora, Película)</b> |
| <b>Descripción</b>   | Productora de una película                                            |
| <b>Tipo OCL</b>      | Productora                                                            |
| <b>Multiplicidad</b> | 1                                                                     |
| <b>Comentarios</b>   | Ninguno                                                               |

**C.5.2.12 Asociación tieneAlquilada entre Película y Cinta****Descripción asociación tieneAlquilada(Película, Cinta)**

|                             |                                                                                 |
|-----------------------------|---------------------------------------------------------------------------------|
| <b>Asociación derivada</b>  | <b>tieneAlquilada entre Película y Cinta</b>                                    |
| <b>Requisitos asociados</b> | • RI-01 Información sobre películas                                             |
| <b>Descripción</b>          | Esta asociación representa el hecho de que una película tiene alquiladas cintas |
| <b>Comentarios</b>          | Ver invariante, es una asociación derivada                                      |

**Roles asociación tieneAlquilada(Película, Cinta)**

|                      |                                                                     |
|----------------------|---------------------------------------------------------------------|
| <b>Rol</b>           | <b>Cinta juega rol alquilada en tieneAlquilada(Película, Cinta)</b> |
| <b>Descripción</b>   | Cintas alquiladas                                                   |
| <b>Tipo OCL</b>      | Set( Cinta )                                                        |
| <b>Multiplicidad</b> | 0..n                                                                |
| <b>Comentarios</b>   | Ninguno                                                             |

|                      |                                                                       |
|----------------------|-----------------------------------------------------------------------|
| <b>Rol</b>           | <b>Película juega rol película en tieneAlquilada(Película, Cinta)</b> |
| <b>Descripción</b>   | Películas contenida en una cinta                                      |
| <b>Tipo OCL</b>      | Película                                                              |
| <b>Multiplicidad</b> | 1                                                                     |
| <b>Comentarios</b>   | Ninguno                                                               |

**Invariante asociación tieneAlquilada(Película, Cinta)**

|                    |                                                                                                                                                                                                                                                                                                    |
|--------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Invariante</b>  | <b>tieneAlquilada(Película, Cinta)</b>                                                                                                                                                                                                                                                             |
| <b>Descripción</b> | • Un par ( <i>película,cinta</i> ) pertenece a esta asociación si y sólo si la cinta contiene a la película y la cinta está alquilada                                                                                                                                                              |
| <b>Expresión</b>   | <pre> Película.allInstances-&gt;forall( p     Cinta.allInstances-&gt;forall( c       tieneAlquilada(Película,Cinta).allInstances-&gt;forall( ta         ( ta.película = p and ta.alquilada = c ) =       ( p.cinta-&gt;includes( c ) and c.película = p and         c.estáAlquilada ) ) ) ) </pre> |
| <b>Comentarios</b> | Ninguno                                                                                                                                                                                                                                                                                            |



### C.5.2.13 Asociación tieneDisponibile entre Película y Cinta

#### Descripción asociación tieneDisponibile(Película, Cinta)

|                             |                                                                                               |
|-----------------------------|-----------------------------------------------------------------------------------------------|
| <b>Asociación derivada</b>  | <b>tieneDisponibile entre Película y Cinta</b>                                                |
| <b>Requisitos asociados</b> | • RI-01 Información sobre películas                                                           |
| <b>Descripción</b>          | Esta asociación representa el hecho de que una película tiene disponible cintas para alquilar |
| <b>Comentarios</b>          | Ver invariante, es una asociación derivada                                                    |

#### Roles asociación tieneDisponibile(Película, Cinta)

|                      |                                                                        |
|----------------------|------------------------------------------------------------------------|
| <b>Rol</b>           | <b>Cinta juega rol disponible en tieneDisponibile(Película, Cinta)</b> |
| <b>Descripción</b>   | Cintas disponibles para alquilar                                       |
| <b>Tipo OCL</b>      | Set( Cinta )                                                           |
| <b>Multiplicidad</b> | 0..n                                                                   |
| <b>Comentarios</b>   | Ninguno                                                                |

|                      |                                                                         |
|----------------------|-------------------------------------------------------------------------|
| <b>Rol</b>           | <b>Película juega rol película en tieneDisponibile(Película, Cinta)</b> |
| <b>Descripción</b>   | Películas contenida en una cinta                                        |
| <b>Tipo OCL</b>      | Película                                                                |
| <b>Multiplicidad</b> | 1                                                                       |
| <b>Comentarios</b>   | Ninguno                                                                 |

#### Invariante asociación tieneDisponibile(Película, Cinta)

|                    |                                                                                                                                                                                                                                                                                                          |
|--------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Invariante</b>  | <b>tieneDisponibile(Película, Cinta)</b>                                                                                                                                                                                                                                                                 |
| <b>Descripción</b> | • Un par ( <i>película,cinta</i> ) pertenece a esta asociación si y sólo si la cinta contiene a la película y la cinta no está alquilada                                                                                                                                                                 |
| <b>Expresión</b>   | <pre> Película.allInstances-&gt;forall( p     Cinta.allInstances-&gt;forall( c       tieneDisponibile(Película,Cinta).allInstances-&gt;forall( td         ( td.película = p and td.alquilada = c ) =       ( p.cinta-&gt;includes( c ) and c.película = p and         not c.estáAlquilada ) ) ) ) </pre> |
| <b>Comentarios</b> | Ninguno                                                                                                                                                                                                                                                                                                  |

### C.5.2.14 Subsistema Socios y Alquileres

Este subsistema contiene los tipos y asociaciones relacionadas con los socios y alquileres del vídeo-club.

### C.5.2.15 Tipo Alquiler

#### Descripción tipo Alquiler

|                             |                                                                                                                                         |                 |              |
|-----------------------------|-----------------------------------------------------------------------------------------------------------------------------------------|-----------------|--------------|
| <b>Tipo</b>                 | <b>Alquiler</b>                                                                                                                         |                 |              |
| <b>Requisitos asociados</b> | <ul style="list-style-type: none"> <li>• RI-02 Información sobre socios</li> <li>• RI-03 Información sobre cuentas de socios</li> </ul> |                 |              |
| <b>Descripción</b>          | Este tipo concreto representa los alquileres de cintas actuales y pasados realizados por parte de los socios del vídeo-club             |                 |              |
| <b>Supertipos</b>           | –                                                                                                                                       |                 |              |
| <b>Subtipos</b>             | –                                                                                                                                       |                 |              |
| <b>Componentes</b>          | <b>Nombre</b>                                                                                                                           | <b>Tipo OCL</b> | <b>Mult.</b> |
|                             | –                                                                                                                                       | –               | –            |
| <b>Comentarios</b>          | Este tipo no se ha modelado como una asociación como tipo porque un socio puede alquilar una misma cinta más de una vez                 |                 |              |

#### Atributos tipo Alquiler

|                          |                                                   |
|--------------------------|---------------------------------------------------|
| <b>Atributo variable</b> | <b>Alquiler::estáDevuelto</b>                     |
| <b>Descripción</b>       | Indica si el socio ha devuelto la cinta alquilada |
| <b>Tipo OCL</b>          | Boolean                                           |
| <b>Valor inicial</b>     | false                                             |
| <b>Comentarios</b>       | Ninguno                                           |

|                           |                                          |
|---------------------------|------------------------------------------|
| <b>Atributo constante</b> | <b>Alquiler::fechaAlquiler</b>           |
| <b>Descripción</b>        | Fecha en la que el socio retira la cinta |
| <b>Tipo OCL</b>           | Date                                     |
| <b>Comentarios</b>        | Ninguno                                  |

|                           |                                                             |
|---------------------------|-------------------------------------------------------------|
| <b>Atributo constante</b> | <b>Alquiler::fechaDevolución</b>                            |
| <b>Descripción</b>        | Fecha tope en la que el socio <i>debe</i> devolver la cinta |
| <b>Tipo OCL</b>           | Date                                                        |
| <b>Comentarios</b>        | Ninguno                                                     |

|                           |                                         |
|---------------------------|-----------------------------------------|
| <b>Atributo constante</b> | <b>Alquiler::horaAlquiler</b>           |
| <b>Descripción</b>        | Hora en la que el socio retira la cinta |
| <b>Tipo OCL</b>           | Time                                    |
| <b>Comentarios</b>        | Ninguno                                 |

|                           |                                                            |
|---------------------------|------------------------------------------------------------|
| <b>Atributo constante</b> | <b>Alquiler::horaDevolución</b>                            |
| <b>Descripción</b>        | Hora tope en la que el socio <i>debe</i> devolver la cinta |
| <b>Tipo OCL</b>           | Time                                                       |
| <b>Comentarios</b>        | Ninguno                                                    |

### Enlaces tipo Alquiler

|                        |                                             |
|------------------------|---------------------------------------------|
| <b>Enlace variable</b> | <b>Alquiler::cargoAlquiler</b>              |
| <b>Descripción</b>     | Cargo correspondiente al alquiler           |
| <b>Tipo OCL</b>        | Alquiler                                    |
| <b>Asociación</b>      | <b>corresponde(CargoAlquiler, Alquiler)</b> |
| <b>Comentarios</b>     | Ninguno                                     |

|                        |                                                                                                           |
|------------------------|-----------------------------------------------------------------------------------------------------------|
| <b>Enlace variable</b> | <b>Alquiler::cargoMulta</b>                                                                               |
| <b>Descripción</b>     | Cargo correspondiente a la multa por devolución tardía                                                    |
| <b>Tipo OCL</b>        | Set( Alquiler ) o Alquiler                                                                                |
| <b>Asociación</b>      | <b>motivadoPor(CargoMulta, Alquiler)</b>                                                                  |
| <b>Comentarios</b>     | Los enlaces asociados a los roles con cardinalidad 0..1 puede actuar como conjuntos o como objetos en OCL |

|                         |                                    |
|-------------------------|------------------------------------|
| <b>Enlace constante</b> | <b>Alquiler::cinta</b>             |
| <b>Descripción</b>      | Cinta que es objeto del alquiler   |
| <b>Tipo OCL</b>         | Cinta                              |
| <b>Asociación</b>       | <b>esObjetoDe(Cinta, Alquiler)</b> |
| <b>Comentarios</b>      | Ninguno                            |

|                         |                                  |
|-------------------------|----------------------------------|
| <b>Enlace constante</b> | <b>Alquiler::socio</b>           |
| <b>Descripción</b>      | Cinta que es objeto del alquiler |
| <b>Tipo OCL</b>         | Socio que realiza el alquiler    |
| <b>Asociación</b>       | <b>realiza(Socio, Alquiler)</b>  |
| <b>Comentarios</b>      | Ninguno                          |

**Invariante tipo Alquiler**

| Invariante  | Alquiler                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   |
|-------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Descripción | <ul style="list-style-type: none"> <li>• No puede haber dos alquileres de la misma cinta en la misma fecha y hora</li> <li>• No puede haber dos alquileres pendientes de la misma cinta</li> <li>• La fecha y hora de devolución tienen que ser posteriores o iguales a las de alquiler</li> </ul>                                                                                                                                                                                                                                                                                                                                                         |
| Expresión   | <pre> Alquiler.allInstances-&gt;forall( a1, a2       ( a1 &lt;&gt; a2 ) = (         a1.cinta &lt;&gt; a2.cinta <b>or</b>         a1.fechaAlquiler &lt;&gt; a2.fechaAlquiler <b>or</b>         a1.horaAlquiler &lt;&gt; a2.horaAlquiler ) ) <b>and</b> Alquiler.allInstances-&gt;forall( a1, a2       ( a1 &lt;&gt; a2 <b>and</b> a1.cinta = a2.cinta ) <b>implies</b>     ( a1.estáDevuelto <b>or</b> a2.estáDevuelto ) ) <b>and</b> <b>self</b>.fechaAlquiler &lt;= <b>self</b>.fechaDevolución <b>and</b> ( <b>self</b>.fechaAlquiler = <b>self</b>.fechaDevolución ) <b>implies</b> ( <b>self</b>.horaAlquiler &lt; <b>self</b>.horaDevolución ) </pre> |
| Comentarios | Ninguno                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    |

### C.5.2.16 Tipo Cargo

#### Descripción tipo Cargo

|                             |                                                                                       |                 |              |
|-----------------------------|---------------------------------------------------------------------------------------|-----------------|--------------|
| <b>Tipo</b>                 | <b>Cargo</b>                                                                          |                 |              |
| <b>Requisitos asociados</b> | • RI-03 Información sobre cuentas de socios                                           |                 |              |
| <b>Descripción</b>          | Este tipo abstracto representa los cargos de las cuentas de los socios del vídeo-club |                 |              |
| <b>Supertipos</b>           | Movimiento                                                                            |                 |              |
| <b>Subtipos</b>             | CargoMulta, CargoAlquiler (disjuntas, completas)                                      |                 |              |
| <b>Componentes</b>          | <b>Nombre</b>                                                                         | <b>Tipo OCL</b> | <b>Mult.</b> |
|                             | –                                                                                     | –               | –            |
| <b>Comentarios</b>          | Ninguno                                                                               |                 |              |

#### Atributos tipo Cargo

|                          |                                                                                                                                    |
|--------------------------|------------------------------------------------------------------------------------------------------------------------------------|
| <b>Atributo variable</b> | <b>Cargo::estáPendiente</b>                                                                                                        |
| <b>Descripción</b>       | Indica si el cargo está pendiente de pago                                                                                          |
| <b>Tipo OCL</b>          | Boolean                                                                                                                            |
| <b>Valor inicial</b>     | true                                                                                                                               |
| <b>Comentarios</b>       | Cuando se realiza un cargo, si no hay saldo disponible para pagarlo se considera pendiente de pago hasta que haya saldo suficiente |

#### Enlaces tipo Cargo

No se han identificado enlaces para este tipo.

#### Invariante tipo Cargo

|                    |                                         |
|--------------------|-----------------------------------------|
| <b>Invariante</b>  | <b>Cargo</b>                            |
| <b>Descripción</b> | • El importe del movimiento es negativo |
| <b>Expresión</b>   | <code>self.importe &lt;= 0</code>       |
| <b>Comentarios</b> | Ninguno                                 |

**C.5.2.17 Tipo CargoAlquiler****Descripción tipo CargoAlquiler**

|                             |                                                                                                                                  |                 |              |
|-----------------------------|----------------------------------------------------------------------------------------------------------------------------------|-----------------|--------------|
| <b>Tipo</b>                 | <b>CargoAlquiler</b>                                                                                                             |                 |              |
| <b>Requisitos asociados</b> | • RI-03 Información sobre cuentas de socios                                                                                      |                 |              |
| <b>Descripción</b>          | Este tipo concreto representa los cargos de las cuentas de los socios del vídeo-club correspondientes a los alquileres de cintas |                 |              |
| <b>Supertipos</b>           | Cargo                                                                                                                            |                 |              |
| <b>Subtipos</b>             | –                                                                                                                                |                 |              |
| <b>Componentes</b>          | <b>Nombre</b>                                                                                                                    | <b>Tipo OCL</b> | <b>Mult.</b> |
|                             | –                                                                                                                                | –               | –            |
| <b>Comentarios</b>          | Ninguno                                                                                                                          |                 |              |

**Atributos tipo CargoAlquiler**

No se han identificado atributos para este tipo.

**Enlaces tipo CargoAlquiler**

|                         |                                         |
|-------------------------|-----------------------------------------|
| <b>Enlace constante</b> | <b>CargoAlquiler::alquiler</b>          |
| <b>Descripción</b>      | El alquiler al que corresponde el cargo |
| <b>Tipo OCL</b>         | Alquiler                                |
| <b>Asociación</b>       | corresponde(CargoAlquiler, Alquiler)    |
| <b>Comentarios</b>      | Ninguno                                 |

**Invariante tipo CargoAlquiler**

No se han identificado restricciones para este tipo.

