DESCRIPCIÓN DE LA ASIGNATURA

Grado/Máster en:	Graduado/a en Ingeniería del Software por la Universidad de Málaga
Centro:	Escuela Técnica Superior de Ingeniería Informática
Asignatura:	Estructura de Datos
Código:	204
Tipo:	Obligatoria
Materia:	Programación de Computadores
Módulo:	Formación común
Experimentalidad:	69 % teórica y 31 % práctica
Idioma en el que se imparte:	Inglés, Español
Curso:	2
Semestre:	1
Nº Créditos	6
Nº Horas de dedicación del estudiante:	150
Nº Horas presenciales:	60
Tamaño del Grupo Grande:	72
Tamaño del Grupo Reducido:	30
Página web de la asignatura:	

EQUIPO DOCENTE

Departamento: LENGUAJES Y CIENCIAS DE LA COMPUTACIÓN

Área: LENGUAJES Y SISTEMAS INFORMÁTICOS

Nombre y Apellidos	Mail	Teléfono Laboral	Despacho	Horario Tutorías
Coordinador/a: PABLO LOPEZ OLIVAS	plopez@uma.es	952133305	3.2.50 - E.T.S.I. INFORMÁTICA	Primer cuatrimestre: Lunes 12:30 - 14:00, Martes 12:00 - 14:00, Jueves 10:00 - 12:30
JOSE ENRIQUE GALLARDO RUIZ	jegallardo@uma.es	952133305	3.2.2 - E.T.S.I. INFORMÁTICA	Primer cuatrimestre: Martes 10:00 - 12:00, Miércoles 09:30 - 12:40, Lunes 12:45 - 13:35
LAURA PANIZO JAIME	laurapanizo@uma.es	951952955	3.3.12 - E.T.S.I. INFORMÁTICA	Primer cuatrimestre: Miércoles 11:00 - 14:00, Viernes 10:45 - 13:45

RECOMENDACIONES Y ORIENTACIONES

Esta asignatura requiere conocimientos básicos de programación y matemáticas. Para su comprensión será necesario haber asimilado los contenidos fundamentales de las siguientes asignaturas de formación básica del primer curso: Fundamentos de la Programación, Programación Orientada a Objetos, Estructuras Algebraicas de la Computación y Matemática Discreta.

CONTEXTO

Esta asignatura se imparte en el primer cuatrimestre del segundo curso de los tres grados de Informática: Ingeniería Informática, Ingeniería del Software e Ingeniería de Computadores. En este mismo cuatrimestre se imparte también la asignatura Análisis y Diseño de Algoritmos. Ambas asignaturas completan el estudio de las técnicas de programación impartidas en las asignaturas de programación de primer curso.

Las estructuras de datos son imprescindibles para que los algoritmos puedan expresarse de forma clara y concisa, así como para que puedan ejecutarse de manera eficiente; por ello existe gran coordinación con la asignatura Análisis y Diseño de Algoritmos. Se describirán las estructuras de datos más importantes de la programación, así como las implementaciones típicas que se suelen utilizar en un lenguaje de programación orientado a objetos y en un lenguaje funcional. La asignatura tiene una fuerte componente práctica y dedica un número importante de clases a prácticas de laboratorio, donde se resuelven problemas clásicos con el uso de las estructuras de datos fundamentales.

COMPETENCIAS

1 Competencias generales y básicas.

BÁSICAS

CB02 Que los estudiantes sepan aplicar sus conocimientos a su trabajo o vocación de una forma profesional y posean las competencias que suelen demostrarse por medio de la elaboración y defensa de argumentos y la resolución de problemas dentro de su área de estudio.

CB04 Que los estudiantes puedan transmitir información, ideas, problemas y soluciones a un público tanto especializado como no especializado.

CB05 Que los estudiantes hayan desarrollado aquellas habilidades de aprendizaje necesarias para emprender estudios posteriores con un alto grado de autonomía.

GENERALES

CG08 Conocimiento de las materias básicas y tecnologías, que capaciten para el aprendizaje y desarrollo de nuevos métodos y tecnologías, así como las que les doten de una gran versatilidad para adaptarse a nuevas situaciones.

