



PROGRAMA DE LA ASIGNATURA "Estructuras de Datos y Algoritmos"

INGENIERO EN INFORMÁTICA (Plan 97)

Departamento de Lenguajes y Sistemas Informáticos

Escuela Técnica Superior de Ingeniería Informática

DATOS BÁSICOS DE LA ASIGNATURA

Titulación:	INGENIERO EN INFORMÁTICA (Plan 97)
Año del plan de estudio:	1997
Centro:	Escuela Técnica Superior de Ingeniería Informática
Asignatura:	Estructuras de Datos y Algoritmos
Código:	260014
Tipo:	Troncal/Formación básica
Curso:	2
Período de impartición:	Segundo Cuatrimestre
Ciclo:	1
Área:	LENGUAJES Y SISTEMAS INFORMATICOS
Departamento:	Lenguajes y Sistemas Informáticos
Dirección postal:	ETS Ingeniería Informática. Av. Reina Mercedes S/N
Dirección electrónica:	http://www.lsi.us.es

OBJETIVOS Y COMPETENCIAS

Objetivos docentes específicos

- * Proveer al alumno de las técnicas algorítmicas básicas que le permitirán abordar el desarrollo de programas correctos y eficientes para resolver problemas no triviales. Las técnicas básicas mencionadas incluyen conocimientos teóricos y prácticos, habilidades, experiencias y sentido crítico, todas ellas fundamentadas en teorías y técnicas sólidas, comprobadas y bien establecidas.
- * Conocer mejor cómo es un lenguaje de programación, en particular un lenguaje orientado a objetos. Con este objetivo se estudiarán aspectos como son la estructura de control, el tipo de datos, la gestión de memoria y los mecanismos de abstracción de un lenguaje de estas características.
- * Conocer nuevas técnicas de programación. En particular, el uso de la memoria dinámica y las estructuras de datos enlazadas, que están en la base de muchas aplicaciones.
- * Introducir herramientas de diseño de algoritmos y la ingeniería algorítmica como selección de las estructuras de datos y de las técnicas algorítmicas más adecuadas para la resolución de un problema concreto.
- * Profundizar en el aprendizaje de la programación estructurada. Introducir técnicas para diseñar programas de tamaño mediano. Proporcionar al alumno más experiencia en el campo de la programación mediante la realización de prácticas.
- * Ampliar el dominio de la recursividad como herramienta de construcción de programas.

Competencias:

Competencias transversales/genéricas

- Capacidad de análisis y síntesis (Se entrena de forma intensa)
- Capacidad de organizar y planificar (Se entrena de forma intensa)
- Capacidad para aplicar la teoría a la práctica (Se entrena de forma intensa)
- Habilidades para trabajar en grupo (Se entrena de forma intensa)

Competencias específicas

Cognitivas(Saber)

- * Análisis de algoritmos: Eficiencia, costes, casos, notación asintótica, resolución de recurrencias simples.
- * Análisis en caso mediano.
- * Estructuras de datos: listas, pilas, colas, colas de prioridad, conjuntos, árboles, grafos.
- * Algoritmos fundamentales: caminos mínimos en grafos, árboles de expansión, dispersión, etc.
- * Tipos de datos en lenguajes de programación.
- * Sentencias de control en lenguajes de programación.
- * Abstracción de datos. Especificación e implementación.
- * Implementaciones de datos con nodos encadenados y memoria dinámica.
- * Gestión de memoria dinámica. Recolección de basura.
- * Implementaciones de TADs fundamentales: listas, colas, pilas, árboles, tablas de dispersión y grafos.
- * Herencia.
- * Comprender los conceptos de módulo y de abstracción de datos.
- * Aprender técnicas de descomposición modular de programas.
- * Acostumbrarse a analizar la eficiencia de los programas como un criterio de calidad.
- * Adquirir conceptos básicos de programación orientada a objetos.
- * Conocer algunos algoritmos clásicos para problemas fundamentales.

Procedimentales/Instrumentales(Saber hacer)

- * Avanzar en técnicas fundamentales de diseño y análisis de algoritmos y de estructuras de datos.
- * Adquirir una metodología para la especificación, el uso, el diseño y la implementación de TAD.
- * Conocer y saber utilizar una variedad de técnicas de implementación eficientes de estructuras de datos, incluyendo el uso de memoria dinámica.
- * Conocer TAD genéricos, saber utilizarlos y saber adaptarlos por dar soluciones a necesidades específicas.
- * Saber razonar sobre la corrección y la eficiencia tanto de las implementaciones de TAD como de los algoritmos que los utilizan.
- * Disponer de criterios que permitan, durante las etapas de especificación, diseño e implementación escoger la alternativa más adecuada, y disponer de elementos por argumentar de forma razonada las elecciones realizadas.
- * Saber particularizar esquemas algorítmicos generales para resolver problemas.
- * Conocer mejor las prestaciones de un lenguaje de programación.
- * Tomar contacto con algunas técnicas avanzadas de programación.
- * Saber identificar las componentes más relevantes de un problema y seleccionar la técnica algorítmica más adecuada.
- * Ser capaz de seleccionar los tipos de datos más adecuados para mejorar la eficiencia de una solución algorítmica.
- * Crear nuevos tipos de datos, separando las fases de especificación e implementación.
- * Descomponer un problema de cierto tamaño en módulos independientes.
- * Producir programas de tamaño medio fiables y fáciles de entender, modificar, mantener y reutilizar.
- * Usar módulos predefinidos.
- * Escribir programas recursivos sobre tipos no necesariamente básicos y poder razonar sobre su corrección y eficiencia.

