



## 1. Identificación

### 1.1. De la Asignatura

Curso Académico	2023/2024
Titulación	GRADO EN INGENIERÍA INFORMÁTICA y PROGRAMA ACADÉMICO DE SIMULTANEIDAD DE DOBLE TITULACIÓN CON ITINERARIO ESPECÍFICO DE GRADO EN MATEMÁTICAS Y GRADO EN INGENIERÍA INFORMÁTICA
Nombre de la Asignatura	ALGORITMOS Y ESTRUCTURAS DE DATOS I
Código	1895
Curso	SEGUNDO y SEGUNDO(IC)
Carácter	OBLIGATORIA
N.º Grupos	4
Créditos ECTS	6
Estimación del volumen de trabajo del alumno	150
Organización Temporal/Temporalidad	1 Cuatrimestre y 1 Cuatrimestre(IC)
Idiomas en que se imparte	ESPAÑOL : Grupo 1,2,3,9(IC)

### 1.2. Del profesorado: Equipo Docente

Coordinación de la asignatura  NORBERTO MARIN PEREZ	Área/Departamento	LINGÜAJES Y SISTEMAS INFORMÁTICOS/INFORMÁTICA Y SISTEMAS
	Categoría	PROFESORES TITULARES DE ESCUELAS UNIVERSITARIAS
	Correo Electrónico /	nmarin@um.es
	Página web / Tutoría electrónica	http://webs.um.es/nmarin/ Tutoría Electrónica: SÍ



Grupo de Docencia:	Teléfono, Horario y Lugar de atención al alumnado	Duración	Día	Horario	Lugar	Observaciones
1, 2, 3 y 9 Coordinación de los grupos:1,2,3 y 9(IC)		Primer Cuatrimestre	Martes	11:05- 12:05	868884637, Facultad de Informática B1.2.035	despacho 2.27
		Primer Cuatrimestre	Miércoles	17:35- 18:35	868884637, Facultad de Informática B1.2.035	despacho 2.27
		Segundo Cuatrimestre	Martes	16:40- 17:25	868884637, Facultad de Informática B1.2.035	despacho 2.27
		Segundo Cuatrimestre	Martes	10:10- 10:55	868884637, Facultad de Informática B1.2.035	despacho 2.27
		Segundo Cuatrimestre	Miércoles	10:10- 10:55	868884637, Facultad de Informática B1.2.035	despacho 2.27
		Segundo Cuatrimestre	Miércoles	16:40- 17:25	868884637, Facultad de Informática B1.2.035	despacho 2.27
		GINES GARCIA MATEOS	Área/Departamento	LENGUAJES Y SISTEMAS INFORMÁTICOS/INFORMÁTICA Y SISTEMAS		
	Categoría	CATEDRATICOS DE UNIVERSIDAD				
Grupo de Docencia: 1, 2, 3 y 9	Correo Electrónico / Página web / Tutoría electrónica	ginesgm@um.es <a href="http://dis.um.es/~ginesgm">http://dis.um.es/~ginesgm</a> Tutoría Electrónica: Sí				



	Teléfono, Horario y Lugar de atención al alumnado	Duración	Día	Horario	Lugar	Observaciones
		Anual	Martes	10:20- 12:00	868888530, Facultad de Informática B1.2.044	Despacho 2.34
		Anual	Jueves	10:50- 12:10	868888530, Facultad de Informática B1.2.044	Despacho 2.34
FRANCISCO JOSE MONTROYA DATO	Área/Departamento	LENGUAJES Y SISTEMAS INFORMÁTICOS/INFORMÁTICA Y SISTEMAS				
Grupo de Docencia: 1, 2, 3 y 9	Categoría	PROFESORES TITULARES DE UNIVERSIDAD				
	Correo Electrónico / Página web / Tutoría electrónica	fmontoya@um.es <a href="https://webs.um.es/fmontoya">https://webs.um.es/fmontoya</a> Tutoría Electrónica: NO				
	Teléfono, Horario y Lugar de atención al alumnado	Duración	Día	Horario	Lugar	Observaciones
		Anual	Lunes	05:00- 05:10	868884620, Facultad de Informática B1.2.008	Visítese esta dirección (cópíela y péguela, no siga el enlace): <a href="https://webs.um.es/fmontoya/miwiki/doku.php?id=tutorias">https:// webs.um.es/ fmontoya/ miwiki/ doku.php? id=tutorias</a>
JESUS SANCHEZ CUADRADO	Área/Departamento	LENGUAJES Y SISTEMAS INFORMÁTICOS/INFORMÁTICA Y SISTEMAS				
	Categoría	PROFESORES TITULARES DE UNIVERSIDAD				