**C.5.2.18 Tipo CargoMulta****Descripción tipo CargoMulta**

|                             |                                                                                                                                             |                 |              |
|-----------------------------|---------------------------------------------------------------------------------------------------------------------------------------------|-----------------|--------------|
| <b>Tipo</b>                 | <b>CargoMulta</b>                                                                                                                           |                 |              |
| <b>Requisitos asociados</b> | • RI-03 Información sobre cuentas de socios                                                                                                 |                 |              |
| <b>Descripción</b>          | Este tipo concreto representa los cargos de las cuentas de los socios del vídeo-club correspondientes a las multas por devoluciones tardías |                 |              |
| <b>Supertipos</b>           | Cargo                                                                                                                                       |                 |              |
| <b>Subtipos</b>             | –                                                                                                                                           |                 |              |
| <b>Componentes</b>          | <b>Nombre</b>                                                                                                                               | <b>Tipo OCL</b> | <b>Mult.</b> |
|                             | –                                                                                                                                           | –               | –            |
| <b>Comentarios</b>          | Ninguno                                                                                                                                     |                 |              |

**Atributos tipo CargoMulta**

No se han identificado atributos para este tipo.

**Enlaces tipo CargoMulta**

|                         |                                                                 |
|-------------------------|-----------------------------------------------------------------|
| <b>Enlace constante</b> | <b>CargoMulta::alquiler</b>                                     |
| <b>Descripción</b>      | El alquiler con devolución tardía que motiva el cargo por multa |
| <b>Tipo OCL</b>         | Alquiler                                                        |
| <b>Asociación</b>       | <b>motivadoPor(CargoMulta,Alquiler)</b>                         |
| <b>Comentarios</b>      | Ninguno                                                         |

**Invariante tipo CargoMulta**

No se han identificado restricciones para este tipo.

### C.5.2.19 Tipo Cuenta

#### Descripción tipo Cuenta

|                             |                                                                        |                        |              |
|-----------------------------|------------------------------------------------------------------------|------------------------|--------------|
| <b>Tipo</b>                 | <b>Cuenta</b>                                                          |                        |              |
| <b>Requisitos asociados</b> | • RI-03 Información sobre cuentas de socios                            |                        |              |
| <b>Descripción</b>          | Este tipo concreto representa las cuentas de los socios del vídeo-club |                        |              |
| <b>Supertipos</b>           | –                                                                      |                        |              |
| <b>Subtipos</b>             | –                                                                      |                        |              |
| <b>Componentes</b>          | <b>Nombre</b>                                                          | <b>Tipo OCL</b>        | <b>Mult.</b> |
|                             | movimiento                                                             | Sequence( Movimiento ) | 0..n         |
| <b>Comentarios</b>          | Los movimientos están ordenados por fecha                              |                        |              |

#### Atributos tipo Cuenta

|                          |                                                             |
|--------------------------|-------------------------------------------------------------|
| <b>Atributo derivado</b> | <b>Cuenta::saldo</b>                                        |
| <b>Descripción</b>       | Saldo actual de la cuenta                                   |
| <b>Expresión</b>         | <code>self.movimiento-&gt;collect( importe )-&gt;sum</code> |
| <b>Comentarios</b>       | Ninguno                                                     |

#### Enlaces tipo Cuenta

|                         |                            |
|-------------------------|----------------------------|
| <b>Enlace constante</b> | <b>Cuenta::socio</b>       |
| <b>Descripción</b>      | El socio de la cuenta      |
| <b>Tipo OCL</b>         | Socio                      |
| <b>Asociación</b>       | <b>tiene(Socio,Cuenta)</b> |
| <b>Comentarios</b>      | Ninguno                    |

#### Invariante tipo Cuenta

|                    |                                                              |
|--------------------|--------------------------------------------------------------|
| <b>Invariante</b>  | <b>Cuenta</b>                                                |
| <b>Descripción</b> | • Los movimientos están ordenados por fecha                  |
| <b>Expresión</b>   | <code>self.movimiento-&gt;isOrderedBy( &lt;=, fecha )</code> |
| <b>Comentarios</b> | Ninguno                                                      |



**C.5.2.20 Tipo Ingreso****Descripción tipo Ingreso**

|                             |                                                                                        |                 |              |
|-----------------------------|----------------------------------------------------------------------------------------|-----------------|--------------|
| <b>Tipo</b>                 | <b>Ingreso</b>                                                                         |                 |              |
| <b>Requisitos asociados</b> | • RI-03 Información sobre cuentas de socios                                            |                 |              |
| <b>Descripción</b>          | Este tipo concreto representa los ingresos de las cuentas de los socios del vídeo-club |                 |              |
| <b>Supertipos</b>           | Movimiento                                                                             |                 |              |
| <b>Subtipos</b>             | –                                                                                      |                 |              |
| <b>Componentes</b>          | <b>Nombre</b>                                                                          | <b>Tipo OCL</b> | <b>Mult.</b> |
|                             | –                                                                                      | –               | –            |
| <b>Comentarios</b>          | Ninguno                                                                                |                 |              |

**Atributos tipo Ingreso**

No se han identificado atributos para este tipo.

**Enlaces tipo Ingreso**

No se han identificado enlaces para este tipo.

**Invariante tipo Ingreso**

|                    |                                    |
|--------------------|------------------------------------|
| <b>Invariante</b>  | <b>Ingreso</b>                     |
| <b>Descripción</b> | • El importe del cargo es positivo |
| <b>Expresión</b>   | <code>self.importe &gt; 0</code>   |
| <b>Comentarios</b> | Ninguno                            |

**C.5.2.21 Tipo Movimiento****Descripción tipo Movimiento**

|                             |                                                                                            |                 |              |
|-----------------------------|--------------------------------------------------------------------------------------------|-----------------|--------------|
| <b>Tipo</b>                 | <b>Movimiento</b>                                                                          |                 |              |
| <b>Requisitos asociados</b> | • RI-03 Información sobre cuentas de socios                                                |                 |              |
| <b>Descripción</b>          | Este tipo abstracto representa los movimientos de las cuentas de los socios del vídeo-club |                 |              |
| <b>Supertipos</b>           | –                                                                                          |                 |              |
| <b>Subtipos</b>             | Ingreso, Cargo (disjuntas, completas)                                                      |                 |              |
| <b>Componentes</b>          | <b>Nombre</b>                                                                              | <b>Tipo OCL</b> | <b>Mult.</b> |
|                             | –                                                                                          | –               | –            |
| <b>Comentarios</b>          | Ninguno                                                                                    |                 |              |

**Atributos tipo Movimiento**

|                           |                                          |
|---------------------------|------------------------------------------|
| <b>Atributo constante</b> | <b>Movimiento::fecha</b>                 |
| <b>Descripción</b>        | Fecha en la que se realiza el movimiento |
| <b>Tipo OCL</b>           | <i>Date</i>                              |
| <b>Comentarios</b>        | Ninguno                                  |

|                           |                                                    |
|---------------------------|----------------------------------------------------|
| <b>Atributo constante</b> | <b>Movimiento::importe</b>                         |
| <b>Descripción</b>        | Cantidad correspondiente al importe del movimiento |
| <b>Tipo OCL</b>           | <i>Integer</i>                                     |
| <b>Comentarios</b>        | Ninguno                                            |

**Enlaces tipo Movimiento**

No se han identificado enlaces para este tipo.

**Invariante tipo Movimiento**

No se han identificado restricciones para este tipo.

### C.5.2.22 Tipo Socio

#### Descripción tipo Socio

|                             |                                                                  |                 |              |
|-----------------------------|------------------------------------------------------------------|-----------------|--------------|
| <b>Tipo</b>                 | <b>Socio</b>                                                     |                 |              |
| <b>Requisitos asociados</b> | • RI-02 Información sobre socios                                 |                 |              |
| <b>Descripción</b>          | Este tipo concreto representa los socios actuales del vídeo-club |                 |              |
| <b>Supertipos</b>           | –                                                                |                 |              |
| <b>Subtipos</b>             | –                                                                |                 |              |
| <b>Componentes</b>          | <b>Nombre</b>                                                    | <b>Tipo OCL</b> | <b>Mult.</b> |
|                             | –                                                                | –               | –            |
| <b>Comentarios</b>          | Ninguno                                                          |                 |              |

#### Atributos tipo Socio

|                          |                            |
|--------------------------|----------------------------|
| <b>Atributo variable</b> | <b>Socio::dirección</b>    |
| <b>Descripción</b>       | Dirección actual del socio |
| <b>Tipo OCL</b>          | String                     |
| <b>Comentarios</b>       | Ninguno                    |

|                           |                                                      |
|---------------------------|------------------------------------------------------|
| <b>Atributo constante</b> | <b>Socio::dni</b>                                    |
| <b>Descripción</b>        | Número del Documento Nacional de Identidad del socio |
| <b>Tipo OCL</b>           | String                                               |
| <b>Comentarios</b>        | Ninguno                                              |

|                           |                         |
|---------------------------|-------------------------|
| <b>Atributo constante</b> | <b>Socio::fechaAlta</b> |
| <b>Descripción</b>        | Fecha de alta del socio |
| <b>Tipo OCL</b>           | Date                    |
| <b>Comentarios</b>        | Ninguno                 |

|                           |                               |
|---------------------------|-------------------------------|
| <b>Atributo constante</b> | <b>Socio::fechaNacimiento</b> |
| <b>Descripción</b>        | Fecha de nacimiento del socio |
| <b>Tipo OCL</b>           | Date                          |
| <b>Comentarios</b>        | Ninguno                       |

|                           |                              |
|---------------------------|------------------------------|
| <b>Atributo constante</b> | <b>Socio::nombre</b>         |
| <b>Descripción</b>        | Nombre y apellidos del socio |
| <b>Tipo OCL</b>           | String                       |
| <b>Comentarios</b>        | Ninguno                      |

|                           |                                 |
|---------------------------|---------------------------------|
| <b>Atributo constante</b> | <b>Socio::número</b>            |
| <b>Descripción</b>        | Número identificativo del socio |
| <b>Tipo OCL</b>           | Integer                         |
| <b>Comentarios</b>        | Se debe generar automáticamente |

|                           |                                                                         |
|---------------------------|-------------------------------------------------------------------------|
| <b>Atributo constante</b> | <b>Socio::sexo</b>                                                      |
| <b>Descripción</b>        | Sexo del socio                                                          |
| <b>Tipo OCL</b>           | enum { Hombre, Mujer }                                                  |
| <b>Comentarios</b>        | Se asume que la probabilidad de que un socio cambie de sexo es muy baja |

|                           |                                                          |
|---------------------------|----------------------------------------------------------|
| <b>Atributo constante</b> | <b>Socio::teléfonos</b>                                  |
| <b>Descripción</b>        | Teléfonos del socio                                      |
| <b>Tipo OCL</b>           | Set( Integer )                                           |
| <b>Comentarios</b>        | Siempre tiene que haber al menos un teléfono de contacto |

### Enlaces tipo Socio

|                        |                                                                                                                        |
|------------------------|------------------------------------------------------------------------------------------------------------------------|
| <b>Enlace variable</b> | <b>Socio::alquiler</b>                                                                                                 |
| <b>Descripción</b>     | Conjunto de alquileres pasados y presentes del socio                                                                   |
| <b>Tipo OCL</b>        | Sequence( Alquiler )                                                                                                   |
| <b>Asociación</b>      | <b>realiza(Socio, Alquiler)</b>                                                                                        |
| <b>Comentarios</b>     | No se permiten duplicados en la secuencia de alquileres y deben estar ordenados por fecha de alquiler (ver invariante) |

|                        |                                                                 |
|------------------------|-----------------------------------------------------------------|
| <b>Enlace derivado</b> | <b>Socio::alquilerActual</b>                                    |
| <b>Descripción</b>     | Conjunto de alquileres actuales del socio                       |
| <b>Expresión</b>       | <code>self.alquiler-&gt;select( a   not a.estáDevuelto )</code> |
| <b>Asociación</b>      | <b>realizaActualmente(Socio, Alquiler)</b>                      |
| <b>Comentarios</b>     | Ninguno                                                         |

|                        |                                                        |
|------------------------|--------------------------------------------------------|
| <b>Enlace derivado</b> | <b>Socio::cinta</b>                                    |
| <b>Descripción</b>     | Conjunto de cintas actualmente alquiladas por el socio |
| <b>Expresión</b>       | <code>self.alquilerActual.cinta-&gt;asSet</code>       |
| <b>Asociación</b>      | <b>tieneAlquilada(Socio, Cinta)</b>                    |
| <b>Comentarios</b>     | Ninguno                                                |

|                         |                             |
|-------------------------|-----------------------------|
| <b>Enlace constante</b> | <b>Socio::cuenta</b>        |
| <b>Descripción</b>      | Cuenta del socio            |
| <b>Tipo OCL</b>         | Cuenta                      |
| <b>Asociación</b>       | <b>tiene(Socio, Cuenta)</b> |
| <b>Comentarios</b>      | Ninguno                     |

### Invariante tipo Socio

| Invariante  | Socio                                                                                                                                                                                                                                                                                                                                                                                                       |
|-------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Descripción | <ul style="list-style-type: none"><li>• No puede haber dos socios con el mismo DNI ni con el mismo número (dni y número son <i>clave</i>)</li><li>• Al menos debe tener un teléfono de contacto</li><li>• Los alquileres deben estar ordenados por fecha de alquiler</li><li>• No hay alquileres duplicados</li></ul>                                                                                       |
| Expresión   | <pre>Socio.allInstances-&gt;forall( s1, s2  <br/>    ( s1 &lt;&gt; s2 ) =<br/>    ( s1.dni &lt;&gt; s2.dni <b>and</b> s1.número &lt;&gt; s2.número ) )<br/><b>and</b><br/><b>not</b> ( <b>self</b>.teléfonos-&gt;isEmpty )<br/><b>and</b><br/><b>self</b>.alquiler-&gt;isOrderedBy( &lt;=, fechaAlquiler )<br/><b>and</b><br/><b>self</b>.alquiler-&gt;size = <b>self</b>.alquiler-&gt;asSet-&gt;size</pre> |
| Comentarios | Ninguno                                                                                                                                                                                                                                                                                                                                                                                                     |

**C.5.2.23 Asociación corresponde entre CargoAlquiler y Alquiler****Descripción asociación corresponde(CargoAlquiler, Alquiler)**

|                             |                                                                                                        |
|-----------------------------|--------------------------------------------------------------------------------------------------------|
| <b>Asociación</b>           | <b>corresponde</b> entre <b>CargoAlquiler</b> y <b>Alquiler</b>                                        |
| <b>Requisitos asociados</b> | <ul style="list-style-type: none"> <li>• RI-03 Información sobre cuentas de socios</li> </ul>          |
| <b>Descripción</b>          | Esta asociación representa el hecho de que un cargo por alquiler corresponde a un determinado alquiler |
| <b>Comentarios</b>          | Ninguno                                                                                                |

**Roles asociación corresponde(CargoAlquiler, Alquiler)**

|                      |                                                                                          |
|----------------------|------------------------------------------------------------------------------------------|
| <b>Rol</b>           | <b>Alquiler</b> juega rol <b>alquiler</b> en <b>corresponde(CargoAlquiler, Alquiler)</b> |
| <b>Descripción</b>   | Alquiler correspondiente al cargo                                                        |
| <b>Tipo OCL</b>      | Alquiler                                                                                 |
| <b>Multiplicidad</b> | 1                                                                                        |
| <b>Comentarios</b>   | Ninguno                                                                                  |

|                      |                                                                                                    |
|----------------------|----------------------------------------------------------------------------------------------------|
| <b>Rol</b>           | <b>CargoAlquiler</b> juega rol <b>cargoAlquiler</b> en <b>corresponde(CargoAlquiler, Alquiler)</b> |
| <b>Descripción</b>   | Cargo correspondiente al alquiler                                                                  |
| <b>Tipo OCL</b>      | CargoAlquiler                                                                                      |
| <b>Multiplicidad</b> | 1                                                                                                  |
| <b>Comentarios</b>   | Ninguno                                                                                            |

