

**CENTRO UNIVERSITARIO DE TECNOLOGÍA Y ARTE DIGITAL**



# **PLANIFICACIÓN DE LA DOCENCIA UNIVERSITARIA**

## **GUÍA DOCENTE**

### **Análisis y Diseño de Algoritmos**

# 1. DATOS DE IDENTIFICACIÓN DE LA ASIGNATURA.

Título:	Grado en Ingeniería del Software
Facultad:	Centro Universitario de Tecnología y Arte Digital (U-TAD)
Materia:	Programación
Denominación de la asignatura:	Análisis y Diseño de Algoritmos
Curso:	Segundo
Cuatrimestre:	Segundo
Carácter:	Básica
Créditos ECTS:	6
Modalidad/es de enseñanza:	Híbrida Presencial
Idioma:	Castellano
Profesor/a:	Gonzalo Bailador Rafael Vida Marcos Novalbos
E-mail:	<a href="mailto:marcos.novalbos@u-tad.com">marcos.novalbos@u-tad.com</a> <a href="mailto:gonzalo.bailador@u-tad.com">gonzalo.bailador@u-tad.com</a> <a href="mailto:rafael.vida@u-tad.com">rafael.vida@u-tad.com</a>
Teléfono:	

## 2. DESCRIPCIÓN DE LA ASIGNATURA.

### 2.1 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.

### 2.2 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 del mismo. 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. )

### 3.COMPETENCIAS

#### Competencias Básicas y Generales

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

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

#### 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

## 4. CONTENIDOS

Complejidad Computacional, Análisis de amortización algorítmica.  
Algoritmos de ordenación.  
Esquemas algorítmicos: backtracking, divide y vencerás, programación dinámica  
Algoritmos y estructuras de datos  
Árboles de búsqueda  
Montículos, Hashing  
Algoritmos de grafos

## 5. DESARROLLO TEMPORAL

UNIDADES DIDÁCTICAS / TEMAS	PERÍODO TEMPORAL
Introducción / Revisión C++	1 semana
Clases / Algoritmos no recursivos	1 semana
Ordenación	1 semana
Recursividad	1 semana
Esquemas algorítmicos: backtracking, divide y vencerás, programación dinámica	2 semanas
Listas Contiguas/ Enlazadas	1 semana
Listas Circulares / Pilas / Colas	1 semana
Tablas Hash	1 semanas
Arboles	2 semanas
Grafos	1 semana

# 6.ACTIVIDADES FORMATIVAS Y MODALIDADES DE ENSEÑANZAS

## 6.1 Modalidades de enseñanza

La asignatura se desarrollará a través de los siguientes métodos y técnicas generales, que se aplicarán diferencialmente según las características propias de la asignatura:

- **Método expositivo/Lección magistral:** el profesor desarrollará, mediante clases magistrales y dinámicas los contenidos recogidos en el temario.
- **Estudio de casos:** análisis de casos reales relacionados con la asignatura.
- **Resolución de ejercicios y problemas:** los estudiantes desarrollarán las soluciones adecuadas aplicando procedimientos de transformación de la información disponible y la interpretación de los resultados.
- **Aprendizaje basado en problemas:** utilización de problemas como punto de partida para la adquisición de conocimientos nuevos.
- **Aprendizaje orientado a proyectos:** se pide a los alumnos que, en pequeños grupos, planifiquen, creen y evalúen un proyecto que responda a las necesidades planteadas en una determinada situación.
- **Aprendizaje cooperativo:** Los estudiantes trabajan en grupo para realizar las tareas de manera colectiva.

## 6.2 Actividades formativas

Actividad Formativa	Horas	Presencialidad
AF1 Clases teóricas / Expositivas	36	100%
AF2 Clases Prácticas	18	100%
AF3 Tutorías	4	50%
AF4 Estudio independiente y trabajo autónomo del alumno	51	0%
AF5 Elaboración de trabajos (en grupo o individuales)	33	0%
AF6: Actividades de Evaluación	6	100%

## 7. SISTEMA DE EVALUACIÓN

ACTIVIDAD DE EVALUACIÓN	VALORACIÓN MÍNIMA RESPECTO A LA CALIFICACIÓN FINAL (%)	VALORACIÓN MÁXIMA RESPECTO A LA CALIFICACIÓN FINAL (%)
SE1 Evaluación de la participación en clase, en prácticas o en proyectos de la asignatura	0%	30%
SE2 Evaluación de trabajos, proyectos, informes, memorias	30%	80%
SE3 Prueba Objetiva	10%	60%

## 8. CRITERIOS DE CALIFICACIÓN

ACTIVIDAD DE EVALUACIÓN	VALORACIÓN RESPECTO A LA CALIFICACIÓN FINAL (%)
Evaluación de la participación en clase, en prácticas o en proyectos de la asignatura. Realización y entrega de dichos ejercicios	10%
Evaluación de trabajos, proyectos, informes, memorias	60%
Prueba Objetiva	30%

Consideraciones generales acerca de la evaluación:

- Calificación numérica final será de 0 a 10, siendo un 5 la mínima nota para aprobar.
- Será necesario aprobar tanto el examen como las prácticas por separado con una nota mínima de 5
- Evaluación global del proceso de aprendizaje y adquisición de competencias y conocimientos.
- Se demandará del alumno una participación activa, necesaria para el desarrollo de las clases.
- Aquellos alumnos que suspendan alguna práctica o que las entreguen pasado el tiempo señalado, podrán repetirlas hasta aprobarlas aunque recibirán una penalización en la calificación. La fecha tope de entrega será el último día de clase.



## 9. BIBLIOGRAFÍA / WEBGRAFÍA

<b>Básica:</b>
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
<b>Recomendada:</b>
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.

## 10. MATERIAL: HARDWARE, SOFTWARE Y HERRAMIENTAS NECESARIAS

### Hardware:

- 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:
  - o Terminal posix: Cygwin
  - o Compilador gcc/g++, ejecutado por terminal
  - o Editor Notepad++
- En general, cualquier entorno de programación que soporte C++ servirá, pero se dará soporte a las herramientas mencionadas anteriormente durante la clase.