Actitudinales(Ser)

- * Capacidad para el razonamiento crítico y lógico-matemático.
- * Capacidad de resolver problemas aplicando los métodos de la ciencia y la ingeniería.
- * Capacidad para diseñar sistemas, componentes o procesos que se ajusten a unas necesidades, utilizando los métodos, técnicas y herramientas más adecuadas en cada caso.
- * Capacidad de abstracción.
- * Capacidad para enfrentarse a problemas nuevos recurriendo conscientemente a estrategias que han sido útiles en problemas resueltos anteriormente.
- * Capacidad para aplicar el ciclo de resolución de problemas típico de la ciencia y la ingeniería: especificación, generación de ideas y alternativas, diseño de una estrategia de solución, ejecución de la estrategia, validación, interpretación y evaluación de los resultados.
- Capacidad de analizar el proceso una vez acabado.
- * Capacidad para crear y utilizar modelos de la realidad.
- * Capacidad de análisis y de síntesis.

CONTENIDOS DE LA ASIGNATURA

- BLOQUE I. Introducción a las Estructuras de Datos
- BLOQUE II. Conjuntos y Funciones
- BLOQUE III. Grafos

ACTIVIDADES FORMATIVAS

Relación de actividades de segundo cuatrimestre

Clases teóricas

Horas presenciales: 50.0

Horas no presenciales: 75.0

Metodología de enseñanza-aprendizaje:

Durante las semanas que dura el curso, organizadas según se adjunta en la temporización previa, se impartirá el contenido teórico de la asignatura mediante el uso de presentaciones en ordenador y en formato tradicional, ilustrando con especial atención los problemas de relevancia con ejemplos clarificadores.

El material proporcionado a los alumnos, en forma de presentaciones elaboradas por los profesores de la asignatura, es un material de apoyo de gran valor para el seguimiento de la asignatura. Dichas presentaciones estarán accesibles en Internet en la página web de la asignatura.

El profesor utilizará fundamentalmente la tradicional pizarra para exponer sus explicaciones y razonamientos.

También se utilizará en clase el retroproyector u ordenador conectado al cañón de proyección, que resulta muy útil para la introducción de conceptos o la explicación de algoritmos que, por su tamaño o nivel de complejidad resultarían arduos de explicar en el encerado. En este caso, el material utilizado se pondrá a disposición de los alumnos con suficiente antelación para su correcto seguimiento de las clases.

Competencias que desarrolla:

Todas las específicas y las siguientes genéricas:

Capacidad de análisis y síntesis
Capacidad de organizar y planificar
Habilidades para trabajar en grupo
Capacidad para aplicar la teoría a la práctica

Prácticas informáticas

Horas presenciales: 25.0

Horas no presenciales: 37.5

Metodología de enseñanza-aprendizaje:

Cada grupo de teoría se subdividirá en tres grupos de laboratorios. Cada semana uno de los grupos recibirá clases prácticas en los laboratorios. Estos laboratorios están dotados con ordenadores y el software adecuado para la puesta en práctica de los conceptos introducidos en las clases teóricas. Los profesores impartirán las clases fundamentalmente con diapositivas que se visualizarán con el ordenador dispuesto en las aulas para tal efecto; dichas enseñanzas estarán apoyadas por el uso de la pizarra en la que se detallarán los aspectos que se consideren necesarios para el correcto aprendizaje del alumno.

Competencias que desarrolla:

Todas las específicas y las siguientes genéricas:

Capacidad de análisis y síntesis
Capacidad de organizar y planificar
Habilidades para trabajar en grupo
Capacidad para aplicar la teoría a la práctica

SISTEMAS Y CRITERIOS DE EVALUACIÓN Y CALIFICACIÓN

EVALUACIÓN ORDINARIA

Tanto la evaluación de la parte teórica como práctica se hará mediante un examen escrito que se realizará en las fechas que indique el Centro para cada una de las convocatorias oficiales.

La nota final del alumno se calculará ponderando las partes prácticas y teóricas según su dedicación. Se considerará que un alumno ha superado la asignatura cuando la calificación que obtiene en el sistema de evaluación elegido sea igual o superior a 5.

EVALUACIÓN ALTERNATIVA

El alumno también tendrá la posibilidad de aprobar la asignatura sin necesidad de presentarse al examen final mediante una evaluación alternativa. Esta evaluación alternativa consistirá en la superación de una serie de ejercicios escritos tanto teóricos como de programación, y la realización de un conjunto de problemas en clases de teoría y de prácticas de laboratorio. Para los ejercicios escritos se realizarán dos pruebas, que contendrán tanto aspectos teóricos como prácticos de la asignatura. En las sesiones de laboratorio se desarrollarán las diferentes prácticas y el profesor evaluará el trabajo realizado por el alumno. Se podrá obtener puntos con la realización de las prácticas en el laboratorio, que serán sumados a la nota final de prácticas siempre que se haya asistido al menos a un 60% de las prácticas de aula. De igual forma, a la nota de teoría se le sumará la puntuación correspondiente a los trabajos realizados en clase, pudiéndose exigir un mínimo de asistencia.

Se considerará que un alumno ha superado la asignatura cuando la calificación que obtiene en el sistema de evaluación elegido sea igual o superior a 5