Grupo de Docencia: 1, 2, 3 y 9	Correo Electrónico / Página web / Tutoría electrónica	jesusc@um.es Tutoría Electrónica: Sí				
	Teléfono, Horario y Lugar de atención al alumnado	Duración Anual	Día Lunes	Horario 08:00- 12:00	Lugar (Sin Extensión), Facultad de Informática B1.2.027	Observaciones Solicitar cita previa
RUBEN FERNANDEZ BELTRAN Grupo de Docencia: 1, 2, 3 y 9	Área/Departamento	LENGUAJES Y SISTEMAS INFORMÁTICOS/INFORMÁTICA Y SISTEMAS				
	Categoría	PROFESOR PERMANENTE LABORAL				
	Correo Electrónico / Página web / Tutoría electrónica	rufernan@um.es Tutoría Electrónica: Sí				
	Teléfono, Horario y Lugar de atención al alumnado	Duración Anual	Día Lunes	Horario 16:30- 18:00	Lugar 868888527, Facultad de Informática B1.2.015	Observaciones Despacho 2.12
		Anual	Martes	16:30- 18:00	868888527, Facultad de Informática B1.2.015	Despacho 2.12

## 2. Presentación

Los algoritmos y las estructuras de datos constituyen los dos pilares básicos de la programación de ordenadores. Esta asignatura se puede ver como una continuación natural de las asignaturas de programación de primer curso. Partiendo de una base previa en los principios de la programación, se intenta desarrollar en



los alumnos las habilidades de resolución de problemas, con herramientas como los esquemas algorítmicos, el razonamiento inductivo, la descomposición modular, la aplicación de procesos metódicos, y en general el pensamiento algorítmico. Los conocimientos adquiridos en la asignatura resultan fundamentales en todas las asignaturas posteriores donde se requieran conocimientos de programación.

### 3. Condiciones de acceso a la asignatura

#### 3.1 Incompatibilidades

No consta

#### 3.2 Recomendaciones

Para cursar esta asignatura es altamente recomendable haber cursado y aprobado las asignaturas de programación de primer curso. En concreto, los alumnos deben saber programar con soltura en el lenguaje C o C++, ya que en caso contrario encontrarán muchas dificultades para superar la asignatura.

### 4. Competencias

#### 4.1 Competencias Básicas

No disponible

#### 4.2 Competencias de la titulación

- CGII1. Capacidad de análisis y síntesis.
- CGII7. Resolución de problemas.
- CEII2. Capacidad para dirigir las actividades objeto de los proyectos del ámbito de la informática de acuerdo con los conocimientos adquiridos.
- CR1. Capacidad para diseñar, desarrollar, seleccionar y evaluar aplicaciones y sistemas informáticos, asegurando su fiabilidad, seguridad y calidad, conforme a principios éticos y a la legislación y normativa vigente.
- CR6. 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.
- CR7. Conocimiento, diseño y utilización de forma eficiente los tipos y estructuras de datos más adecuados a la resolución de un problema.

#### 4.3 Competencias transversales y de materia

- Competencia 1. CR6: 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.



- Competencia 2. CR7: Conocimiento, diseño y utilización de forma eficiente los tipos y estructuras de datos más adecuados a la resolución de un problema.
- Competencia 3. CR8: Capacidad para analizar, diseñar, construir y mantener aplicaciones de forma robusta, segura y eficiente, eligiendo el paradigma y los lenguajes de programación más adecuados.

## 5. Contenidos

### TEMA 1. Abstracciones y Especificaciones

Introducción. Especificaciones informales. Especificaciones formales.

### TEMA 2. Conjuntos y diccionarios

Repaso del TAD Conjunto. Implementaciones básicas. El TAD Diccionario. Las tablas de dispersión.

Relaciones muchos a muchos.

### TEMA 3. Representación de conjuntos mediante árboles

Árboles Trie. Relaciones de equivalencia. Árboles de búsqueda balanceados. Árboles B.

### TEMA 4. Grafos

Introducción, notación y definiciones. Representación de grafos. Problemas y algoritmos sobre grafos.

