



## **GUÍA DOCENTE**

# **ANÁLISIS Y DISEÑO DE ALGORITMOS**

## **GRADO EN INGENIERÍA DEL SOFTWARE**

***MODALIDAD: A DISTANCIA***

***CURSO ACADÉMICO: 2023-2024***

Denominación de la asignatura:	<b>Análisis y Diseño de Algoritmos</b>
Titulación:	Ingeniería del Software
Facultad o Centro:	Centro Universitario de Tecnología y Arte Digital
Materia:	Programación
Curso:	2º
Cuatrimestre:	2
Carácter:	OB
Créditos ECTS:	6
Modalidad de enseñanza:	A distancia
Idioma:	Castellano
Profesor / Email:	Gonzalo Bailador / gonzalo.bailador@u-tad.com Gonzalo Nicolás Barreales / gonzalo.nicolas@u-tad.com Jesús González Álvaro / jesus.alvaro@u-tad.com Gonzalo N. Barreales/ gonzalo.nicolas@u-tad.com
Página Web:	<a href="http://www.u-tad.com/">http://www.u-tad.com/</a>

## DESCRIPCIÓN DE LA ASIGNATURA

### Descripción de la materia

Esta asignatura pertenece a la materia de programación. Esta materia se dedica al estudio de las técnicas y los lenguajes de programación en los que se fundamentarán los estudios del grado de ingeniería del software.

### Descripción de la asignatura

En esta asignatura se estudia el análisis y diseño de algoritmos desde un punto de vista teórico y práctico.

Se profundizará en el análisis de la complejidad temporal y espacial de un algoritmo y las implicaciones de estos resultados en la ejecución de este.

Se estudiarán diferentes algoritmos para la resolución de problemas variados enfatizando, en todo momento, las ventajas y desventajas de cada uno de ellos, así como los contextos en los que debe ser aplicado.

Adicionalmente se realizarán casos de estudio en los que se programarán los algoritmos y se comprobará su eficacia a la hora de resolver problemas

## COMPETENCIAS Y RESULTADOS DE APRENDIZAJE

### Competencias (genéricas, específicas y transversales)

#### COMPETENCIAS BÁSICAS Y GENERALES:

CG1 - Capacidad para entender, planificar y resolver problemas a través del desarrollo de soluciones informáticas

CG3 - Conocimiento de los fundamentos científicos aplicables a la resolución de problemas informáticos

CG4 - Capacidad para simplificar y optimizar los sistemas informáticos atendiendo a la comprensión de su complejidad

CG9 - Capacidad para aprender, modificar y producir nuevas tecnologías informáticas

CG10 - Uso de técnicas creativas para la realización de proyectos informáticos

CB1 - Que los estudiantes hayan demostrado poseer y comprender conocimientos en un área de estudio que parte de la base de la educación secundaria general, y se suele encontrar a un nivel que, si bien se apoya en libros de texto avanzados, incluye también algunos aspectos que implican conocimientos procedentes de la vanguardia de su campo de estudio

CB2 - 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

CB3 - Que los estudiantes tengan la capacidad de reunir e interpretar datos relevantes (normalmente dentro de su área de estudio) para emitir juicios que incluyan una reflexión sobre temas relevantes de índole social, científica o ética

CB4 - Que los estudiantes puedan transmitir información, ideas, problemas y soluciones a un público tanto especializado como no especializado

CB5 - Que los estudiantes hayan desarrollado aquellas habilidades de aprendizaje necesarias para emprender estudios posteriores

con un alto grado de autonomía

#### COMPETENCIAS ESPECÍFICAS

CE1 - Conocimiento de la estructura de las computadoras, de los conceptos de codificación, manipulación, tratamiento de la información y uso de lenguajes de bajo nivel

CE7 - Conocimiento de los principales tipos de estructuras de datos y utilización de las librerías y de los técnicas algorítmicas asociadas a dichas estructuras junto con los órdenes de complejidad que caracterizan a dichas técnicas

CE8 - Conocimiento de los distintos paradigmas detrás de los lenguajes de programación

CE9 - Conocimiento de las estructuras de control, variables, sintaxis de programación y gestión del uso de la memoria de manera eficaz en el desarrollo de una aplicación informática

CE10 - Capacidad para manejar un gestor de versiones de código y generar la documentación de una aplicación de forma automática.

