



## Guía docente de algoritmia (Curso 2021-2022)

### 1. Identificación de la asignatura

NOMBRE	Algoritmia		CÓDIGO	GIISOF01-2-010
TITULACIÓN	Grado en Ingeniería Informática del Software	CENTRO	Escuela de Ingeniería Informática	
TIPO	Obligatoria	Nº TOTAL DE CRÉDITOS	6	
PERIODO	Segundo semestre	IDIOMA	Español & inglés	
COORDINADOR		TELÉFONO /EMAIL		UBICACIÓN
Vicente García Díaz		<a href="mailto:garciavicente@uniovi.es">garciavicente@uniovi.es</a>		Facultad de Ciencias
PROFESORADO		EMAIL		UBICACIÓN
Antonio Martínez Sánchez		<a href="mailto:martinezsantonio@uniovi.es">martinezsantonio@uniovi.es</a>		Facultad de Geología
Cristian González García		<a href="mailto:gonzalezcristian@uniovi.es">gonzalezcristian@uniovi.es</a>		Facultad de Ciencias
Juan Ramón Pérez Pérez		<a href="mailto:jrpp@uniovi.es">jrpp@uniovi.es</a>		Facultad de Ciencias
Oliverio González Alonso		<a href="mailto:oliver@uniovi.es">oliver@uniovi.es</a>		Facultad de Ciencias
Vicente García Díaz		<a href="mailto:garciavicente@uniovi.es">garciavicente@uniovi.es</a>		Facultad de Ciencias

### 2. Contextualización

Esta asignatura pertenece a la **materia de Programación**, y de forma transversal, al **módulo Común de Informática**. Se imparte en el segundo semestre de segundo curso y está muy relacionada con la asignatura de Estructura de Datos, que se imparte en el primer semestre del mismo curso.

Al igual que el resto de las asignaturas de la materia de programación es una asignatura **eminentemente práctica** ya que casi el 50% de las horas presenciales son de prácticas de laboratorio.

En esta asignatura se profundiza en el concepto de **complejidad computacional** (que han planteado otras asignaturas), fundamental para poder comparar la eficiencia de distintos algoritmos. Y se abordan distintas **técnicas de diseño de algoritmos** que permiten resolver problemas computacionalmente complejos. Supone, junto con Estructura de Datos, un puente entre asignaturas básicas de programación hacia asignaturas que abordan problemas específicos: Sistemas Inteligentes, Diseño de Lenguajes de Programación, Software de Entretenimiento y Videojuegos, Software para Dispositivos Móviles.



### 3. Prerrequisitos

Para afrontar esta asignatura es altamente recomendable disponer de competencias que se adquieren en las asignaturas de Introducción a la Programación, Autómatas y Matemáticas Discretas y Metodología de la Programación de primer curso y de Estructura de Datos de segundo curso. También es importante cursar la asignatura de Tecnologías y Paradigmas de la Programación de forma simultánea en el segundo semestre de segundo curso.

### 4. Competencias y resultados de aprendizaje

#### Competencias específicas

- **Bas.3.** Capacidad para comprender y dominar los conceptos básicos de matemática discreta, lógica, algorítmica y complejidad computacional, y su aplicación para la resolución de problemas propios de la ingeniería.
- **Com.6.** 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.
- **Com.8.** Capacidad para analizar, diseñar, construir y mantener aplicaciones de forma robusta, segura y eficiente, eligiendo el paradigma y los lenguajes de programación más adecuados.
- **Com.14.** Conocimiento y aplicación de los principios fundamentales y técnicas básicas de la programación paralela, concurrente, distribuida y de tiempo real.
- **ISW.1.** Capacidad para desarrollar, mantener y evaluar servicios y sistemas software que satisfagan todos los requisitos del usuario y se comporten de forma fiable y eficiente, sean asequibles de desarrollar y mantener y cumplan normas de calidad, aplicando las teorías, principios, métodos y prácticas de la Ingeniería del Software.

#### Competencias generales

- **CG-1.** Competencia para el diseño de soluciones a problemas.
- **CG-4.** Análisis y síntesis.
- **CG-7.** Destreza en la expresión escrita.
- **CG-11.** Competencia para el trabajo en equipo.