## PRÁCTICAS

Práctica 1. 1. Especificaciones formales con Maude: Relacionada con los contenidos Tema 1

Práctica 2. 2-3. Implementación y manejo de estructuras de datos: Relacionada con los contenidos Tema 2 y Tema 3

Práctica 3. 4. Solución de problemas con grafos: Relacionada con los contenidos Tema 4

## 6. Metodología Docente

Actividad Formativa	Metodología	Horas Presenciales	Trabajo Autónomo	Volumen de trabajo
Teoría		24	36	60
Seminarios		7.2	10.8	18
Prácticas		24.8	32.2	57
Exámenes		4	11	15
	Total	60	90	150



## 7. Horario de la asignatura

<https://www.um.es/web/estudios/grados/informatica/2023-24#horarios>

## 8. Sistema de Evaluación

Métodos / Instrumentos	Examen teórico-práctico. En este instrumento incluimos desde el tradicional examen escrito o tipo test hasta los exámenes basados en resolución de problemas, pasando por los de tipo mixto que incluyen cuestiones cortas o de desarrollo teórico junto con pequeños problemas. También se incluye aquí la consideración de la participación activa del alumno en clase, la entrega de ejercicios o realización de pequeños trabajos escritos y presentaciones.
Criterios de Valoración	<p>Los alumnos pueden optar por realizar la evaluación de la teoría mediante el sistema de evaluación continua, o bien mediante el sistema alternativo de examen final.</p> <p>1. La evaluación continua consistirá en una serie de actividades a lo largo del curso para cada tema, y requerirá un mínimo de asistencias a clase (se tendrán en cuenta situaciones especiales particulares). Si algún tema no es superado, se podrá recuperar en el examen final. Los contenidos, criterios y temporización de las actividades de evaluación continua se presentarán el primer día de clase.</p> <p>2. En el sistema alternativo la nota de teoría se limitará al examen final, no siendo necesario hacer las actividades de evaluación continua.</p> <p>En cualquier caso, la nota final de la asignatura se calculará como: 60% de la nota de teoría (todos los temas tienen el mismo peso), más 40% de la nota de prácticas, siempre que ambas partes estén aprobadas.</p> <p>Además, la nota final, una vez aprobada la asignatura, podrá verse incrementada hasta en un punto por participación en clase o por realización de soluciones programadas de ejercicios propuestos.</p>
Ponderación	60



Métodos / Instrumentos	Informe técnico. En este instrumento incluimos los resultados de actividades prácticas, o de laboratorio, junto con sus memorias descriptivas. Los resúmenes del estado del arte o memorias de investigación sobre temas concretos. Y la posibilidad de realizar entrevistas personales o presentaciones de los trabajos realizados también entran en esta categoría.
Criterios de Valoración	Práctica en grupos de 2 alumnos, realizada en C++ usando el juez online Mooshak sobre el contenido de los temas 2 y 3. Se evalúa sobre 10 y se requiere un 5 para realizar media con la teoría. Puede requerirse entrevista con los alumnos.
Ponderación	40

## Fechas de exámenes

<https://www.um.es/web/estudios/grados/informatica/2023-24#exámenes>

## 9. Resultados del Aprendizaje

- Entender el desarrollo de programas como un proceso metódico e ingenieril, formado por una serie de etapas con distintos niveles de abstracción, frente a la idea de la programación como arte.
- Reconocer la importancia de la abstracción y conocer los tipos de abstracciones que aparecen en programación: funcional, de datos y de iteradores.
- Concienciarse de la utilidad de desarrollar especificaciones completas y precisas, entendiendo la especificación como un punto de acuerdo entre el usuario y el implementador de una abstracción.
- Comprender el método de especificación formal algebraico o axiomático (basado en una definición mediante axiomas) y el método constructivo u operacional (basado en el uso de precondiciones y postcondiciones).
- Conocer la importancia y ubicuidad de los tipos conjunto y diccionario en el desarrollo de programas, independientemente de la estructura que se use para implementarlos.
- Ser capaz de diseñar, implementar y analizar la eficiencia de las principales estructuras de representación no arbóreas para los tipos conjunto y diccionario, adaptando el diseño a las necesidades específicas de cada aplicación.
- Conocer la estructura de datos de tablas de dispersión, sus distintas variantes y los factores que influyen en su eficiencia y uso de memoria.



- Conocer y comprender una variedad de técnicas eficientes de representación de conjuntos y diccionarios mediante estructuras arbóreas.
- Adquirir la capacidad de evaluar las necesidades de representación de una aplicación específica, tomando decisiones justificadas sobre las estructuras de representación más adecuadas.
- Comprender la necesidad de usar mecanismos de equilibrado o balanceo para conseguir eficiencia en las representaciones arbóreas.
- Ser capaz de diseñar e implementar una estructura de datos para el tipo grafo (en sus distintas variantes) usando listas y matrices de adyacencia.
- Valorar críticamente las ventajas e inconvenientes de las representaciones de grafos mediante listas y matrices de adyacencia, y su influencia en la eficiencia de los algoritmos sobre grafos.
- Conocer y comprender el funcionamiento de una variedad de algoritmos clásicos sobre grafos (tales como los algoritmos de Prim, Kruskal, Dijkstra, Floyd y Warshall), razonando sobre las ideas subyacentes que aportan y analizando su complejidad computacional.
- Ser capaz de usar los algoritmos estudiados como herramientas prácticas para la resolución de problemas en un contexto genérico, a través de la transformación de un problema de interés en un problema sobre grafos.

