



GUÍA DOCENTE  
**ALGORÍTMICA Y PROGRAMACIÓN**

Coordinación: MARTINEZ RODRIGUEZ, SANTIAGO

Año académico 2023-24

## Información general de la asignatura

<b>Denominación</b>	ALGORÍTMICA Y PROGRAMACIÓN			
<b>Código</b>	102364			
<b>Semestre de impartición</b>	1R Q(SEMESTRE) EVALUACIÓN CONTINUADA			
<b>Carácter</b>	<b>Grado/Máster</b>	<b>Curso</b>	<b>Carácter</b>	<b>Modalidad</b>
	Grado en Técnicas de Interacción Digital y de Computación	1	TRONCAL/BÁSICA	Presencial
<b>Número de créditos de la asignatura (ECTS)</b>	6			
<b>Tipo de actividad, créditos y grupos</b>	Solo examen			
<b>Coordinación</b>	MARTINEZ RODRIGUEZ, SANTIAGO			
<b>Departamento/s</b>	INGENIERÍA INFORMÁTICA Y DISEÑO DIGITAL			
<b>Distribución carga docente entre la clase presencial y el trabajo autónomo del estudiante</b>	6 ECTS = 25x6 = 150 horas de trabajo. No hay clases de esta asignatura.			
<b>Información importante sobre tratamiento de datos</b>	Consulte <a href="#">este enlace</a> para obtener más información.			
<b>Idioma/es de impartición</b>	Catalán.			
<b>Distribución de créditos</b>	No hay clases de esta asignatura.			

Profesor/a (es/as)	Dirección electrónica\profesor/a (es/as)	Créditos impartidos por el profesorado	Horario de tutoría/lugar
MARTINEZ RODRIGUEZ, SANTIAGO	santi.martinez@udl.cat	0	Concertar con el profesor. Opcionalmente, por videoconferencia.

## Información complementaria de la asignatura

Para abordar la asignatura es recomendable mostrar interés en plantear y analizar problemas reales, para los que se deberá buscar y desarrollar una solución tecnológica que los resuelva.

Es también recomendable mostrar capacidad de análisis, de razonamiento lógico, trabajo metódico y organizado, y capacidad crítica.

Los conocimientos y competencias adquiridas en esta asignatura serán de utilidad en asignaturas posteriores con contenidos de programación, estructuras de datos y algorítmica.

## Objetivos académicos de la asignatura

Los objetivos de aprendizaje de la asignatura son analizar y diseñar algoritmos para, posteriormente, implementar en un lenguaje de programación compilable.

Concretamente, el lenguaje imperativo escogido es ANSI C/C++ y los problemas a resolver son principalmente de tratamiento y búsqueda en secuencias.

En concreto, los objetivos de aprendizaje del estudiante son los siguientes:

- Diseñar e implementar estructuras algorítmicas adecuadas para resolver las diferentes tipologías de problemas.
- Diseñar e implementar estructuras de datos adecuadas para representar la información propia de cada problema.
- Diseñar e implementar algoritmos iterativos.
- Identificar la tipología del problema y aplicar la estrategia algorítmica adecuada.
- Diseñar e implementar algoritmos para resolver problemas complejos de forma estructurada.
- Diseñar e implementar soluciones algorítmicas básicas utilizando la técnica de diseño descendente.
- Utilizar un entorno de desarrollo de programas basado en un lenguaje de programación de alto nivel.

## Competencias

### Competencias Básicas

- **B01.** Que los estudiantes hayan demostrado poseer y comprender conocimientos en su 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.

### Competencias Transversales

- **CT3.** Adquirir capacitación en el uso de las nuevas tecnologías y de las tecnologías de la información y la comunicación.
- **CT5.** Adquirir nociones esenciales del pensamiento científico.

### Competencias Generales

- **CG2.** Capacidad para diseñar, desarrollar, evaluar y asegurar la accesibilidad, ergonomía, usabilidad y seguridad de los sistemas informáticos.
- **CG3.** Capacidad para utilizar plataformas hardware y software adecuadas para el desarrollo y la ejecución de aplicaciones digitales interactivas.
- **CG5.** 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.
- **CG7.** Capacidad para resolver problemas con iniciativa, toma de decisiones, autonomía y creatividad.
- **CG8.** Capacidad para la abstracción y el razonamiento crítico, lógico y matemático.

### Competencias Específicas

- **CE2.** 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 computación.

- **CE3.** Conocimientos básicos sobre el uso y programación de los ordenadores, sistemas operativos y bases de datos útiles para el desarrollo de aplicaciones informáticas interactivas.
- **CE4.** Capacidad de conocer, comprender y evaluar la estructura y arquitectura de los computadores, así como los componentes básicos que los conforman.
- **CE16.** Capacidad para diseñar y evaluar interfaces persona computador que garanticen la usabilidad de los sistemas, servicios y aplicaciones informáticas.
- **CE17.** Saber aplicar los conocimientos de diseño suficientes para proponer y defender un concepto de diseño de un entorno interactivo y desarrollarlo hasta que pueda ser llevado a la práctica utilizando las tecnologías creativas adecuadas a cada proyecto.
- **CE24.** Ser capaz de comprender los factores humanos que intervienen en todo proceso de interacción entre personas y tecnología, así como saber aplicarlos de forma adecuada al diseño de productos y servicios interactivos y sus interfaces.