**C.5.2.24 Asociación esObjetoDe entre Cinta y Alquiler****Descripción asociación esObjetoDe(Cinta, Alquiler)**

|                             |                                                                                                                                         |
|-----------------------------|-----------------------------------------------------------------------------------------------------------------------------------------|
| <b>Asociación</b>           | <b>esObjetoDe entre Cinta y Alquiler</b>                                                                                                |
| <b>Requisitos asociados</b> | <ul style="list-style-type: none"> <li>• RI-02 Información sobre socios</li> <li>• RI-03 Información sobre cuentas de socios</li> </ul> |
| <b>Descripción</b>          | Esta asociación representa el hecho de que una cinta es objeto de alquileres                                                            |
| <b>Comentarios</b>          | Ninguno                                                                                                                                 |

**Roles asociación esObjetoDe(Cinta, Alquiler)**

|                      |                                                                   |
|----------------------|-------------------------------------------------------------------|
| <b>Rol</b>           | <b>Alquiler juega rol alquiler en esObjetoDe(Cinta, Alquiler)</b> |
| <b>Descripción</b>   | Alquileres de los que ha sido objeto una cinta                    |
| <b>Tipo OCL</b>      | Sequence( Alquiler )                                              |
| <b>Multiplicidad</b> | 0..n                                                              |
| <b>Comentarios</b>   | Los alquileres están ordenados por fecha de alquiler              |

|                      |                                                             |
|----------------------|-------------------------------------------------------------|
| <b>Rol</b>           | <b>Cinta juega rol cinta en esObjetoDe(Cinta, Alquiler)</b> |
| <b>Descripción</b>   | Cinta que es objeto del alquiler                            |
| <b>Tipo OCL</b>      | Cinta                                                       |
| <b>Multiplicidad</b> | 1                                                           |
| <b>Comentarios</b>   | Ninguno                                                     |

**C.5.2.25 Asociación esObjetoActualmenteDe entre Cinta y Alquiler****Descripción asociación esObjetoActualmenteDe(Cinta, Alquiler)**

|                             |                                                                                                                                         |
|-----------------------------|-----------------------------------------------------------------------------------------------------------------------------------------|
| <b>Asociación derivada</b>  | <b>esObjetoActualmenteDe entre Cinta y Alquiler</b>                                                                                     |
| <b>Requisitos asociados</b> | <ul style="list-style-type: none"> <li>• RI-02 Información sobre socios</li> <li>• RI-03 Información sobre cuentas de socios</li> </ul> |
| <b>Descripción</b>          | Esta asociación derivada representa el hecho de que una cinta es objeto actualmente de un alquiler                                      |
| <b>Comentarios</b>          | Ninguno                                                                                                                                 |

**Roles asociación esObjetoActualmenteDe(Cinta, Alquiler)**

|                      |                                                                                     |
|----------------------|-------------------------------------------------------------------------------------|
| <b>Rol</b>           | <b>Alquiler juega rol alquilerActual en esObjetoActualmente-De(Cinta, Alquiler)</b> |
| <b>Descripción</b>   | Alquiler actual (aún no devuelto) del que es objeto la cinta                        |
| <b>Tipo OCL</b>      | Alquiler o Set( Alquiler )                                                          |
| <b>Multiplicidad</b> | 0..1                                                                                |
| <b>Comentarios</b>   | Los roles con cardinalidad 0..1 puede actuar como conjuntos o como objetos en OCL   |

|                      |                                                                        |
|----------------------|------------------------------------------------------------------------|
| <b>Rol</b>           | <b>Cinta juega rol cinta en esObjetoActualmenteDe(Cinta, Alquiler)</b> |
| <b>Descripción</b>   | Cinta que es objeto del alquiler                                       |
| <b>Tipo OCL</b>      | Cinta                                                                  |
| <b>Multiplicidad</b> | 1                                                                      |
| <b>Comentarios</b>   | Ninguno                                                                |

**Invariante asociación esObjetoActualmenteDe(Cinta, Alquiler)**

|                    |                                                                                                                                                                                                                                                                                                                           |
|--------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Invariante</b>  | <b>esObjetoActualmenteDe(Cinta, Alquiler)</b>                                                                                                                                                                                                                                                                             |
| <b>Descripción</b> | <ul style="list-style-type: none"> <li>• Un par (<i>cinta,alquiler</i>) pertenece a esta asociación si y sólo existe un alquiler en el que la cinta es alquilada y dicho alquiler está sin devolver</li> </ul>                                                                                                            |
| <b>Expresión</b>   | <pre> Cinta.allInstances-&gt;forall( c     Alquiler.allInstances-&gt;forall( a       esObjetoActualmenteDe(Cinta, Alquiler).allInstances-&gt;       forall( eoa           ( eoa.cinta = c and eoa.alquiler = c ) =         ( a.cinta = c and c.alquiler-&gt;includes( a ) and           not a.estáDevuelto ) ) ) ) </pre> |
| <b>Comentarios</b> | Ninguno                                                                                                                                                                                                                                                                                                                   |



**C.5.2.26 Asociación motivadoPor entre CargoMulta y Alquiler****Descripción asociación motivadoPor(CargoMulta, Alquiler)**

|                             |                                                                                                                                        |
|-----------------------------|----------------------------------------------------------------------------------------------------------------------------------------|
| <b>Asociación</b>           | <b>motivadoPor entre CargoMulta y Alquiler</b>                                                                                         |
| <b>Requisitos asociados</b> | • RI-03 Información sobre cuentas de socios                                                                                            |
| <b>Descripción</b>          | Esta asociación representa el hecho de que un cargo por multa por alquiler con devolución tardía corresponde a un determinado alquiler |
| <b>Comentarios</b>          | Ninguno                                                                                                                                |

**Roles asociación motivadoPor(CargoMulta, Alquiler)**

|                      |                                                                         |
|----------------------|-------------------------------------------------------------------------|
| <b>Rol</b>           | <b>Alquiler juega rol alquiler en motivadoPor(CargoMulta, Alquiler)</b> |
| <b>Descripción</b>   | Alquiler correspondiente al cargo por multa                             |
| <b>Tipo OCL</b>      | Alquiler                                                                |
| <b>Multiplicidad</b> | 1                                                                       |
| <b>Comentarios</b>   | Ninguno                                                                 |

|                      |                                                                                   |
|----------------------|-----------------------------------------------------------------------------------|
| <b>Rol</b>           | <b>CargoMulta juega rol cargoMulta en motivadoPor(CargoMulta, Alquiler)</b>       |
| <b>Descripción</b>   | Cargo correspondiente a la multa por alquiler con devolución tardía               |
| <b>Tipo OCL</b>      | Set( CargoAlquiler ) o CargoAlquiler                                              |
| <b>Multiplicidad</b> | 0..1                                                                              |
| <b>Comentarios</b>   | Los roles con cardinalidad 0..1 puede actuar como conjuntos o como objetos en OCL |

**C.5.2.27 Asociación realiza entre Socio y Alquiler****Descripción asociación realiza(Socio, Alquiler)**

|                             |                                                                                  |
|-----------------------------|----------------------------------------------------------------------------------|
| <b>Asociación</b>           | <b>realiza entre Socio y Alquiler</b>                                            |
| <b>Requisitos asociados</b> | • RI-02 Información sobre socios                                                 |
| <b>Descripción</b>          | Esta asociación representa el hecho de que un socio realiza alquileres de cintas |
| <b>Comentarios</b>          | Ninguno                                                                          |

**Roles asociación realiza(Socio, Alquiler)**

|                      |                                                                |
|----------------------|----------------------------------------------------------------|
| <b>Rol</b>           | <b>Alquiler juega rol alquiler en realiza(Socio, Alquiler)</b> |
| <b>Descripción</b>   | Alquileres realizados por el socio                             |
| <b>Tipo OCL</b>      | Sequence( Alquiler )                                           |
| <b>Multiplicidad</b> | 0..n                                                           |
| <b>Comentarios</b>   | Los alquileres están ordenados por fecha de alquiler           |

|                      |                                                          |
|----------------------|----------------------------------------------------------|
| <b>Rol</b>           | <b>Socio juega rol socio en realiza(Socio, Alquiler)</b> |
| <b>Descripción</b>   | Socio que realiza el alquiler                            |
| <b>Tipo OCL</b>      | Socio                                                    |
| <b>Multiplicidad</b> | 1                                                        |
| <b>Comentarios</b>   | Ninguno                                                  |

**C.5.2.28 Asociación realizaActualmente entre Socio y Alquiler****Descripción asociación realizaActualmente(Socio, Alquiler)**

|                             |                                                                                                       |
|-----------------------------|-------------------------------------------------------------------------------------------------------|
| <b>Asociación derivada</b>  | <b>realizaActualmente entre Socio y Alquiler</b>                                                      |
| <b>Requisitos asociados</b> | • RI-02 Información sobre socios                                                                      |
| <b>Descripción</b>          | Esta asociación derivada representa el hecho de que un socio realiza actualmente alquileres de cintas |
| <b>Comentarios</b>          | Ninguno                                                                                               |

**Roles asociación realizaActualmente(Socio, Alquiler)**

|                      |                                                                                 |
|----------------------|---------------------------------------------------------------------------------|
| <b>Rol</b>           | <b>Alquiler juega rol alquilerActual en realizaActualmente(Socio, Alquiler)</b> |
| <b>Descripción</b>   | Alquileres realizados actualmente (aún no devueltos) por el socio               |
| <b>Tipo OCL</b>      | Set( Alquiler )                                                                 |
| <b>Multiplicidad</b> | 0..n                                                                            |
| <b>Comentarios</b>   | Ninguno                                                                         |

|                      |                                                                     |
|----------------------|---------------------------------------------------------------------|
| <b>Rol</b>           | <b>Socio juega rol socio en realizaActualmente(Socio, Alquiler)</b> |
| <b>Descripción</b>   | Socio que realiza el alquiler                                       |
| <b>Tipo OCL</b>      | Socio                                                               |
| <b>Multiplicidad</b> | 1                                                                   |
| <b>Comentarios</b>   | Ninguno                                                             |

**Invariante asociación realizaActualmente(Socio, Alquiler)**

|                    |                                                                                                                                                                                                                                                                                                                   |
|--------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Invariante</b>  | <b>realizaActualmente(Socio, Alquiler)</b>                                                                                                                                                                                                                                                                        |
| <b>Descripción</b> | • Un par ( <i>socio, alquiler</i> ) pertenece a esta asociación si y sólo si el socio tiene dicho alquiler sin devolver                                                                                                                                                                                           |
| <b>Expresión</b>   | <pre>Socio.allInstances-&gt;forall( s     Alquiler.allInstances-&gt;forall( a       realizaActualmente(Socio, Alquiler).allInstances-&gt;       forall( ra           ( ra.socio = s and ra.alquiler = a ) =         ( s.alquiler-&gt;includes( a ) and a.socio = s and           not a.estáDevuelto ) ) ) )</pre> |
| <b>Comentarios</b> | Ninguno                                                                                                                                                                                                                                                                                                           |

**C.5.2.29 Asociación tiene entre Socio y Cuenta****Descripción asociación tiene(Socio, Cuenta)**

|                             |                                                                                       |
|-----------------------------|---------------------------------------------------------------------------------------|
| <b>Asociación</b>           | <b>tiene entre Socio y Cuenta</b>                                                     |
| <b>Requisitos asociados</b> | • RI-03 Información sobre cuentas de socios                                           |
| <b>Descripción</b>          | Esta asociación representa el hecho de que un socio tiene una cuenta en el vídeo-club |
| <b>Comentarios</b>          | Ninguno                                                                               |

**Roles asociación tiene(Socio, Cuenta)**

|                      |                                                        |
|----------------------|--------------------------------------------------------|
| <b>Rol</b>           | <b>Cuenta juega rol cuenta en tiene(Socio, Cuenta)</b> |
| <b>Descripción</b>   | Cuenta del socio                                       |
| <b>Tipo OCL</b>      | Cuenta                                                 |
| <b>Multiplicidad</b> | 1                                                      |
| <b>Comentarios</b>   | Ninguno                                                |

|                      |                                                      |
|----------------------|------------------------------------------------------|
| <b>Rol</b>           | <b>Socio juega rol socio en tiene(Socio, Cuenta)</b> |
| <b>Descripción</b>   | Socio titular de la cuenta                           |
| <b>Tipo OCL</b>      | Cuenta                                               |
| <b>Multiplicidad</b> | 1                                                    |
| <b>Comentarios</b>   | Ninguno                                              |

### C.5.2.30 Asociación tieneAlquilada entre Socio y Cinta

#### Descripción asociación tieneAlquilada(Socio, Cinta)

|                             |                                                                                                                 |
|-----------------------------|-----------------------------------------------------------------------------------------------------------------|
| <b>Asociación derivada</b>  | <b>tieneAlquilada entre Socio y Cinta</b>                                                                       |
| <b>Requisitos asociados</b> | • RI-02 Información sobre socios                                                                                |
| <b>Descripción</b>          | Esta asociación derivada representa el hecho de que un socio tiene actualmente cintas alquiladas (sin devolver) |
| <b>Comentarios</b>          | Ninguno                                                                                                         |

#### Roles asociación tieneAlquilada(Socio, Cinta)

|                      |                                                               |
|----------------------|---------------------------------------------------------------|
| <b>Rol</b>           | <b>Cinta juega rol cinta en tieneAlquilada(Socio, Cinta)</b>  |
| <b>Descripción</b>   | Cintas alquiladas actualmente (aún no devueltas) por un socio |
| <b>Tipo OCL</b>      | Set( Cinta )                                                  |
| <b>Multiplicidad</b> | 0..n                                                          |
| <b>Comentarios</b>   | Ninguno                                                       |

|                      |                                                                                   |
|----------------------|-----------------------------------------------------------------------------------|
| <b>Rol</b>           | <b>Socio juega rol socio en tieneAlquilada(Socio, Cinta)</b>                      |
| <b>Descripción</b>   | Socio que tiene alquilada la cinta                                                |
| <b>Tipo OCL</b>      | Set( Cinta ) o Cinta                                                              |
| <b>Multiplicidad</b> | 0..1                                                                              |
| <b>Comentarios</b>   | Los roles con cardinalidad 0..1 puede actuar como conjuntos o como objetos en OCL |

#### Invariante asociación tieneAlquilada(Socio, Cinta)

|                    |                                                                                                                                                                                                                                                                                             |
|--------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Invariante</b>  | <b>tieneAlquilada(Socio, Cinta)</b>                                                                                                                                                                                                                                                         |
| <b>Descripción</b> | • Un par ( <i>socio, cinta</i> ) pertenece a esta asociación si y sólo existe un alquiler no devuelto realizado por el socio en el que la cinta es alquilada                                                                                                                                |
| <b>Expresión</b>   | <pre>Socio.allInstances-&gt;forall( s     Cinta.allInstances-&gt;forall( c       tieneAlquilada(Socio, Cinta).allInstances-&gt;forall( ta         ( ta.socio = s and ta.cinta = c ) =       ( s.alquilerActual.cinta-&gt;includes( c ) and         c.alquilerActual.socio = s ) ) ) )</pre> |
| <b>Comentarios</b> | Ninguno                                                                                                                                                                                                                                                                                     |