## 10. Bibliografía

### Bibliografía Básica



García Mateos, Ginés, et al. -- Algoritmos y estructuras de datos, Diego Marín (2003)



Aho, Alfred V. -- Estructuras de datos y algoritmos (1998)



Heileman, Gregory L. -- Estructuras de datos, algoritmos, programación orientada a o (1998)



Horebeek, Ivo Van, et al. -- Algebraic specifications in software engineering: an introduction (1989)



Kernighan, Brian W. -- El lenguaje de programación C (1991)



Martí Oliet, Narciso -- Estructuras de datos y métodos algorítmicos : ejercicios res (2004)



Peña Marí, Ricardo -- Diseño de programas : formalismo y abstracción (1998)



Stroustrup, Bjarne -- El lenguaje de programación C++ (1998)



Weiss, Mark Allen -- Estructuras de datos y algoritmos (1995)



Wirth, Niklaus -- Algoritmos estructuras de datos = programas (1994)



Cormen, T.H., et al. -- Introduction to algorithms, second edition. The MIT Press (2001)



Mehlhorn, K -- Data Structures and algorithms, (1984)



Brassard, Gilles -- Fundamentos de algoritmia (1997)



Baase, Sara -- Computer algorithms : introduction to design and analysis (2000)

## Bibliografía Complementaria



Joyanes Aguilar, Luis., Estructura de datos : algoritmos, abstracción y objetos /(1998) ,McGraw-Hill,

## 11. Observaciones y recomendaciones

Criterio suspenso/no presentado:

Se pueden dar los siguientes casos:

1. Si un alumno suspende una parte (teoría o prácticas) y no se presenta a la otra parte, la calificación total será de Suspenso.
2. Si un alumno aprueba una parte y no presenta la otra parte, la calificación será de No presentado.
3. Si un alumno aprueba una parte y suspende la otra, la calificación será de Suspenso.

Cuando la calificación sea Suspenso, la nota se calculará como la media ponderada de la nota de teoría (60%) y de prácticas (40%), que se redondeará a 4,5 si dicha media fuera superior a 4,5.

El criterio para decidir si la nota de teoría está presentada o no, es la presentación en el examen final de la asignatura. Por lo tanto, un alumno que siga el método de evaluación continua pero no se presente al examen final para recuperar algún tema pendiente, se considerará como no presentado.

Las notas de partes aprobadas se guardarán durante las tres convocatorias del curso académico.

Otras actividades: olimpiada de programación:



Como actividad complementaria y voluntaria (aunque muy recomendable para los alumnos de la asignatura), se preparará a los alumnos para participar en el concurso internacional de programación ACM International Collegiate Programming Contest. Para ello se ofrecerá una preparación específica y habrá un concurso de programación (Olimpiadas Murcianas de Programación) a nivel local. Los alumnos ganadores conseguirán la participación en el concurso internacional (ACM International Collegiate Programming Contest, SWERC), que será considerado como un viaje de prácticas de la asignatura.

#### FECHAS APROXIMADAS DE LAS ACTIVIDADES EVALUATORIAS:

Ev. continua: Tema 1. Actividad de especificaciones formales: entrega 5ª semana

Ev. continua: Temas 2 y 3. Examen teórico/práctico: 10ª semana

Práctica en C/C++ sobre conjuntos y árboles: 12ª semana

Ev. continua: Tema 4. Actividad entregable realizada en C/C++ sobre el contenido del tema 4 (grafos): 15ª semana

“NECESIDADES EDUCATIVAS ESPECIALES. Aquellos estudiantes con discapacidad o necesidades educativas especiales podrán dirigirse al Servicio de Atención a la Diversidad y Voluntariado (ADYV; <http://www.um.es/adyv/>) para recibir orientación sobre un mejor aprovechamiento de su proceso formativo y, en su caso, la adopción de medidas de equiparación y de mejora para la inclusión, en virtud de la Resolución Rectoral R-358/2016. El tratamiento de la información sobre este alumnado, en cumplimiento con la normativa vigente, es de estricta confidencialidad.”

Esta asignatura no se encuentra vinculada de forma directa con los Objetivos de Desarrollo Sostenible.