#### Resultados de aprendizaje

- **RA.P-3.** Analizar, diseñar, desarrollar, seleccionar, evaluar y mantener aplicaciones y sistemas informáticos, asegurando su fiabilidad y calidad aplicando las teorías, principios, métodos y prácticas de la Ingeniería del Software, eligiendo el paradigma y los lenguajes de programación más adecuados, considerando las limitaciones derivadas del coste, del tiempo, de la existencia de sistemas ya desarrollados y de las propias organizaciones. [Com.1] [Com.8] [Com.16] [ISW.1] [ISW.2] [ISW.4] [ISW.6] [CG-1] [CG-3] [CG-4] [CG-11] [CG-14] [CG-20] [CG-26] [CG-28]
- **RA.P-4.** Conocer y aplicar procedimientos algorítmicos básicos, tipos y estructuras de datos más adecuados a la resolución de un problema, analizando la idoneidad y complejidad de los mismos. [Bas.3] [Com.6] [Com.7] [CG-3]
- **RA.P-5.** Conocer y aplicar los principios fundamentales y técnicas básicas de la programación paralela, concurrente, distribuida y de tiempo real. [Com.14] [CG-11]
- **RA.P-6.** Capacidad de dar solución a problemas de integración en función de las estrategias, estándares y tecnologías disponibles. [ISW.3] [CG-1] [CG-4] [CG-6] [CG-22]



- **RA.P-7.** Documentar y exponer la solución a un problema a través de textos y diagramas, cumpliendo normas y estándares del diseño y desarrollo del software en español/inglés. [CG-10] [CG-7]

## 5. Contenidos

1. Principios de Algoritmia.
  - a. Conceptos
  - b. Complejidad analítica
  - c. Tiempos de ejecución empíricos
2. Ordenación.
  - a. Algoritmos básicos de ordenación
  - b. Algoritmos rápidos de ordenación
  - c. Análisis y comparación de algoritmos de ordenación
3. Técnicas de diseño de algoritmos
  - a. Divide y Vencerás (Recursividad).
  - b. Algoritmos Voraces.
  - c. Programación dinámica.
  - d. Vuelta atrás.
  - e. Algoritmos de ramificación y poda.
4. Algoritmos paralelos y avanzados.
  - a. Divide y vencerás paralelo
  - b. Paralelización de otras técnicas
  - c. Algoritmos avanzados (Árboles de juegos y otras técnicas)

## 6. Metodología y plan de trabajo

Siguiendo la filosofía de los créditos europeos, en la asignatura se plantearán actividades presenciales y no presenciales en las que se llevará a cabo un seguimiento por parte de los profesores de la asignatura.

Las actividades presenciales seguirán cinco modalidades:

1. Clases expositivas, en donde se plantearán los fundamentos de la materia y se guiará al alumno para que pueda realizar el trabajo autónomo complementario.
2. Prácticas de aula/Seminarios, donde se planteará un aprendizaje activo y colaborativo integrando el trabajo en clase con el trabajo que se realice en el campus virtual.
3. Prácticas de laboratorio, donde se realizarán diferentes proyectos en los que se diseñará e implementarán algoritmos para dar solución a diferentes problemas propuestos. Estos proyectos requerirán del trabajo no presencial de los alumnos.
4. Tutorías grupales, se realizará un seguimiento de los alumnos para detectar lagunas y orientarles para solucionarlas.
5. Sesiones de evaluación, se realizarán exámenes tanto teóricos como prácticos con ordenador, para evaluar los conocimientos de los alumnos.

MODALIDADES		Horas	%	Totales
Presencial	Clases Expositivas	21	14%	60
	Práctica de aula / Seminarios / Talleres	7	5%	
	Prácticas de laboratorio / campo / aula de informática / aula de idiomas	28	19%	
	Prácticas clínicas hospitalarias		0%	



	Tutorías grupales	2	1%	
	Prácticas Externas		0%	
	Sesiones de evaluación	2	1%	
No presencial	Trabajo en Grupo	0	0%	90
	Trabajo Individual	90	60%	
	Total	150		

*De forma excepcional, si las condiciones sanitarias lo requieren, se podrán incluir actividades de docencia no presencial. En cuyo caso, se informará al estudiantado de los cambios efectuados.*

## 7. Evaluación del aprendizaje de los estudiantes

### Convocatoria ordinaria

**Asistencia Mínima.** Se requiere una **asistencia mínima de un 80%** a las clases de prácticas de laboratorio.

La evaluación del aprendizaje se realizará a través de un proceso de evaluación continua basado en:

1. **Nota de Teoría (NT):** Esta nota se compondrá de dos partes:
  - a. **Primer examen (PE).** A mitad del semestre se realizará un examen en el que se evaluará aproximadamente la mitad del contenido teórico de la asignatura.
  - b. **Segundo examen (SE).** Coincidiendo con el examen final de la convocatoria ordinaria, se realizará un examen en el que se evaluará aproximadamente la segunda mitad del contenido teórico de la asignatura.

$$NT = 0,5 * PE + 0,5 * SE$$

2. **Nota de Laboratorio (NL):**
  - a. **Evaluación continua de prácticas (NCP).** Evaluación de las prácticas individuales realizadas en el laboratorio a lo largo del curso. El profesor realizará un seguimiento del trabajo realizado por cada alumno y puede proponer la realización de modificaciones o ampliaciones a las prácticas implementadas durante las sesiones de laboratorio. Es necesario cumplir una **asistencia mínima de un 80% y la entrega mínima del 80% de las prácticas** para obtener nota en esta parte.



**b. Examen final de prácticas (NEP).**

- i. Los alumnos que NO hayan superado la asignatura en la evaluación continua y hayan aprobado la teoría, pueden presentarse a un examen final de prácticas. Cuya evaluación constituirá la nota de laboratorio.

$$NL = NEP$$

- ii. Por otra parte, los alumnos cuya nota de evaluación continua de prácticas (NCP) sea  $\geq 5$ , también pueden presentarse para subir la nota de laboratorio.

$$NL = 0,7 * NCP + 0,3 * NEP$$

Para superar la asignatura es necesario que **ambas partes** (NT y NL) tengan una nota igual o superior a 5, la nota final se calculará de la siguiente forma:

$$NF = 0,5 * NT + 0,5 * NL$$

En caso de no superarse alguna de las condiciones requeridas para aprobar y haberse presentado a pruebas de evaluación ponderadas por al menos el 50% del peso total de la asignatura, la nota final se calculará como:

$$NF = \text{Mínimo} (4, 0,5 * NT + 0,5 * NL)$$

Todas las notas parciales tanto de teoría como de prácticas de laboratorio sin asistencia se considerarán como 0 y se aplicará la misma fórmula.

Si el alumno ha aprobado alguna de las dos partes (NT o NL), se le guardará para la convocatoria extraordinaria.

**Convocatoria extraordinaria y evaluación diferenciada**

Habrà un examen escrito para la obtención de **NT**.

Asimismo, habrá un examen práctico con ordenador para obtener **NL**.

Si ambas notas son iguales o superiores a 5, se procederá se aplicará la siguiente formula:

$$NF = 0,5 * NT + 0,5 * NL$$

En caso contrario, la nota final se calculará como:

$$NF = \text{Mínimo} (4, 0,5 * NT + 0,5 * NL)$$

**Convocatoria extraordinaria adelantada**

No se guarda ninguna nota anterior y las pruebas de evaluación y el cálculo de la nota serán realizadas como en la convocatoria extraordinaria.



*De forma excepcional, si las condiciones sanitarias lo requieren, se podrán incluir métodos de evaluación no presencial. En cuyo caso, se informará al estudiantado de los cambios efectuados.*

## 8. Recursos, bibliografía y documentación complementaria

- Graphs, algorithms, and optimization. *William Kocay, Donald L. Kreher*. Chapman & Hall-CRC, 2017 (ISBN: 978-1-482-25116-6).
- Fundamentos de Algoritmia. *Gilles Brassard, Paul Bratley* (traducción de *Fundamentals of algorithmics* realizada por *Rafael García-Bermejo*). Pearson Educación, 2006 (ISBN: 978-8-489-66000-7).
- Introduction to algorithms. *Thomas H. Cormen [et al.]*. The MIT Press, 2003 (ISBN: 0262032937).
- Estructura de datos y métodos algorítmicos: ejercicios resueltos. *Narciso Martí Oliet, Yolanda Ortega Mallén, José Alberto Verdejo López*. Pearson Educación, 2003 (ISBN: 8420538493).
- Esquemas Algorítmicos: Enfoque Metodológico y Problemas Resueltos. *Julio Gonzalo Arroyo, Miguel Rodríguez Artacho*. Universidad Nacional de Educación a Distancia (UNED), 1998 (ISBN: 843623622X).

La asignatura dispondrá de un **curso en el campus virtual de la Universidad de Oviedo**, en el que se proporcionará materiales y se crearán tareas de entrega para los distintos trabajos y ejercicios que se pidan en la asignatura.