## Contenidos fundamentales de la asignatura

**Introducción:** Procesos, algoritmos y programas.

### Tema 1. Estructuras algorítmicas básicas

- 1.1 Constantes, variables, tipos elementales, y expresiones válidas
- 1.2 La asignación, la composición secuencial, la composición alternativa y la composición iterativa
- 1.3 Entorno de programación

### Tema 2. Diseño de programas iterativos

2.1 Acceso secuencial

- Esquemas algorítmicos de tratamiento de secuencias
- Esquemas algorítmicos de búsqueda en secuencias

2.2 Acceso directo. Las tablas

- Tratamiento secuencial de tablas
- Tratamiento directo de tablas
- Tablas unidimensionales y multidimensionales
- Algoritmos de ordenación clásicos

### Tema 3. Tratamiento de datos no elementales

- 3.1. Acciones y funciones
- 3.2 Mecanismos de paso de parámetros
- 3.3 Diseño descendente de algoritmos

## Ejes metodológicos de la asignatura

No hay clases de esta asignatura.

Sin embargo, el estudiante que lo desee podrá asistir como oyente a la asignatura equivalente (Programación I) del Grado en Ingeniería Informática que se imparte en el mismo campus de Igualada.

## Plan de desarrollo de la asignatura

No hay clases de esta asignatura.

## Sistema de evaluación

La **evaluación continua** de la asignatura se basa en tres bloques:

- **Bloque de Prácticas (25%):** Consiste en dos actividades: Práctica 1 y Práctica 2. No son recuperables. No tienen nota mínima.
- **Bloque de teoría 1 (25%):** Consiste en una actividad: Examen 1r Parcial. Recuperable con el bloque de teoría 2. No tiene nota mínima. Fecha de realización: la fecha para la realización del examen del 1er parcial fijada por la EPS.
- **Bloque de teoría 2 (50%):** Consiste en una actividad: Examen 2o Parcial. Recuperable. No tiene nota mínima. Fecha de

realización: la fecha para la realización del examen del 2o parcial fijada por la EPS.

**Recuperación del Bloque de teoría 2:** Consiste en la recuperación del Examen 2o Parcial. No tiene nota mínima. Fecha de realización: la fecha para la realización del examen de recuperación fijada por la EPS. La realización de la recuperación del examen del 2o parcial no condiciona la calificación máxima alcanzable en la asignatura.

## Actividades de evaluación

Acrónimo	Actividad de Evaluación	Peso	Nota Mínima	En grupo	Obligatoria	Recuperable
EP1	1.º Examen Parcial	25%	No	No	No	Sí
EP2	2.º Examen Parcial	50%	No	No	No	Sí
PR1	Práctica 1	10%	No	Sí ( $\leq 2$ )	No	No
PR2	Práctica 2	15%	No	Sí ( $\leq 2$ )	No	No
Para aprobar la asignatura la nota final deberá ser $\geq 5$ .						
<b>Nota Final</b> = $0,25 \cdot EP1 + 0,5 \cdot EP2 + 0,1 \cdot PR1 + 0,15 \cdot PR2$						

## Observaciones:

- Si la nota obtenida en el examen parcial EP2 es  $\geq 5$ , entonces la nota obtenida podrá actuar como recuperación/mejora del primer examen parcial EP1.
- El estudiante puede optar a mejorar/recuperar el examen parcial EP2. La recuperación será una única prueba escrita. La nota obtenida sustituirá la nota de los dos exámenes parciales del curso.

## Evaluación alternativa (estudiantes que renuncian a la evaluación continua):

El estudiante que cuente con el visto bueno para ser evaluado mediante evaluación alternativa (ved requisitos y procedimiento a la normativa de evaluación) tendrá que hacer las siguientes actividades.

- **Práctica 1** (10%): No es recuperable y carece de nota mínima. Fecha de entrega: la fecha para la realización del 1er parcial fijada por la EPS.
- **Práctica 2** (15%): No es recuperable y carece de nota mínima. Fecha de entrega: la fecha para la realización del 2o parcial fijada por la EPS.
- **Examen 2o parcial** (75%): Recuperable. No tiene nota mínima. Fecha de realización: la fecha para la realización del examen del 2o parcial fijada por la EPS.
- **Recuperación del examen del 2o parcial** (75%): No tiene nota mínima. Fecha de realización: la fecha para la realización del examen de recuperación fijada por la EPS. La realización de la recuperación del examen del 2o parcial no condiciona la calificación máxima alcanzable en la asignatura.

## Bibliografía y recursos de información

### Algorítmica

- Teresa Alsinet, Josep Argelich, Sergi Vila: Programació I. Notes del curs. Eines; Edicions i Publicacions de la Universitat de Lleida.
- Jorge Castro, Felipe Cucker, Xavier Messeguer, Albert Rubio, Lluís Solano, Borja Valles: Curs de Programació. McGraw-Hill, 1992.
- Gilles Brassard, Paul Bratley: Fundamentos de Algoritmia. Prentice Hall, 1997.
- Luis Joyanes: Fundamentos de Programación. Algoritmos, Estructuras de Datos y Objetos. McGraw-Hill, 2003.

### El lenguaje ANSI C y C++

- Harvey M. Deitel, Paul J. Deitel: Cómo Programar en C/C++. Prentice-Hall, segunda edición, 2002.
- Bjarne Stroustrup: Programming: Principles and Practice Using C++. Addison Wesley, 2008.
- Luis Joyanes: Programación en C++. McGraw-Hill, 2006.