CE15 - Conocimiento de la tolerancia a los fallos, la adaptabilidad, el balance de carga y la predictividad del sistema para el desarrollo de aplicaciones distribuidas

CE17 - Conocimiento de las características de paralelización de tarjetas gráficas y de arquitecturas de altas prestaciones para el desarrollo de aplicaciones.

CE20 - Capacidad para testar el funcionamiento y funcionalidad de una aplicación informática, elaborando planes de pruebas y empleando técnicas de diseño y programación orientado a las pruebas

CE23 - Conocimiento de los principios de la inteligencia artificial y uso de algoritmos de búsqueda deterministas y máquinas de estado

### **Resultados de aprendizaje**

Al acabar la titulación, el graduado o graduada será capaz de:

- Entender y manejar el concepto de memoria dinámica
- Identificar clases de objetos con los datos de un problema.
- Crear clases y objetos y manipularlos.
- Entender y utilizar los mecanismos de herencia, polimorfismo y sobrecarga de operadores.
- Identificar las relaciones entre clases en distintos casos de uso.
- Dominar un lenguaje de programación orientado a objetos.
- Dominar los patrones de programación
- Conocer las distintas formas de resolución de problemas desde el punto de vista de la algoritmia, como, por ejemplo, el esquema divide y vencerás,
- programación dinámica, backtracking o algoritmos genéticos.
- Estudiar la complejidad de un determinado algoritmo, interpretar dicha complejidad y analizar posibles optimizaciones.
- Codificar un programa que sea capaz de encontrar el camino óptimo que une dos nodos de un grafo siguiendo los distintos algoritmos de pathfinding.
- Crear y entrenar redes neuronales que solucionen problemas concretos.

## **CONTENIDO**

Algoritmos de grafos

Algoritmos de ordenación

Árboles de búsqueda

Análisis de amortización algorítmica

Esquemas algorítmicos: Backtracking, divide y vencerás, programación dinámica

## TEMARIO

Unidad 1: Construcción de algoritmos y eficiencia

Unidad 2: Algoritmos recursivos

Unidad 3: Listas, pilas y colas

Unidad 4: Ordenación de listas

Unidad 5: Tablas hash

Unidad 6: Árboles y algoritmos genéticos

## ACTIVIDADES FORMATIVAS Y METODOLOGÍAS DE APRENDIZAJE

### Actividades formativas

Actividad Formativa	Horas totales	Horas síncronas
<i>Sesiones teóricas virtuales síncronas</i>	4,36	4
<i>Sesiones teóricas virtuales asíncronas</i>	27,64	0
<i>Sesiones prácticas virtuales síncronas</i>	2,18	2
<i>Sesiones prácticas virtuales asíncronas</i>	10,18	0
<i>Debate y discusión oral y/o escrita.</i>	7,27	0
<i>Tutorías</i>	4,00	4
<i>Estudio independiente y trabajo autónomo del alumno</i>	51,64	0
<i>Elaboración de trabajos (en grupo o individuales)</i>	33,09	0
<i>Actividades de Evaluación</i>	4,36	4
<i>Test de autoevaluación</i>	5,27	0
<i>Prácticas externas</i>	0,00	0
<i>Preparación y defensa virtual del TFG</i>	0,00	0
<i>Seguimiento de proyectos</i>	0,00	0