## C.5.3 Operaciones del sistema

### C.5.3.1 Operación AltaDeSocio

#### Descripción operación AltaDeSocio

|                              |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         |
|------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Operación Sistema</b>     | <b>AltaDeSocio</b>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |
| <b>Requisitos asociados</b>  | • RF-01 Alta de socio                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   |
| <b>Descripción</b>           | El empleado del vídeo-club da de alta un nuevo socio                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    |
| <b>Tipo resultado</b>        | Socio -- para imprimir el carnet                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        |
| <b>Parámetros</b>            | dni : String -- DNI del nuevo socio<br>n : String -- nombre y apellidos del nuevo socio<br>fn : Date -- fecha nacimiento del nuevo socio<br>sx : enum{ Hombre, Mujer } -- sexo del nuevo socio<br>d : String -- dirección del nuevo socio<br>t : Set( Integer ) -- teléfonos del nuevo socio                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |
| <b>Precondiciones</b>        | <b>pre<sub>1</sub></b> : No existe en el sistema un socio con el mismo número de DNI                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    |
| <b>Precondiciones (OCL)</b>  | <b>pre<sub>1</sub></b> : <code>not Socio.allInstaces-&gt;exist( s   s.dni = dni )</code>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |
| <b>Postcondiciones</b>       | <b>post<sub>1</sub></b> : Existe un nuevo socio cuyos atributos coinciden con los parámetros, que no tiene ningún alquiler y que tiene una nueva cuenta sin movimientos<br><b>post<sub>2</sub></b> : Sólo se ha creado un nuevo socio y una nueva cuenta<br><b>post<sub>3</sub></b> : El sistema informa de que el proceso ha terminado con éxito<br><b>post<sub>4</sub></b> : El resultado de la operación es el nuevo socio (para imprimir el carnet)                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |
| <b>Postcondiciones (OCL)</b> | <b>post<sub>1</sub></b> : <code>Socio.new-&gt;exists( s  <br/>               s.dni = dni                      <b>and</b><br/>               s.nombre = n                   <b>and</b><br/>               s.fechaNacimiento = fn       <b>and</b><br/>               s.sexo = sx                    <b>and</b><br/>               s.fechaAlta = Date.today   <b>and</b><br/>               s.dirección = d               <b>and</b><br/>               s.teléfonos = t               <b>and</b><br/>               s.alquiler-&gt;isEmpty           <b>and</b><br/>               Cuenta.new-&gt;exists( c  <br/>                   (s.cuenta = c) <b>and</b><br/>                   (c.socio = s) <b>and</b><br/>                   c.movimiento-&gt;isEmpty ) )<br/> <b>post<sub>2</sub></b>: <code>Socio.new-&gt;size = 1 <b>and</b> Cuenta.new-&gt;size = 1</code><br/> <b>post<sub>3</sub></b>: <code>respuesta = Set{"Proceso de alta de socio terminado con éxito"}</code><br/> <b>post<sub>4</sub></b>: <code>Socio.new-&gt;includes( resultado )</code> </code> |
| <b>Excepciones</b>           | <b>pre<sub>1</sub></b> : El sistema informa de que el socio ya está registrado                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          |
| <b>Excepciones (OCL)</b>     | <b>not pre<sub>1</sub></b> : <code>respuesta-&gt;includes( "Error: socio ya registrado" )</code>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        |
| <b>Comentarios</b>           | Ninguno                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |

### Diagrama de acción conjunta de AltaDeSocio



Figura C.8: Diagrama de acción conjunta de *AltaDeSocio*

### Diagrama de secuencia de AltaDeSocio



Figura C.9: Diagrama de secuencia de *AltaDeSocio*

**Interfaz de usuario de AltaDeSocio**

**Alta de Socio**

Nº del socio  Fecha alta

Datos personales

DNI  Apellidos, Nombre

Fecha nacimiento  Sexo

Dirección

Teléfonos

**Vídeo--Club Ejemplo** Núm: 23456

-----

*dirección y teléfono del vídeo--club*

**CARNET DE SOCIO**

**Nombre:** Amador Durán Toro  
**Dirección:** Departamento de Lenguajes y Sistemas Informáticos, Avda. Reina Mercedes s/n, 41012 Sevilla

**DNI:** 12345678-X **Firma**  
**Fecha alta:** 21/12/1998

Figura C.10: Interfaz de usuario de *AltaDeSocio*

En el diálogo, el empleado del vídeo-club introduce los datos del nuevo socio. El número del socio y la fecha de alta son proporcionados por el sistema. Una vez que se finaliza pulsando el botón OK el sistema imprime el carnet de socio.



### C.5.3.2 Operación AlquilarCinta

#### Descripción operación AlquilarCinta

|                              |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    |
|------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Operación Sistema</b>     | <b>AlquilarCinta</b>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               |
| <b>Requisitos asociados</b>  | • RF-06 Alquiler de cintas de vídeo                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |
| <b>Descripción</b>           | Un socio alquila una cinta de vídeo                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |
| <b>Tipo resultado</b>        | Alquiler -- para imprimir el ticket                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |
| <b>Parámetros</b>            | s : Socio -- socio que realiza el alquiler<br>c : Cinta -- cinta que se alquila<br>imp : Integer -- importe del alquiler<br>fd : Date -- fecha de devolución<br>hd : Time -- hora de devolución<br>fp : enum{ Contado, Cuenta } -- forma de pago                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   |
| <b>Precondiciones</b>        | <b>pre<sub>1</sub></b> : La cinta no está registrada como alquilada                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |
| <b>Precondiciones (OCL)</b>  | <b>pre<sub>1</sub></b> : <code>not c.estáAlquilada@pre</code>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |
| <b>Postcondiciones</b>       | <b>post<sub>1</sub></b> : Existe un nuevo alquiler no devuelto correspondiente al socio y a la cinta y existe también el cargo correspondiente en la cuenta del socio<br><b>post<sub>2</sub></b> : Sólo se ha creado un nuevo alquiler y un nuevo cargo<br><b>post<sub>3</sub></b> : Si se paga al contado se crea el ingreso oportuno en la cuenta del socio<br><b>post<sub>3</sub></b> : El sistema informa de que el proceso ha terminado con éxito<br><b>post<sub>4</sub></b> : El resultado de la operación es el nuevo alquiler (para imprimir el ticket)                                                                                                                                                                                                                                                                                                                                                    |
| <b>Postcondiciones (OCL)</b> | <b>post<sub>1</sub></b> : <code>Alquiler.new-&gt;exists( a  <br/> (not a.estáDevuelto) and<br/> a.socio = s and a.cinta = c and<br/> a.fechaAlquiler = Date.today and<br/> a.horaAlquiler = Time.now and<br/> a.fechaDevolución = fd and<br/> a.horaDevolución = hd and<br/> CargoAlquiler.new-&gt;exists( ca  <br/> ca.cuenta = s.cuenta and ca.alquiler = a and<br/> ca.importe = -imp and<br/> ca.fecha = Date.today ) )</code><br><b>post<sub>2</sub></b> : <code>Alquiler.new-&gt;size = 1 and<br/> CargoAlquiler.new-&gt;size = 1</code><br><b>post<sub>3</sub></b> : <code>( fp = #Contado ) = Ingreso.new-&gt;exists( i  <br/> i.cuenta = s.cuenta and i.fecha = Date.today and<br/> i.importe = imp )</code><br><b>post<sub>4</sub></b> : <code>respuesta = Set{ "Proceso de alquiler terminado<br/> con éxito" }</code><br><b>post<sub>5</sub></b> : <code>Alquiler.new-&gt;includes( resultado )</code> |
| <b>Excepciones</b>           | <b>pre<sub>1</sub></b> : El sistema informa de que la cinta está registrada como alquilada                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         |
| <b>Excepciones (OCL)</b>     | <code>not pre<sub>1</sub>: respuesta-&gt;includes( "Error: la cinta está<br/> registrada como alquilada" )</code>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |
| <b>Comentarios</b>           | Ninguno                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |

### Diagrama de acción conjunta de AlquilerCinta

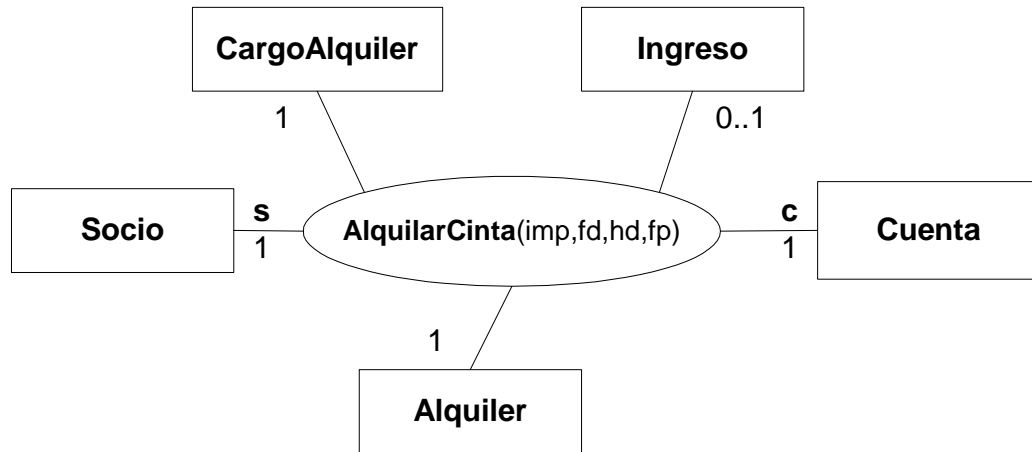


Figura C.11: Diagrama de acción conjunta de *AlquilerCinta*

### Diagrama de secuencia de AlquilerCinta

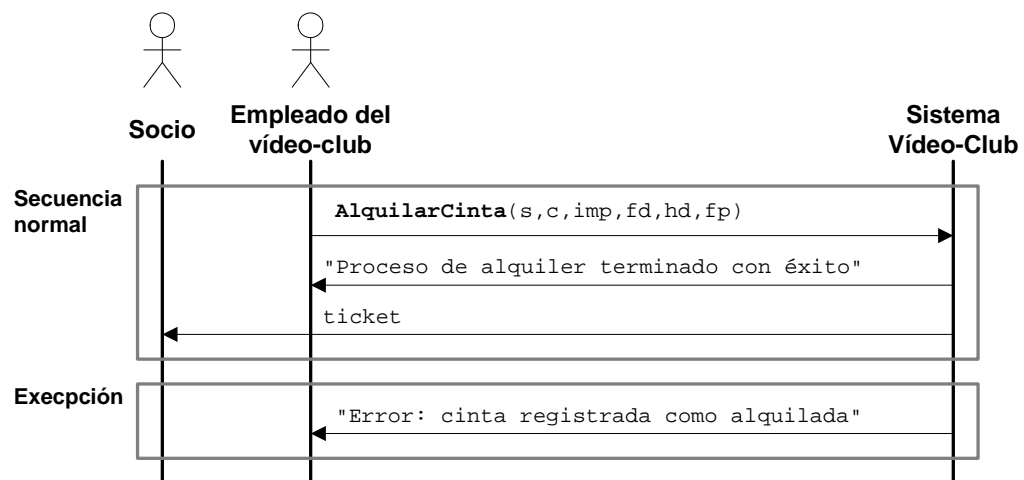


Figura C.12: Diagrama de secuencia de *AlquilerCinta*

### Interfaz de usuario de AlquilerCinta

**Alquiler de cinta**

Nº del socio: 2345

Nombre: Durán Toro, Amador Saldo: 1000

Película: La guerra de las galaxias Cinta: 1234

Fecha alquiler: 21/12/1998 Importe: 300

Hora alquiler: 18:05:32

Fecha devolución: 22/12/1998

Hora devolución: 21:30:00

Forma de pago:

☒ A cuenta

☐ Al contado

OK Cancel

**Vídeo--Club Ejemplo**

-----

dirección y teléfono del vídeo--club

**TICKET DE ALQUILER**

Socio: 23456 (Amador Durán Toro)

Cinta: 1234 (La guerra de las galaxias)

Fecha alquiler: 21/12/1998 18:05:32

Devolver antes de: 22/12/1998 21:30:00

Son 300 Ptas

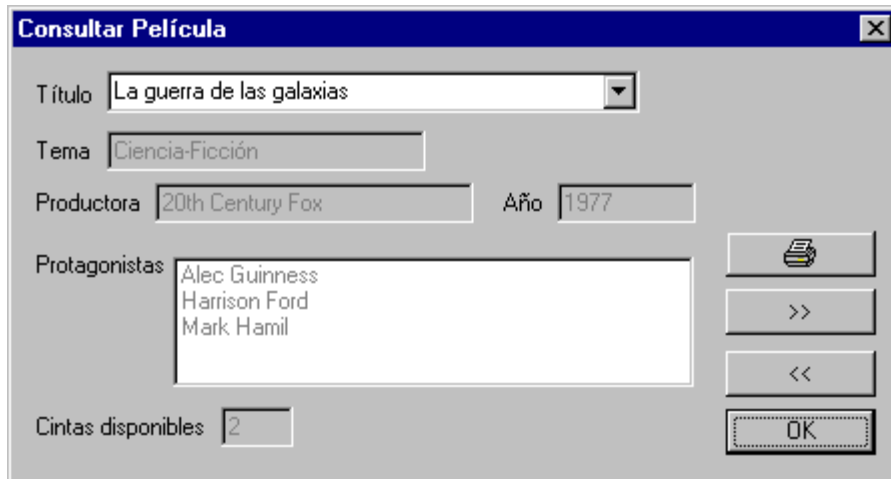
Figura C.13: Interfaz de usuario de *AlquilarCinta*

En el diálogo, el empleado del vídeo-club introduce el número del socio que realiza el alquiler y selecciona el nombre de la película. El sistema le muestra el nombre y el saldo del socio y le propone una cinta disponible (si hay) de la película. El sistema propone la fecha y hora de devolución así como el importe del alquiler. El empleado puede aceptar los valores propuestos o cambiarlos. Por último el empleado indica la forma de pago. Una vez que se finaliza pulsando el botón OK el sistema imprime el ticket.

### C.5.3.3 Operación ConsultarPelícula

#### Descripción operación ConsultarPelícula

|                              |                                                                                                                                                                                                                                                                                                                                                                                                     |
|------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Operación Sistema</b>     | <b>ConsultarPelícula</b>                                                                                                                                                                                                                                                                                                                                                                            |
| <b>Requisitos asociados</b>  | • RF-10 Consulta de una película                                                                                                                                                                                                                                                                                                                                                                    |
| <b>Descripción</b>           | El empleado del vídeo-club consulta la información de una película                                                                                                                                                                                                                                                                                                                                  |
| <b>Tipo resultado</b>        | titulo:String ×<br>tema:enum{Infantil,Acción,Ciencia-Ficción,Terror,Adultos} ×<br>productora:String ×<br>protagonistas:Set(String) ×<br>disponible:Integer                                                                                                                                                                                                                                          |
| <b>Parámetros</b>            | <i>p : Película -- la película a consultar</i>                                                                                                                                                                                                                                                                                                                                                      |
| <b>Precondiciones</b>        | <b>pre<sub>1</sub></b> : Ninguna                                                                                                                                                                                                                                                                                                                                                                    |
| <b>Precondiciones (OCL)</b>  | <b>pre<sub>1</sub></b> : true                                                                                                                                                                                                                                                                                                                                                                       |
| <b>Postcondiciones</b>       | <b>post<sub>1</sub></b> : El resultado coincide con la información deseada<br><b>post<sub>2</sub></b> : El sistema informa de que el proceso ha terminado con éxito                                                                                                                                                                                                                                 |
| <b>Postcondiciones (OCL)</b> | <b>post<sub>1</sub></b> : <b>resultado</b> .titulo = p.titulo and<br><b>resultado</b> .tema = p.tema and<br><b>resultado</b> .productora = p.productora.nombre and<br><b>resultado</b> .protagonistas =<br>p.protagonistas.nombre->asSet and<br><b>resultado</b> .disponibles = p.disponible->size<br><b>post<sub>2</sub></b> : <b>respuesta</b> = Set{ "Proceso de consulta terminado con éxito" } |
| <b>Comentarios</b>           | Ninguno                                                                                                                                                                                                                                                                                                                                                                                             |

**Interfaz de usuario operación ConsultarPelícula**

**Consultar Película**

Título: La guerra de las galaxias

Tema: Ciencia-Ficción

Productora: 20th Century Fox Año: 1977

Protagonistas: Alec Guinness, Harrison Ford, Mark Hamill

Cintas disponibles: 2

Buttons: Print, >>, <<, OK

```
Consulta de película 21/12/1998

Título: La Guerra de las Galaxias
Tema: Ciencia-Ficción
Productora: 20th Century Fox (1977)
Protagonistas:
 Mark Hamill
 Harrison Ford
 Alec Guinness
Cintas disponibles: 2
```

Figura C.14: Interfaz de usuario del operación *ConsultarPelícula*

En el diálogo, el empleado del vídeo-club introduce el nombre de la película que quiere consultar y el sistema le muestra la información asociada. El empleado puede avanzar o retroceder en orden alfabético por la películas utilizando los botones » y «. Si lo desea, puede imprimir los datos de la película pulsando el botón con una imagen de impresora. La consulta finaliza una vez que se pulsa el botón OK.