CG09 Capacidad para resolver problemas con iniciativa, toma de decisiones, autonomía y creatividad. Capacidad para saber comunicar y transmitir los conocimientos, habilidades y destrezas de la profesión de Ingeniero Técnico en Informática

2 Competencias específicas.

Formacion Comun

CC05	Conocimiento, administración y mantenimiento sistemas, servicios y aplicaciones informáticas.
CC06	Conocimiento y aplicación de los procedimientos algorítmicos básicos de las tecnologías informáticas para diseñar soluciones a problemas, analizando la idoneidad y complejidad de los algoritmos propuestos
CC07	Conocimiento, diseño y utilización de forma eficiente los tipos y estructuras de datos más adecuados a la resolución de un problema.
CC08	Capacidad para analizar, diseñar, construir y mantener aplicaciones de forma robusta, segura y eficiente, eligiendo el paradigma y los lenguajes de programación más adecuados.
CC17	Capacidad para diseñar y evaluar interfaces persona computador que garanticen la accesibilidad y usabilidad a los sistemas, servicios y aplicaciones informáticas.

CONTENIDOS DE LA ASIGNATURA

Introducción a la programación funcional

El estilo de programación funcional. Introducción a Haskell.

Características habituales de los lenguajes funcionales

Polimorfismo. Parcialización. Tipos Algebraicos. Funciones de orden superior. Listas por comprensión. Sobrecarga y clases de tipos. Módulos. Pruebas con QuickCheck.

Introducción a los tipos abstractos. Estructuras básicas y representaciones lineales.

Tipos de datos concretos y tipos de datos abstractos. Especificación de un tipo abstracto. Pilas, colas, colas con prioridad, listas, conjuntos, multiconjuntos y diccionarios. Iteradores.

Árboles

Árboles, árboles binarios, auténticos, perfectos y completos. Recorrido de árboles binarios. Montículos. Montículos binarios. Montículos zurdos. Colas de prioridad con montículos. Árboles binarios de búsqueda. Árboles equilibrados. Árboles AVL.

Tablas hash

Hashing. Colisiones. Funciones hash. Congruencia modular y polinómica. Resolución de colisiones. Encadenamiento separado y sondeo lineal. Factor de carga, rendimiento y reubicación (rehashing).

Grafos

Grafos dirigidos y no dirigidos. Terminología y propiedades fundamentales. Representación e implementaciones. Exploración de grafos. Exploración en profundidad y en anchura. Propiedades de los algoritmos de exploración. Aplicaciones: conexión, detección de ciclos, coloreado, orden topológico.

ACTIVIDADES FORMATIVAS

Actividades presenciales

Actividades expositivas

Lección magistral

Actividades prácticas en instalaciones específicas

Prácticas en laboratorio

Actividades no presenciales

Actividades de discusión, debate, etc.

Discusiones

Actividades de documentación

Búsqueda bibliográfica/documental

Actividades prácticas

Resolución de problemas

Estudio personal

Estudio personal

ACTIVIDADES DE EVALUACIÓN

Actividades de evaluación presenciales

Actividades de evaluación de la asignatura con participación alumnos

Otras actividades eval.asignatura

RESULTADOS DE APRENDIZAJE / CRITERIOS DE EVALUACIÓN

Tras superar la asignatura el estudiante será capaz de:

- RA1. Entender los conceptos de tipo abstracto de datos y de tipo de dato algebraico.
- RA2. Aplicar las distintas formas de representación de datos algebraicos y de los tipos abstractos de datos, en un lenguaje orientado a objetos y en un lenguaje funcional.
- RA3. Aplicar las técnicas fundamentales de diseño de algoritmos utilizando el paradigma funcional y orientado a objetos, valorando las ventajas e inconvenientes de las distintas soluciones.
- RA4. Identificar las fortalezas y debilidades del paradigma de programación funcional
- RA5. Identificar patrones de cómputo, abstraerlos e implementarlos usando funciones de orden superior y polimorfismo
- RA6. Diseñar, codificar, probar y depurar algoritmos usando el paradigma funcional

Las competencias generales CG8 y CG9 están parcialmente cubiertas. Por un lado, se estudian dos paradigmas de programación y las estructuras de datos fundamentales, lo que sirve de base para un posterior estudio más avanzado (CG8). Por otro lado, la evaluación continua incluye prácticas de laboratorio que suponen la resolución de problemas con cierto grado de autonomía y creatividad (CG9).

Los resultados de aprendizaje están claramente enfocados a cubrir las siguientes competencias específicas:

- CC06 Conocimiento y aplicación de los procedimientos algorítmicos básicos de las tecnologías informáticas para diseñar soluciones a problemas, analizando la idoneidad y complejidad de los algoritmos propuestos
- CC07 Conocimiento, diseño y utilización de forma eficiente los tipos y estructuras de datos más adecuados a la resolución de un problema
- CC08 Capacidad para analizar, diseñar, construir y mantener aplicaciones de forma robusta, segura y eficiente, eligiendo el paradigma y los lenguajes de programación más adecuados

Dada la temática de esta asignatura, solo es posible cubrir tangencialmente las competencias específicas CC05 (Conocimiento, administración y mantenimiento sistemas, servicios y aplicaciones informáticas) y CC17 (Capacidad para diseñar y evaluar interfaces persona computador que garanticen la accesibilidad y usabilidad a los sistemas, servicios y aplicaciones informáticas). Obviamente, la simplicidad y flexibilidad de los diseños y, muy especialmente, la eficiencia de las aplicaciones que requieren de estas competencias depende en buena medida del adecuado uso de las estructuras de datos.