TOTAL	150	14
-------	-----	----

### Metodologías docentes

Método expositivo o lección magistral

Aprendizaje de casos

Aprendizaje basado en la resolución de problemas

Aprendizaje cooperativo o colaborativo

Aprendizaje por indagación

Metodología Flipped classroom o aula invertida

Gamificación

Just in time Teaching (JITT) o aula a tiempo

Método expositivo o lección magistral

Método del caso

Aprendizaje basado en la resolución de problemas

Aprendizaje cooperativo o colaborativo

Aprendizaje por indagación

Metodología flipped classroom o aula invertida

Gamificación

### DESARROLLO TEMPORAL

Presentación - semana 1

Unidad 1 - semana 2-3

Unidad 2 - semana 4-5

Unidad 3 - semana 6-7

Unidad 4 - semana 7-8

Unidad 5 - semana 9-10

Unidad 6 - semana 11-12

Repaso - semana 13-14

Evaluación - semana 15

### SISTEMA DE EVALUACIÓN

ACTIVIDAD DE EVALUACIÓN	VALORACIÓN MÍNIMA RESPECTO A LA CALIFICACIÓN FINAL (%)	VALORACIÓN MÁXIMA RESPECTO A LA CALIFICACIÓN FINAL (%)
<i>Evaluación de la participación en clase, en prácticas o en proyectos de la asignatura</i>	10	20
<i>Evaluación de trabajos, proyectos, informes, memorias</i>	10	20
<i>Prueba Objetiva</i>	60	70

## CRITERIOS ESPECÍFICOS DE EVALUACIÓN

ACTIVIDAD DE EVALUACIÓN	CONVOCATORIA ORDINARIA	CONVOCATORIA EXTRAORDINARIA
<i>Evaluación de la participación en clase, en prácticas o en proyectos de la asignatura</i>	20	10
<i>Evaluación de trabajos, proyectos, informes, memorias</i>	20	20
<i>Prueba Objetiva</i>	60	70

### Consideraciones específicas acerca de la evaluación

Será necesario que obtener una nota mínima de 4 puntos (sobre 10) en la prueba final presencial para que se realice la media con las actividades formativas.

## BIBLIOGRAFÍA / WEBGRAFÍA

Básica:

Hernández, R., Lázaro, J.C., Dormido, R. y Ros, S. (2001). Estructuras de Datos y Algoritmos. Ed. Prentice Hall. ISBN: 978-84-205-2980-6

Martí, N., Ortega, Y. y Verdejo, J.A. (2003). Estructuras de datos y métodos algorítmicos: ejercicios resueltos. Ed. Pearson Alhambra.

Rodríguez, M. y Gonzalo, J. (1998). Esquemas algorítmicos: enfoque metodológico y problemas resueltos. Ed. UNED. ISBN: 9788436236224.

Hernández, R., Carmona, E.J., Martínez, R. y Pastor, R. (2006). Problemas de estructuras de datos y algoritmos. Ed. Ramón Areces. ISBN: 84-8004-723-2.

Murphy, J. (2005). El inversor visual. Ed. Netbiblo. ISBN: 978-8497451017

Recomendada:

Guerrequeta, R. y Vallecillo, A. (1998). Técnicas de diseño de algoritmos. Ed. Universidad de Málaga.

Brassard, G. y Bratley, P. (1997). Fundamentos de Algoritmia. Ed. Prentice Hall. ISBN: 0-13-335068-1.

Edwards, R. D. y Magee J. (2010). Análisis técnico de las tendencias de las acciones. Ed. BN Publishing. ISBN: 978-1607960799

Serafini, M.T. (2007). Cómo se escribe. Ed. Paidós. ISBN: 978-84-493-1959-4. Es para aprender a redactar, por si algún alumno lo necesita.

## **MATERIALES, SOFTWARE Y HERRAMIENTAS NECESARIAS**

### **Materiales:**

PC personal, con windows, linux o Mac

### **Software:**

Para acostumbrarse al uso de las herramientas que se dispondrán en el examen, se aconseja el siguiente entorno:

Terminal posix: Cygwin

Compilador gcc/g++, ejecutado por terminal

Editor Notepad++

En general, cualquier entorno de programación que soporte C