### C.5.4 Conflictos detectados en los requisitos–C

Como resultado del análisis realizado, se deberían realizar los siguientes cambios en los requisitos–C originales:

|                                 |                                                                                                                                                                                                                                                                                                                     |
|---------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>CFL–1</b>                    | <b>Necesidad de un identificador de cintas</b>                                                                                                                                                                                                                                                                      |
| <b>Objs./Reqs. en conflicto</b> | <ul style="list-style-type: none"> <li>• RI-01 <i>Información sobre películas</i></li> <li>• RF-05 <i>Alta de cinta de vídeo</i></li> </ul>                                                                                                                                                                         |
| <b>Descripción</b>              | Es necesario que cada cinta tenga un identificador (un número, una combinación de letras y números, ...), para que el empleado del vídeo-club las pueda localizar en las estanterías                                                                                                                                |
| <b>Alternativas</b>             | <ul style="list-style-type: none"> <li>• Crear un nuevo requisito para registrar específicamente la necesidad de que el sistema genere y conozca el identificador de cada cinta (–)</li> <li>• Añadir al requisito RI–01 un nuevo dato específico que recoja la necesidad del identificador de cinta (–)</li> </ul> |
| <b>Solución</b>                 | Crear un nuevo requisito RI–04 <i>Información sobre cintas</i> que recoja la necesidad del identificador                                                                                                                                                                                                            |
| <b>Comentarios</b>              | Ninguno                                                                                                                                                                                                                                                                                                             |

|                                 |                                                                                                                                                                                                                                                                 |
|---------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>CFL–2</b>                    | <b>Necesidad de especificar la información sobre alquileres</b>                                                                                                                                                                                                 |
| <b>Objs./Reqs. en conflicto</b> | <ul style="list-style-type: none"> <li>• RI-02 <i>Información sobre socios</i></li> <li>• RI-03 <i>Información sobre cuentas de socios</i></li> <li>• RF-06 <i>Alquiler de cintas de vídeo</i></li> <li>• RF-06 <i>Devolución de cintas de vídeo</i></li> </ul> |
| <b>Descripción</b>              | No se describe en ningún requisito la información que se debe almacenar sobre los alquileres de cintas                                                                                                                                                          |
| <b>Alternativas</b>             | <ul style="list-style-type: none"> <li>• Crear un nuevo requisito de almacenamiento de información para registrar específicamente la información que se debe almacenar sobre los alquileres (–)</li> </ul>                                                      |
| <b>Solución</b>                 | Crear un nuevo requisito RI–05 <i>Información sobre alquileres</i> que describa la información a almacenar sobre los alquileres                                                                                                                                 |
| <b>Comentarios</b>              | Ninguno                                                                                                                                                                                                                                                         |

|                                 |                                                                                                                                                                                   |
|---------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>CFL–3</b>                    | <b>Especificar que un pago en metálico es un ingreso</b>                                                                                                                          |
| <b>Objs./Reqs. en conflicto</b> | <ul style="list-style-type: none"> <li>• RI-03 <i>Información sobre cuentas de socios</i></li> <li>• RF-06 <i>Alquiler de cintas de vídeo</i></li> </ul>                          |
| <b>Descripción</b>              | No queda claro que cuando el socio paga en metálico se considera un ingreso en su cuenta, además el intervalo temporal indica sólo presente                                       |
| <b>Alternativas</b>             | <ul style="list-style-type: none"> <li>• Modificar el requisito RI–03 para aclarar el concepto de ingreso y para cambiar el intervalo temporal a pasado y presente (–)</li> </ul> |
| <b>Solución</b>                 | Modificar el requisito RI–03 y dejar también claro el concepto de <i>pago pendiente</i> , definiéndolo como aquellos cargos para los que no hay saldo suficiente                  |
| <b>Comentarios</b>              | Ninguno                                                                                                                                                                           |

## C.6 Bibliografía

- [Booch *et al.* 1999] G. Booch, J. Rumbaugh, y I. Jacobson. *The Unified Modeling Language User Guide*. Addison–Wesley, 1999.
- [D’Souza y Wills 1999] D. F. D’Souza y A. C. Wills. *Objects, Components, and Frameworks with UML: The Catalysis Approach*. Addison–Wesley, 1999.
- [Rat 1997] Rational Software Corporation. *Object Constraint Language Specification*, 1.1 edición, Septiembre 1997. Disponible en <http://www.rational.com>.
- [Warmer y Kleppe 1999] J. B. Warmer y A. G. Kleppe. *The Object Constraint Language: Precise Modeling with UML*. Addison–Wesley, 1999.





## Apéndice D

# Metodología para la validación de requisitos de sistemas de información

### D.1 Objetivo de la metodología

El objetivo de esta metodología es la definición de las tareas a realizar, los productos a obtener y las técnicas a emplear durante la actividad de *validación de requisitos* de la fase de *ingeniería de requisitos* del ciclo de vida de desarrollo del software.

En esta metodología se distinguen dos tipos de productos: los productos *entregables* y los productos *no entregables* o *internos*. Los productos entregables son aquellos que se entregan oficialmente al cliente como parte del desarrollo en fechas previamente acordadas, mientras que los no entregables son productos internos al desarrollo que no se entregan al cliente.

El principal producto entregable de esta metodología es la versión validada del *Documento de Requisitos del Sistema*, descrito en la *Metodología para la Elicitación de Requisitos de Sistemas de Información*. Opcionalmente, pueden incluirse también las versiones validadas del *Documento de Análisis del Sistema*, descrito en la *Metodología para el Análisis de Requisitos de Sistemas de Información*, y del prototipo en el caso de que se haya desarrollado.

La estructura de este documento es la siguiente: en la sección D.2 se describen las tareas recomendadas para realizar la validación, en la sección D.3 se definen los productos entregables, y por último, en la sección D.4 se describen las técnicas recomendadas para obtener los productos.

## **D.2 Tareas recomendadas**

Las tareas recomendadas para obtener los productos descritos en esta metodología son las siguientes:

**Tarea 1:** Validar los requisitos de almacenamiento de información y funcionales

**Tarea 2:** Validar los requisitos no funcionales

**Tarea 3:** Cerrar la versión de los requisitos

El orden recomendado de realización para estas tareas es: 1...3, aunque las tareas 1 y 2 pueden realizarse simultáneamente y el tipo de sistema puede determinar un orden u otro.

En las siguientes secciones se describen cada una de las tareas mencionadas.

### **D.2.1 Tarea 1: Validar los requisitos de almacenamiento de información y funcionales**

#### **D.2.1.1 Objetivos**

- Asegurar que los requisitos de almacenamiento de información y los requisitos funcionales representan las necesidades de los clientes y usuarios
- Identificar posibles conflictos en los requisitos de almacenamiento de información y los requisitos funcionales que no se hayan detectado durante la actividad de análisis de requisitos

#### **D.2.1.2 Descripción**

En esta tarea se deben validar los requisitos de almacenamiento de información y los requisitos funcionales para asegurar, en la medida de lo posible, que representan realmente las necesidades de los clientes y usuarios y que describen el producto deseado. De no ser así, se deben identificar los posibles conflictos que impiden la validación de estos tipos de requisitos.

### D.2.1.3 Productos internos

No hay productos internos en esta tarea.

### D.2.1.4 Productos entregables

- Requisitos de almacenamiento de información y requisitos funcionales validados, total o parcialmente, como parte del *Documento de Requisitos del Sistema* (DRS) validado
- Catálogo de tipos y asociaciones y catálogo de operaciones del sistema validados, total o parcialmente, como parte del *Documento de Análisis del Sistema* (DAS) validado, en el caso de que se haya considerado necesaria su validación por parte de clientes y usuarios
- Conflictos pendientes de resolución como parte del DRS, como parte del DAS o como parte de un documento específico de conflictos

### D.2.1.5 Técnicas

- Walkthroughs asistidos con prototipos
- Plantillas para conflictos

## D.2.2 Tarea 2: Validar los requisitos no funcionales

### D.2.2.1 Objetivos

- Asegurar que los requisitos no funcionales representan las necesidades de los clientes y usuarios
- Identificar posibles conflictos en los requisitos no funcionales que no se hayan detectado durante la actividad de análisis de requisitos

### D.2.2.2 Descripción

En esta tarea se deben validar los requisitos no funcionales para asegurar, en la medida de lo posible, que representan realmente las necesidades de los clientes y usuarios y que describen el producto deseado. De no ser así, se deben identificar los posibles conflictos que impiden la validación de este tipo de requisitos.

#### **D.2.2.3 Productos internos**

No hay productos internos en esta tarea.

#### **D.2.2.4 Productos entregables**

- Requisitos no funcionales validados, total o parcialmente, como parte del *Documento de Requisitos del Sistema* (DRS) validado
- Conflictos pendientes de resolución como parte del DRS, como parte del DAS o como parte de un documento específico de conflictos

#### **D.2.2.5 Técnicas**

- Revisión por parte de clientes y usuarios con el asesoramiento de los ingenieros de requisitos o de otros participantes
- Plantillas para conflictos

### **D.2.3 Tarea 3: Cerrar la versión de los requisitos**

#### **D.2.3.1 Objetivos**

- Establecer un hito en el desarrollo

#### **D.2.3.2 Descripción**

En el caso de que no queden conflictos pendientes de resolución, o que se acuerde pasar a fases posteriores del desarrollo, se deben cerrar las versiones de los productos obtenidos para establecer un hito en el desarrollo, sin por ello dejar de tener en cuenta que probablemente serán necesarias futuras iteraciones del proceso a la luz de los resultados que se vayan obteniendo al avanzar en el desarrollo.

#### **D.2.3.3 Productos internos**

No hay productos internos en esta tarea.

#### D.2.3.4 Productos entregables

- Versión cerrada del DRS validado
- Versión cerrada del DAS validado, en el caso de que se haya considerado necesaria su validación por parte de clientes y usuarios
- Versión cerrada del prototipo, en el caso de que se haya desarrollado

#### D.2.3.5 Técnicas

No se ha considerado ninguna técnica específica para cerrar las versiones de los productos obtenidos.

Para evitar posibles interpretaciones erróneas de lo que puede significar llegar a un acuerdo y cerrar una versión de los requisitos, puede incluirse un texto similar al siguiente en los documentos de requisitos validados o en el texto del acuerdo al que se llegue, tal como se propone en [Wiegers 1999]:

*Los abajo firmantes acordamos que este documento representa nuestro conocimiento actual de los requisitos del proyecto al día de hoy. Los futuros cambios de este documento se realizarán de acuerdo al procedimiento de cambio definido para el proyecto. Entendemos que los cambios aprobados podrán requerir que se renegocien los acuerdos actuales sobre costes, recursos asignados y fecha de entrega del proyecto.*

### D.3 Productos entregables

Los productos entregables resultantes de las tareas descritas son los productos resultantes de las actividades anteriores validados total o parcialmente, por lo que pueden consultarse las metodologías correspondientes a las actividades de elicitación y análisis para obtener su descripción.

### D.4 Técnicas

A continuación, se describen algunas de las técnicas que se proponen en esta metodología para obtener los productos de las tareas que se han descrito.

### **D.4.1 Walkthrough**

**Nota:** Esta sección coincide con la sección 5.4.1, pág. 200, por lo que se ha omitido para evitar repeticiones innecesarias.

## **D.5 Bibliografía**

[Wiegers 1999] K. Wiegers. Customer Rights and Responsibilities. *Software Development*, Diciembre 1999. Disponible en <http://www.sdmagazine.com/breakrm/features/s9912f2.shtml>.

## Apéndice E

# Una herramienta CASE para la ingeniería de requisitos

### E.1 Introducción

En este apéndice se describe un prototipo de una posible herramienta CASE que dé soporte a la metodología de ingeniería de requisitos de sistemas de información descrita en esta tesis.

Sin una herramienta CASE apropiada, la aplicación de cualquier metodología o técnica de desarrollo de software suele fracasar por tener que realizarse manualmente, tal como se describe en [Davis 1995] en lo relativo a la rastreabilidad.

El prototipo que se presenta es sólo una primera aproximación que probablemente, tal como se expone en [Brooks 1995, cap. 11 *Plan to Throw One Away (You Will, Anyhow)*], servirá para obtener experiencia sobre el problema pero que deberá abandonarse posteriormente para construir otro mejor.

La estructura de este apéndice es la siguiente: en la sección E.2 se describe brevemente el modelo de objetos de los requisitos que se ha utilizado para la realización del prototipo. En las secciones E.3 y E.4 se describen la arquitectura y los aspectos más relevantes de la interfaz de usuario del prototipo. En la sección E.5 se comparan las características de otras herramientas CASE comerciales de gestión de requisitos.

## E.2 Modelo de los requisitos

El modelo que se ha utilizado para la construcción del prototipo está basado en la estructura de las plantillas descritas en los capítulos 3 y 4.

El modelo se ha organizado en los cuatro paquetes que pueden verse en la figura E.1, de forma que los aspectos comunes se han dejado dentro del paquete denominado Requirements Project, y los aspectos específicos de los requisitos–C, requisitos–D y conflictos se han organizado en los subsistemas del mismo nombre.

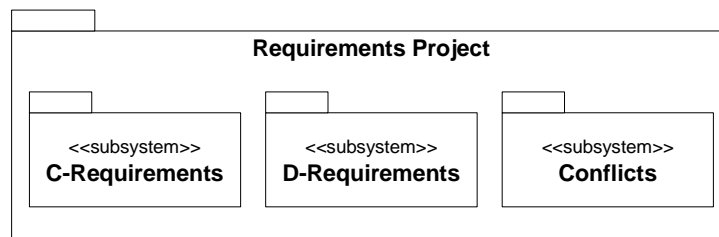


Figura E.1: Subsistemas del modelo de requisitos

En las siguientes secciones se describen cada uno de los paquetes de la figura E.1. Aunque podría haberse reducido el número de tipos de objetos creando tipos abstractos que tuvieran los atributos y asociaciones comunes y reorganizando las jerarquías de clasificación, se ha optado por no hacerlo para facilitar la comprensión de los modelos.

### E.2.1 Modelo de un proyecto de ingeniería de requisitos

Tal como puede verse en el diagrama de tipos de la figura E.2, un proyecto de ingeniería de requisitos se considera conceptualmente compuesto de una especificación o documento de requisitos–C (CRS), de una especificación o documento de requisitos–D (DRS) y de una especificación o documento de conflictos (CFS).

Estas especificaciones no tienen porqué coincidir con documentos físicos, simplemente son instrumentos para organizar la información de forma abstracta.