Los criterios de evaluación de los resultados de aprendizaje incluyen la resolución de prácticas de laboratorio en las que el estudiante deberá implementar estructuras de datos en ambos lenguajes, determinar la complejidad y corrección de su implementación, y resolver problemas aplicando las estructuras de datos estudiadas. Durante la resolución de estas prácticas el estudiante contará con la supervisión del docente. Además, el estudiante deberá realizar un examen final con actividades similares a las desarrolladas durante las prácticas de laboratorio.

PROCEDIMIENTO DE EVALUACIÓN

I. Evaluación en la primera convocatoria ordinaria:

La evaluación en esta convocatoria es continua y, por lo tanto, la asistencia a las clases de teoría y a los laboratorios es obligatoria. Las posibles faltas puntuales deberán ser justificadas. La evaluación continua consta de las siguientes 2 componentes:

PE - Dos prácticas de laboratorio evaluables, con un valor total de 2,5 puntos
EF - Examen final, con dos partes con ejercicios en Haskell y Java, respectivamente, y un valor total de 7,5 puntos

Para optar al aprobado, el alumno debe obtener al menos 2,5 puntos sobre 10 en cada parte del examen final. Si es así, la nota final será la suma de las anteriores componentes (PE + EF). Para aprobar, esta suma debe ser igual o superior a 5 puntos.

Los alumnos a tiempo parcial y deportistas de alto rendimiento que no puedan seguir el proceso de evaluación continua serán evaluados exclusivamente por un examen final. Este examen final constará de dos partes con ejercicios en Haskell y Java, respectivamente. Para optar al aprobado, el alumno debe obtener al menos 2,5 puntos sobre 10 en cada parte. Si es así, la nota final será la media aritmética de las calificaciones de ambas partes del examen final. Para aprobar, esta calificación final debe ser igual o superior a 5 puntos.

II. Evaluación en la segunda convocatoria ordinaria:

Si el alumno ha seguido el proceso de evaluación continua y desea conservar las puntuaciones de la componente continua, se aplicarán las normas de la primera convocatoria ordinaria.

El resto de alumnos serán evaluados exclusivamente por examen final, aplicándose las normas de la primera convocatoria ordinaria para alumnos que no han seguido el proceso de evaluación continua.

III. Evaluación en convocatorias extraordinarias:

La evaluación se realizará exclusivamente por examen final, aplicándose las normas de la primera convocatoria ordinaria para alumnos que no han seguido el proceso de evaluación continua.

BIBLIOGRAFÍA Y OTROS RECURSOS

Básica

Algorithms: a functional programming approach; Fethi Rabbi y Guy Lapalme; Addison; 1999

Algorithms; Robert Sedgewick y Kevin Wayne; Addison-Wesley; 2011

Data Structures & Algorithms in Java; Michael T. Goodrich y Roberto Tamassia; John Wiley & Sons; 2006

Introducción a la Programación Funcional; Richard Bird; 2000
Introduction to Algorihms; T.H. Cormen, C.E. Leiserson, R.L. Rivest y C. Stein; McGraw-Hill; 2010
Material elaborado para la asignatura (transparencias, enunciados, código fuente...) disponible en el Campus Virtual
Razonando con Haskell; Blas Ruiz, Fco. Gutiérrez, Pablo Guerrero y José Gallardo; Thomson; 2004

DISTRIBUCIÓN DEL TRABAJO DEL ESTUDIANTE

ACTIVIDAD FORMATIVA PRESENCIAL

Descripción	Horas	Grupo grande	Grupos reducidos
Lección magistral	41,4	☑	☐
Prácticas en laboratorio	18,6	☐	☑
TOTAL HORAS ACTIVIDAD FORMATIVA PRESENCIAL	**60**		

ACTIVIDAD FORMATIVA NO PRESENCIAL

Descripción	Horas
Resolución de problemas	30
Búsqueda bibliográfica/documental	5
Discusiones	5
Estudio personal	30
TOTAL HORAS ACTIVIDAD FORMATIVA NO PRESENCIAL	**75**
TOTAL HORAS ACTIVIDAD EVALUACIÓN	**15**
TOTAL HORAS DE TRABAJO DEL ESTUDIANTE	**150**