También se consideran como componentes del proyecto de ingeniería de requisitos la información correspondiente a los participantes y a las organizaciones para las que éstos trabajan, que pueden jugar el papel de desarrolladoras o clientes en el proyecto.



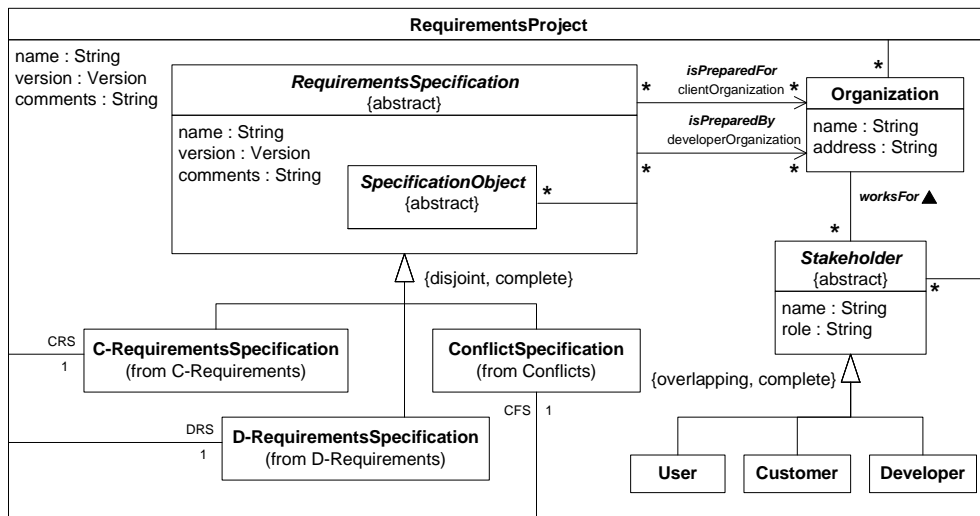


Figura E.2: Estructura de los proyectos de ingeniería de requisitos

De forma general, una especificación abstracta de requisitos contiene objetos abstractos de especificación, que pueden clasificarse en objetos de especificación más concretos, tal como se muestra en la figura E.3.

Los objetos de especificación pueden tener relaciones de rastreabilidad entre ellos y hacia aquellos participantes que sean sus autores y, en el caso de los requisitos-*C*, actores y conflictos, hacia aquellos que sean sus fuentes.

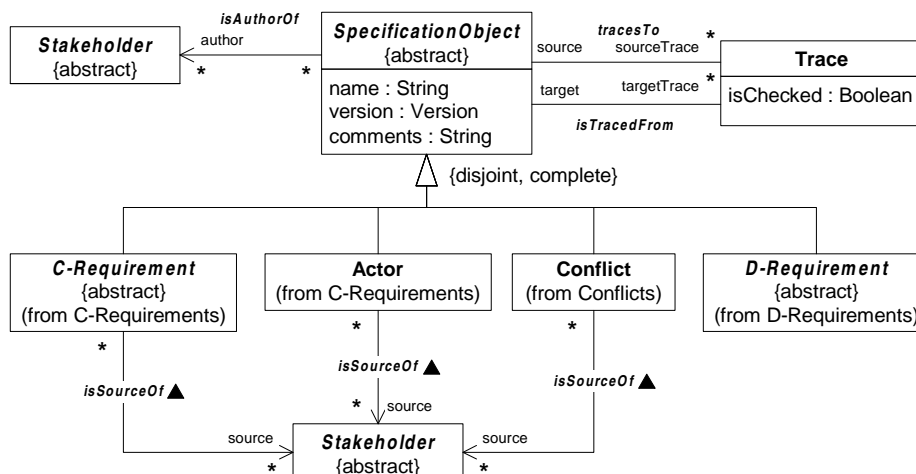


Figura E.3: Jerarquía de objetos de especificación

## E.2.2 Modelo de los requisitos–C

Una especificación de requisitos–C puede considerarse que está compuesta por un conjunto de requisitos–C, dentro del cual se incluyen los objetivos, los requisitos de almacenamiento de información, los requisitos funcionales o casos de uso y los requisitos no funcionales, tal como puede verse en la figura E.4.

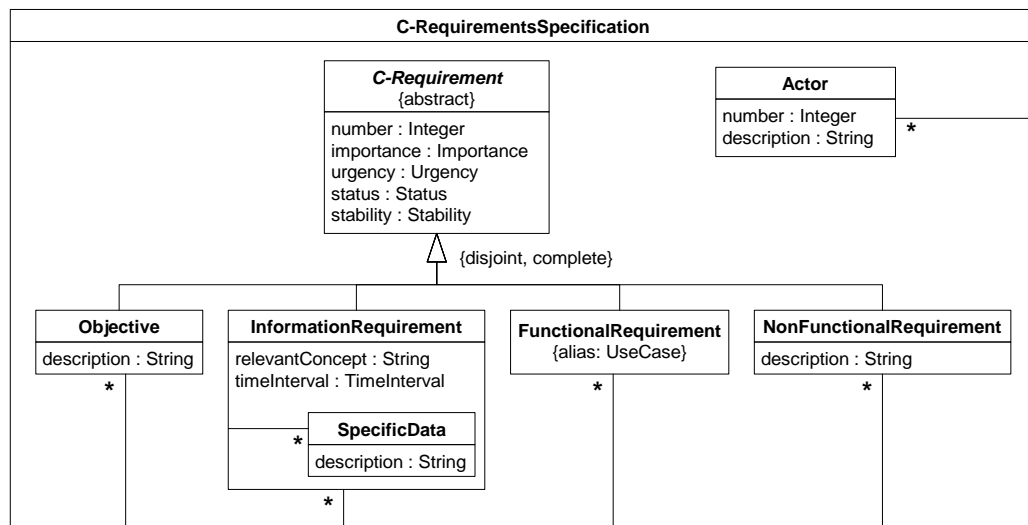


Figura E.4: Estructura las especificaciones de requisitos–C

Además de estos elementos ya mencionados, es necesario incluir a los actores que participarán en los casos de uso. Los actores no son estrictamente requisitos, de ahí que no se hayan considerado como un subtipo de C–Requirement.

En la figura E.4 puede también observarse la composición de cada tipo de requisito–C a excepción de los requisitos funcionales, cuya estructura puede verse en la figura E.5. Mientras que los actores, objetivos y requisitos no funcionales están compuestos básicamente por su descripción textual, los requisitos de almacenamiento de información, representados por el tipo InformationRequirement, están compuestos por los datos específicos sobre el concepto relevante del que el sistema debe almacenar información.

En el caso de los requisitos funcionales, la estructura es más compleja. Aparte de los atributos para expresar información como si es o no un caso de uso abstracto, el evento de activación, la precondition, etc., un caso de uso se considera que está formado por una secuencia de pasos.

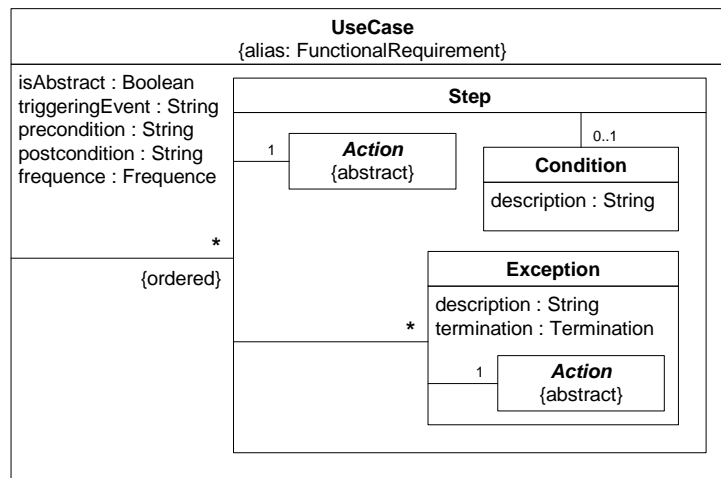


Figura E.5: Estructura de los casos de uso

Un paso puede o no tener una condición, debe tener siempre una acción y puede tener varias excepciones asociadas, cada una de ellas compuesta a su vez por la acción a realizar en el caso de que se produzca alguna situación excepcional.

Las acciones pueden clasificarse tal como puede verse en la figura E.6. Una acción puede consistir en realizar otro caso de uso, es decir una *inclusión* o una *extensión* en función de que la acción venga determinada o no por una condición [Booch *et al.* 1999]. Puede ocurrir que la acción consista en que el sistema realice algún tipo de operación, a la que puede asignársele un tiempo máximo de realización mediante el atributo *performance*, o que la acción se realice por parte de algún actor.

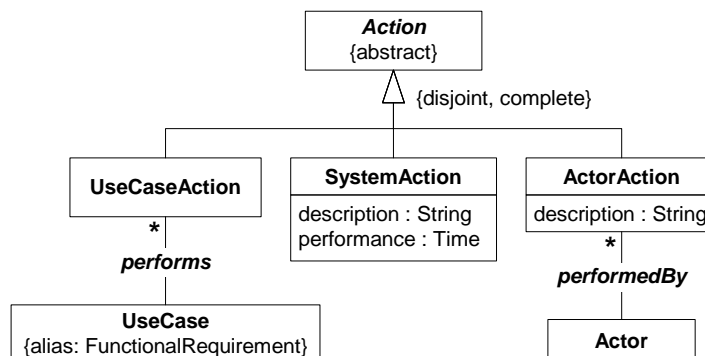


Figura E.6: Jerarquía de acciones



Dentro de los requisitos-D, se han considerado las asociaciones como un subtipo de ObjectType debido a la gran cantidad de propiedades que comparten y porque, a nuestro entender, es la forma más sencilla de modelar asociaciones como tipos.

En el caso de una asociación normal, siempre puede generalizarse al caso de una asociación como tipo aplicando la transformación que puede verse en la figura E.8.

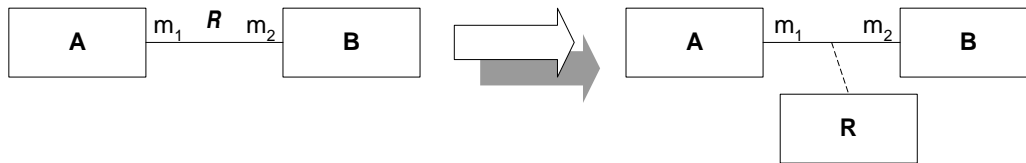


Figura E.8: Transformación de asociaciones en asociaciones como tipo

Las propiedades abstractas de un tipo de objetos pueden clasificarse por un lado en atributos y enlaces y por otro como constantes, variables y derivados.

Por motivos de simplicidad conceptual, en esta clasificación se ha optado por expresar el hecho de que una propiedad es constante, variable o derivada mediante atributos booleanos en lugar de usar una clasificación ortogonal a la que especializa las propiedades como atributos o enlaces. De esta forma la clasificación es más sencilla aunque es necesario una invariante más compleja, tal como puede verse en la figura E.9.

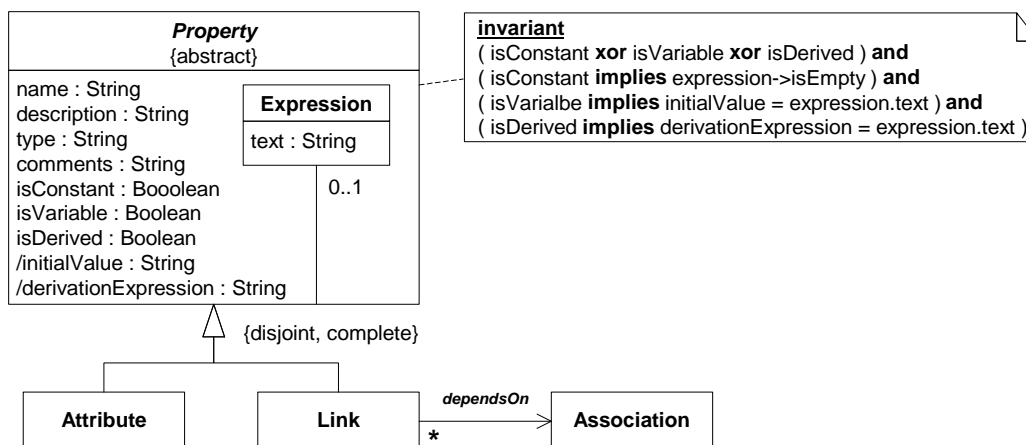


Figura E.9: Clasificación de propiedades de tipos de objetos

## E.2.4 Modelo de los conflictos

Una especificación de conflictos, representando o no a un documento físico, está compuesta, obviamente, por conflictos. Cada conflicto, tal como puede verse en la figura E.10, afecta a varios requisitos-C y está compuesto por varias alternativas y una solución.

A su vez, cada alternativa tiene uno o más participantes como sus autores.

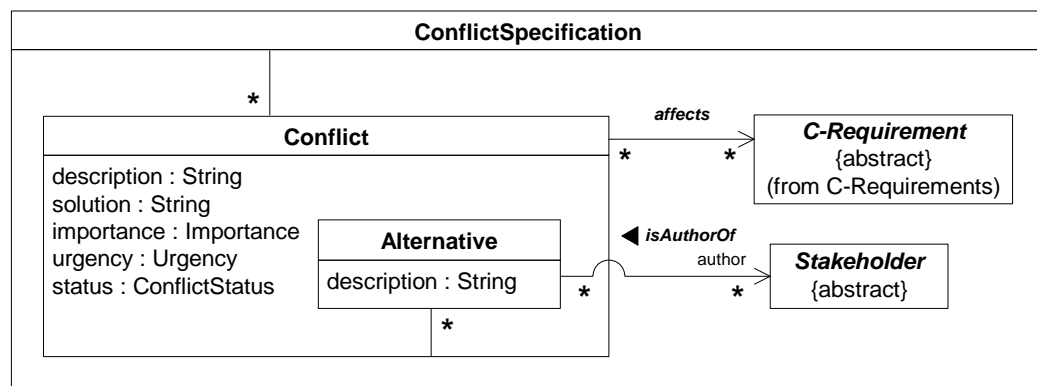


Figura E.10: Estructura de los conflictos

## E.3 Arquitectura del prototipo

La arquitectura externa de la herramienta CASE, a la que se le ha denominado tentativamente como *REM* (*Requirements Engineering Methodology*), puede verse en la figura E.11.

En la versión actual, los proyectos de ingeniería de requisitos se almacenan en bases de datos relacionales, en concreto MS–Access en su versión 97, cuya capacidad para tener vistas actualizables facilita enormemente la aplicación de las heurísticas, descritas entre otros en [Durán *et al.* 1997], para dar persistencia a sistemas de objetos en entornos relacionales.

La herramienta permite la manipulación de los contenidos de los proyectos de requisitos y genera con dicha información ficheros XML (*eXtensible Markup Language*) [Simpson 1999] que, al combinarse con hojas de estilo XSL (*eXtended Style Language*), generan páginas HTML (*HyperText Markup Language*) que pueden considerarse como la documentación final siguiendo una filosofía WYSIWYG (*What You See Is What You Get*) y que pueden

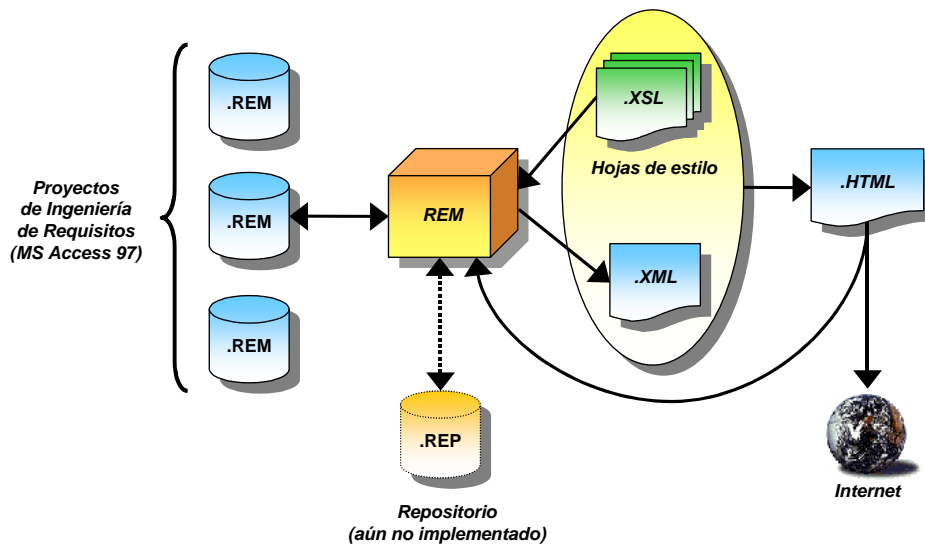


Figura E.11: Arquitectura de la herramienta CASE

distribuirse a través de Internet, facilitando de esta forma el trabajo de grupos aunque sus miembros se encuentren distantes geográficamente.

La separación de la información de su representación mediante el uso combinado de XML y XSL permite, de forma externa a la herramienta, representar la misma información de formas diferentes, permitiendo la adaptación de los formatos de los documentos a las necesidades de los distintos usuarios.

Aunque aún no se ha implementado, está prevista la incorporación de un repositorio donde puedan almacenarse los patrones-R descritos en los capítulos 3 y 4. Para la implementación de este repositorio se están teniendo en cuenta las propuestas descritas en [García 2000].

En próximas versiones puede que se sustituya la persistencia en bases de datos relacionales por la persistencia directamente en XML, de forma que se elimine el paso intermedio de generar XML a partir de la información almacenada en la base de datos.

Internamente, la arquitectura de la herramienta sigue la arquitectura *documento-vista* impuesta por el framework para aplicaciones basadas en documento que genera automáticamente el generador de aplicaciones de MS Visual C++ en su versión 6.0, que ha sido el entorno utilizado para su implementación.

## E.4 Interfaz de usuario del prototipo

Para la interfaz de usuario de la herramienta se ha intentado seguir lo más fielmente la filosofía WYSIWYG recomendada en las técnicas de elicitación descritas en el capítulo 3.

La ventana principal de la aplicación, que puede verse en las figuras E.12 a E.15, está dividida en dos zonas principales. La parte izquierda muestra cuatro vistas solapadas (proyecto, requisitos-C, requisitos-D y conflictos) mientras que la parte derecha muestra en HTML la información correspondiente al objeto seleccionado de la vista activa de la parte izquierda.

A continuación se describe cada una de las cuatro vistas solapadas.

### E.4.1 Vista de proyecto

En la vista de proyecto (figura E.12), el usuario puede editar los datos del proyecto así como añadir, eliminar o editar la información sobre las organizaciones involucradas y sobre los participantes en el proyecto, que pueden ser clientes, usuarios o desarrolladores, indicado por las iniciales entre corchetes que aparecen a la derecha de los nombres.

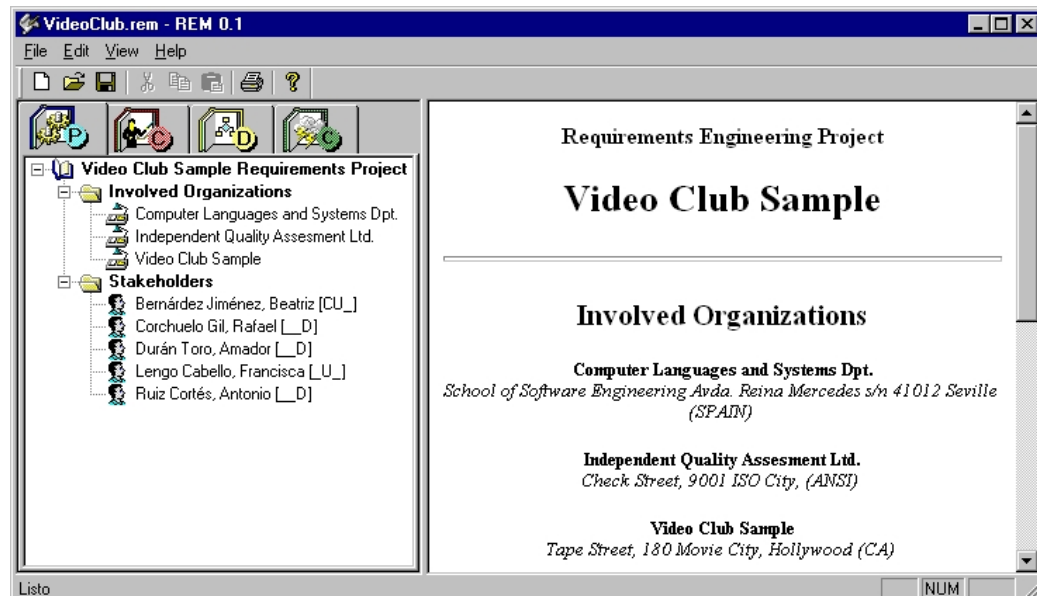


Figura E.12: Vista de proyecto de la herramienta CASE



## E.4.2 Vista de requisitos–C

En la vista de requisitos–C (figura E.13), el usuario puede editar los datos de la especificación de requisitos–C (nombre, versión, etc.) así como añadir, eliminar o editar objetivos, requisitos de almacenamiento de información, requisitos funcionales, requisitos no funcionales y actores.

Tal como puede verse en la figura E.13, la estructura de los distintos objetos de especificación compuestos es visible para el usuario en la estructura jerárquica que forma la vista.

Así, los requisitos de almacenamiento de información tienen dependiendo de ellos a los datos específicos sobre el concepto relevante y los requisitos funcionales, al expandirse la rama correspondiente, muestran los distintos pasos que los componen y éstos a su vez las posibles excepciones que tengan asociadas.

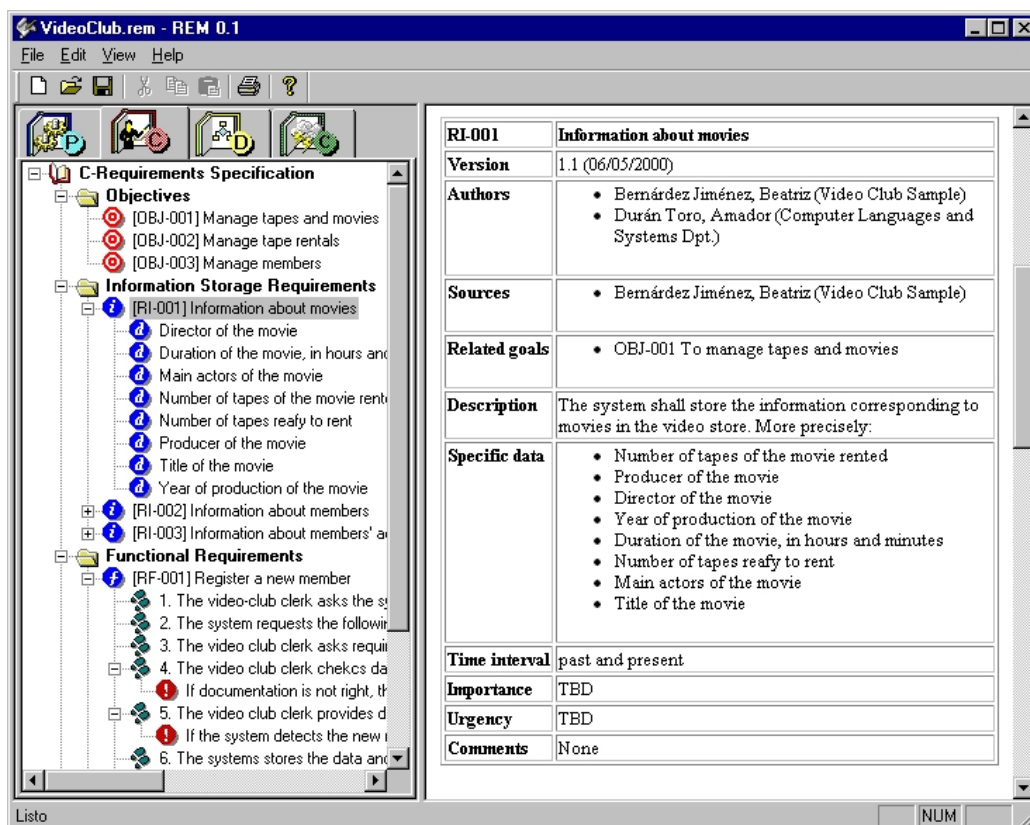


Figura E.13: Vista de requisitos–C de la herramienta CASE

### E.4.3 Vista de requisitos-D

En la vista de requisitos-D (figura E.14), el usuario, además de editar los datos sobre la especificación de requisitos-D, puede añadir, eliminar o editar los distintos elementos que componen el modelo del sistema, es decir, los requisitos-D.

La estructura jerárquica de la especificación de requisitos-D que puede manipular el usuario está dividida en tres secciones principales, correspondientes a los tipos de objetos, las asociaciones entre los tipos de objetos y las operaciones del sistema.

En el caso de los tipos de objetos, al expandirse muestran sus enlaces y atributos. Las asociaciones muestran los roles que las componen así como los posibles atributos y enlaces en el caso de que los tengan y las operaciones del sistema muestran sus parámetros, precondiciones, post-condiciones y excepciones.

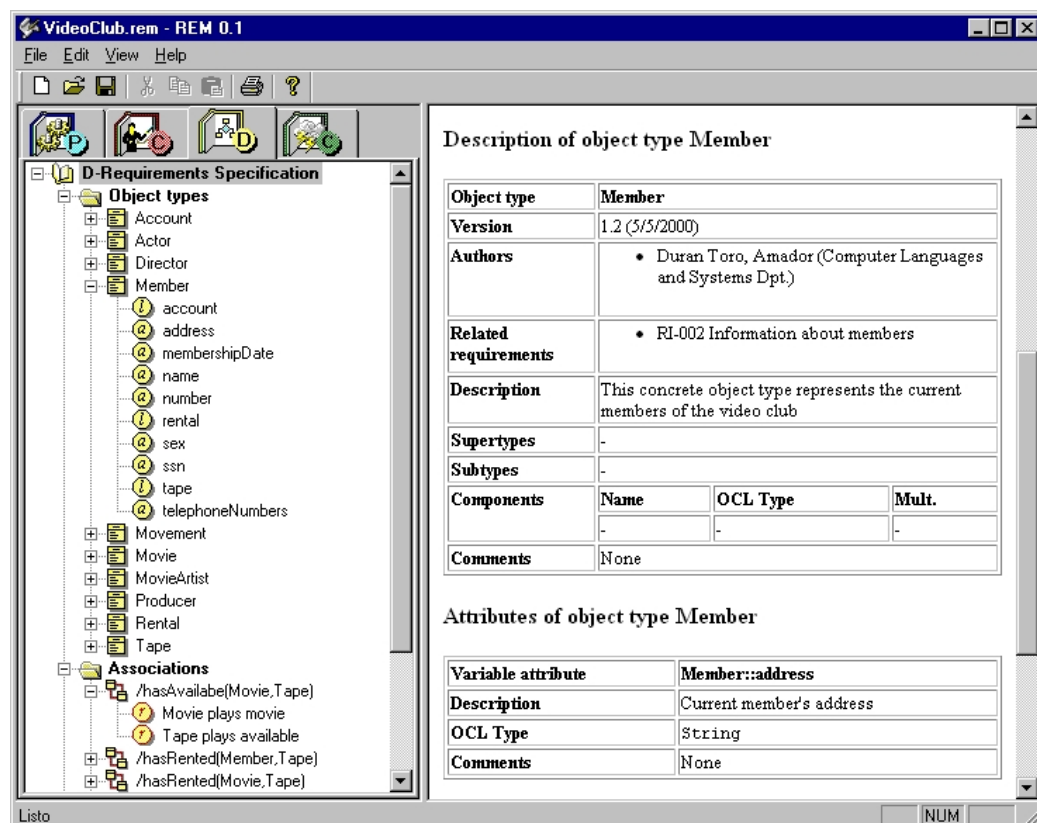


Figura E.14: Vista de requisitos-D de la herramienta CASE

### E.4.4 Vista de conflictos

En la vista de conflictos (figura E.15), el usuario puede añadir, eliminar y editar la información correspondiente a los conflictos descubiertos durante el proceso.

Para representar a los conflictos se utiliza un icono que representa una balanza, reforzando la idea de que para la resolución de los conflictos es necesario la mayor parte de la veces negociar una solución que resulte justa a todos los involucrados [Boehm *et al.* 1994].

Cada conflicto, al igual que ocurre con los demás objetos compuestos al expandirse, muestra las distintas alternativas, representadas mediante un icono de una puerta que simboliza una posible salida al conflicto.

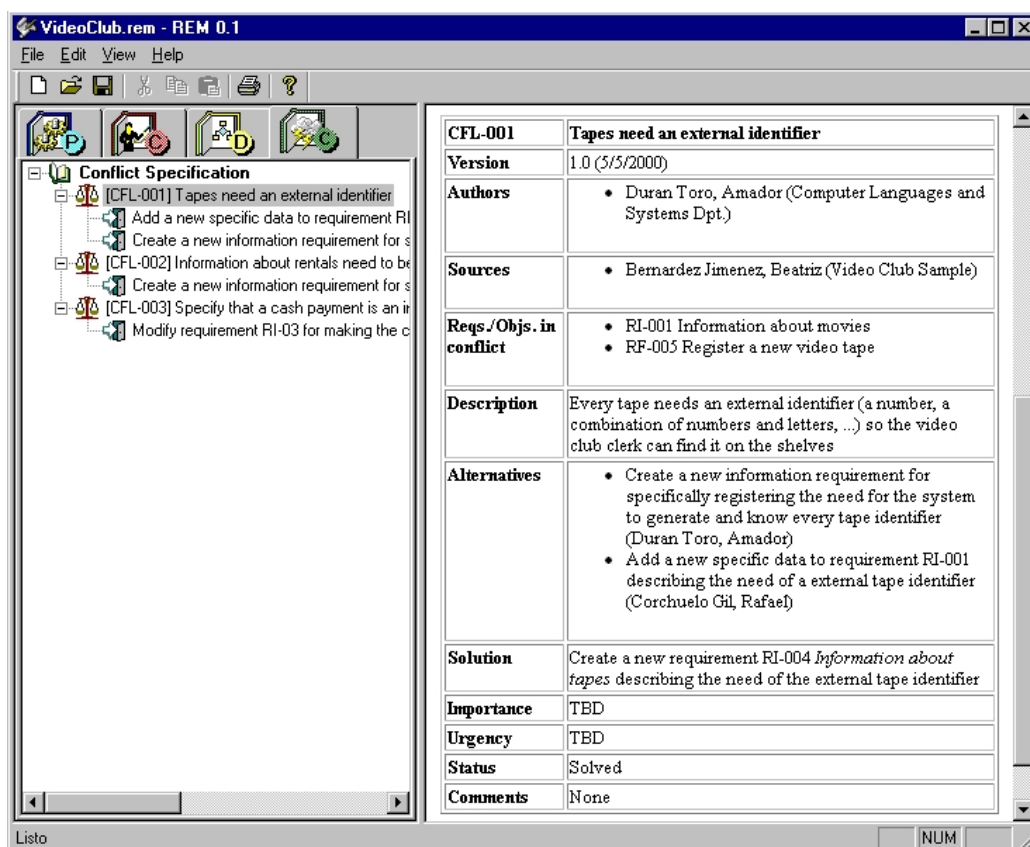


Figura E.15: Vista de conflictos de la herramienta CASE

### E.4.5 Otros elementos de la interfaz de usuario

En general, la filosofía de la interfaz de usuario de la herramienta desarrollada, además de seguir la filosofía WYSIWYG mostrando en todo momento al usuario el aspecto de la documentación final en la ventana de la parte derecha, es mostrar de la forma más visual posible la estructura de los objetos que puede manipular el usuario.

Cada objeto de los que aparecen en cualquiera de las cuatro vistas solapadas tiene asociado un menú contextual que permite realizar una serie de operaciones sobre él.

Al seleccionar un objeto, la información de la ventana de la parte derecha desplaza su contenido para mostrar la información correspondiente al objeto, manteniendo así sincronizadas ambas vistas.

A la hora de implementar los distintos diálogos de edición de los objetos de especificación se ha conseguido un alto grado de reutilización al utilizar diálogos solapados (ver figura E.16).

De esta forma, para editar un objeto de una clase que tiene por encima uno o más niveles de herencia se reutilizan los diálogos solapados correspondientes a sus superclases, por lo que sólo es necesario implementar el diálogo solapado correspondiente a los atributos o enlaces específicos de la nueva clase y combinarlos después en un único diálogo.

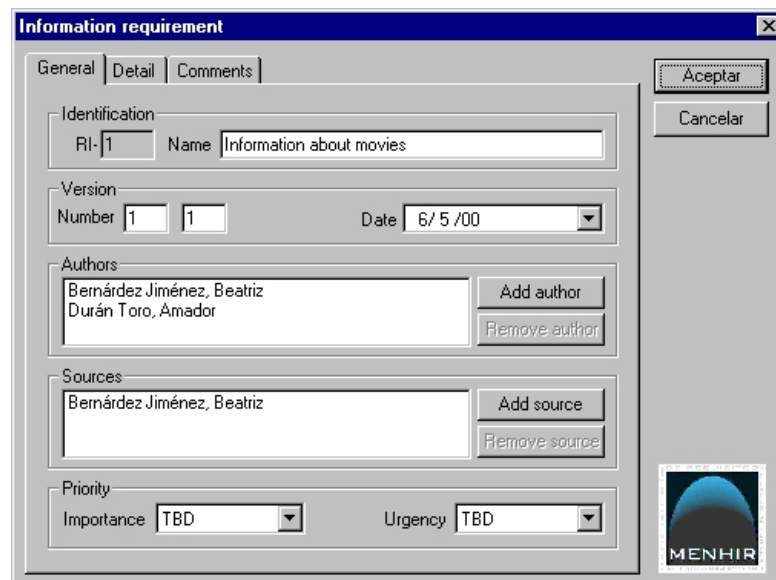


Figura E.16: Ejemplo de diálogo compuesto

Por ejemplo, en la figura E.16 puede verse el diálogo correspondiente a un requisito de almacenamiento de información. Los diálogos solapados de nombre *Comments* y *Traceability* son *heredados* de la clase *CSpecificationObject*<sup>1</sup>, el diálogo de nombre *General* es heredado de la clase *CC\_Requirement* y el de nombre *Detail* es propio de la clase que representa a los requisitos de almacenamiento de información.

## E.5 Otras herramientas CASE de requisitos

El INCOSE (*International Council on Systems Engineerings*) ha realizado una encuesta comparativa sobre herramientas comerciales de gestión de requisitos [INCOSE 2000]. En dicha encuesta se comparan 15 herramientas comerciales en función de si soportan o no 14 funcionalidades generales divididas a su vez en funcionalidades más específicas.

La mayor parte de estas herramientas comerciales son en realidad gestores de documentación, que permiten gestionar documentos en formato electrónico, normalmente Word o un formato propietario, y establecer relaciones entre los distintos párrafos o elementos de otro tipo que componen dichos documentos.

En general, todas se basan en sistemas centralizados de gestión de bases de datos para almacenar la información correspondiente a los requisitos, que suelen consistir en párrafos de texto libre con una serie de atributos predefinidos y a los que la mayoría de herramientas permiten asociar nuevos tipos de atributos por parte del usuario.

Todas las herramientas asumen que la estructura de los requisitos es jerárquica, de forma que un requisito puede estar formado o tener asociados otros requisitos de nivel inferior, y la mayoría permite extraer párrafos de ficheros generados por procesadores de texto comerciales y convertirlos en requisitos.

Otras de las características comunes a la mayor parte de las herramientas es la posibilidad de realizar consultas sobre los requisitos en función de determinados valores de sus atributos.

La única herramienta a la que se ha tenido acceso durante el desarrollo de este trabajo ha sido *Requisite Pro* en sus versiones 3.1 y 4.5 [Rat 2000]. Esta herramienta almacena los requisitos en bases de datos relacionales,

---

<sup>1</sup>Esta clase es la implementación del tipo *SpecificationObject* que puede verse en la figura E.3.

Access, Oracle o SQL Server, enlazándolos con documentos Word. También presenta la capacidad, aunque bastante limitada, de asociar requisitos-C con elementos de modelado, requisitos-D, generados por la herramienta Rational Rose.

Los requisitos, que son básicamente párrafos de un documento Word, tienen una serie de atributos predefinidos y se permite al usuario añadirles nuevos tipos de atributos. También se permite al usuario crear tipos de requisitos, de forma que sus identificadores tengan un prefijo determinado y se presenten con determinado tipo de letra y en determinados colores.

Desde nuestro punto de vista, la principal característica de esta herramienta es el tratamiento de la rastreabilidad, especialmente mediante matrices de rastreabilidad como la que puede verse en la figura E.17.

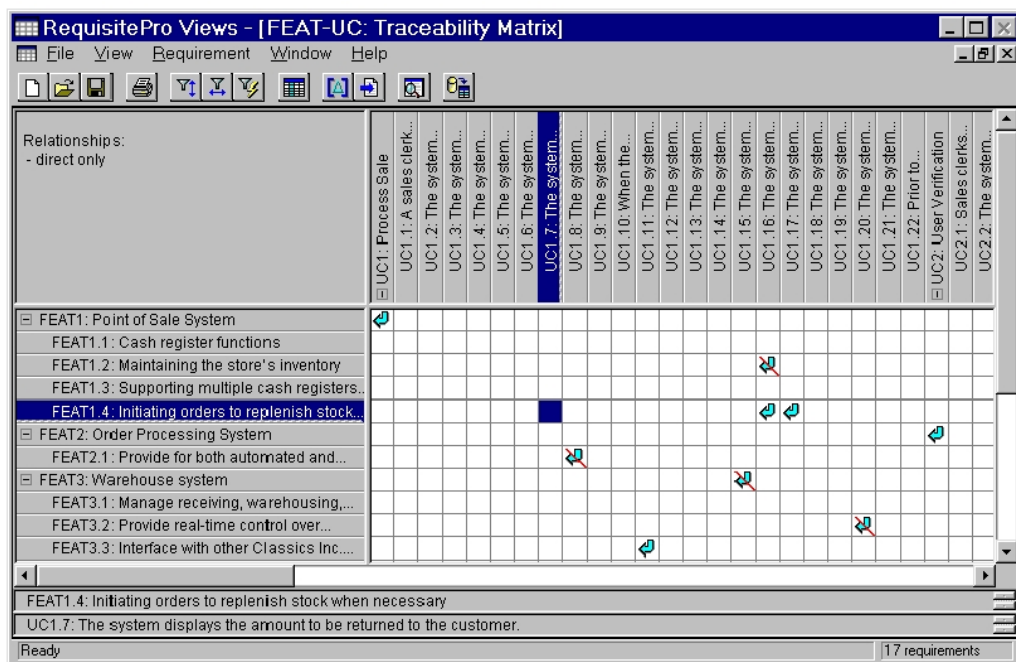


Figura E.17: Matriz de rastreabilidad de Requisite Pro

Comparada con el prototipo de herramienta CASE desarrollado como complemento a esta tesis, Requisite Pro ofrece ventajas como el acceso multiusuario a un proyecto de requisitos, una mayor facilidad para establecer y visualizar las relaciones de rastreabilidad, la posibilidad de ampliar el conjunto de atributos de los requisitos de forma dinámica y la capacidad para establecer filtros o realizar consultas sobre los requisitos.

En la versión 4.5 existe incluso la posibilidad de acceder a través de Internet mediante un navegador para visualizar requisitos y editarlos con ciertas restricciones (ver figura E.18), y la interesante capacidad de organizar discusiones electrónicas entre los participantes en el proyecto de forma similar a las que se mantienen en las *news* de Internet (ver figura E.19).

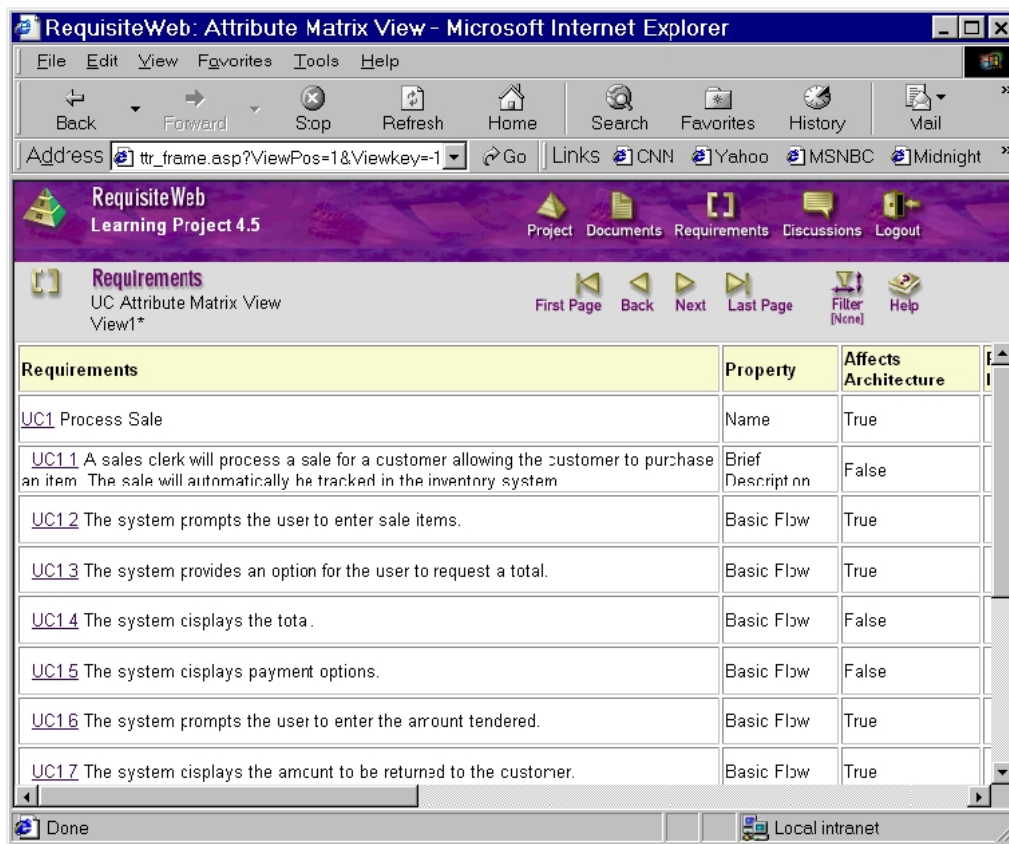


Figura E.18: Acceso a través de Internet de Requisite Pro

Sin embargo, nos parece que aún permanece la idea de que un requisito es básicamente un párrafo de un documento, sobre todo por la gran dependencia que presenta la herramienta con respecto a Word, y su escasa integración con herramientas de modelado hace que su integración con el resto del desarrollo sea bastante limitado.

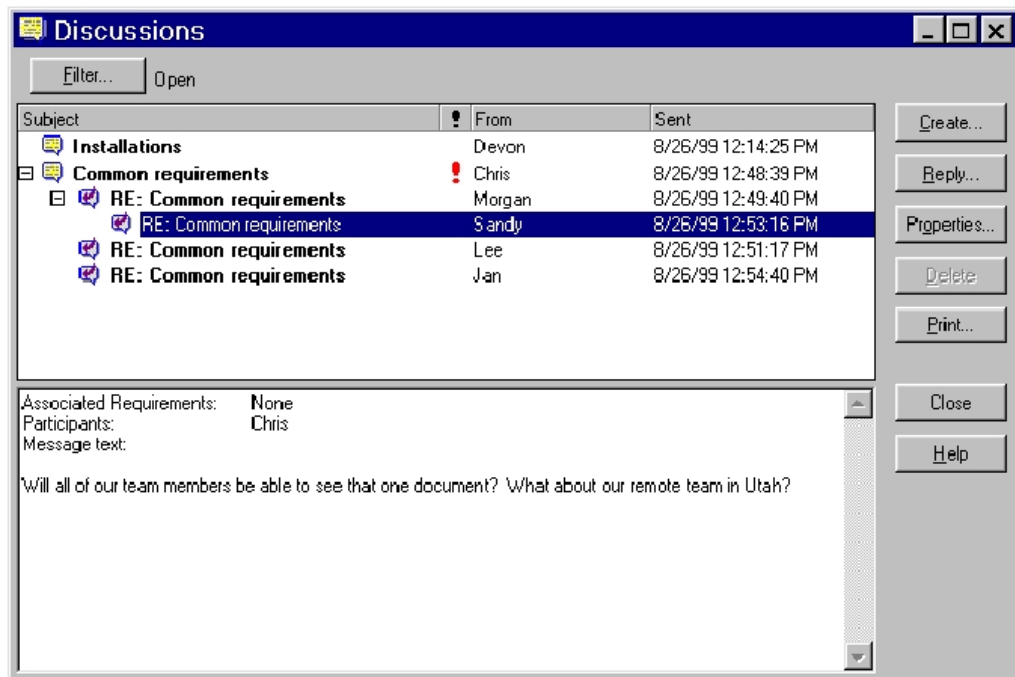


Figura E.19: Foros de discusión de Requisite Pro

## E.6 Bibliografía

- [Boehm *et al.* 1994] B. W. Boehm, P. Bose, E. Horowitz, y M.-J. Lee. Software Requirements as Negotiated Win Conditions. En *Proceedings of the First International Conference on Requirements Engineering*, 1994. Disponible en <http://sunset.usc.edu/TechRpts/Papers/NGPM-Requirements93.ps>.
- [Booch *et al.* 1999] G. Booch, J. Rumbaugh, y I. Jacobson. *The Unified Modeling Language User Guide*. Addison-Wesley, 1999.
- [Brooks 1995] F. P. Brooks, Jr. *The Mythical Man-Month: Essays on Software Engineering Anniversary Edition*. Addison-Wesley, 1995.
- [Davis 1995] A. M. Davis. Tracing: A Simple Necessity Neglected. *IEEE Software*, 12(5), Septiembre 1995.
- [Durán *et al.* 1997] A. Durán, A. Amaya, y M. Toro. Design of an Automatic Generator of Object-Relational Persistency Mechanisms. En *Actas de las II Jornadas de Ingeniería del Software*, San Sebastián, 1997.



- [García 2000] F. J. García. *Modelo de Reutilización Soportado por Estructuras Complejas de Reutilización Denominadas Mecanos*. Tesis doctoral, Universidad de Salamanca, 2000.
- [INCOSE 2000] INCOSE. Tools Survey: Requirements Management (RM) Tools. Informe técnico, International Council on Systems Engineering, 2000. Disponible en <http://www.incose.org/tools/tooltax.html>.
- [Rat 2000] Rational Software Corporation. *Rational Requisite Pro 4.5 Evaluators Guide*, 2000. Disponible en <http://www.rational.com>.
- [Simpson 1999] J. E. Simpson. *Just XML*. Prentice-Hall, 1999.

