



## GUÍA DOCENTE

### COMPETENCIAS

- 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 en el campo de la Ingeniería Informática.
- CB4 Que los estudiantes puedan transmitir información, ideas, problemas y soluciones a un público tanto especializado como no especializado.
- CEB3 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.
- CEB4 Conocimientos básicos sobre el uso y programación de los ordenadores, sistemas operativos, bases de datos y programas informáticos con aplicación en ingeniería.
- CEB5 Conocimiento de la estructura, organización, funcionamiento e interconexión de los sistemas informáticos, los fundamentos de su programación, y su aplicación para la resolución de problemas propios de la ingeniería.

### OBJETIVOS

Proporcionar una importante base teórica y práctica en el aprendizaje de la programación de ordenadores. Esta base permitirá cursar posteriores estudios de programación avanzada. Concretamente, se pretende que el alumno:

- + Aprecie el papel central que juega la abstracción en la tarea de programar.
- + Utilice una metodología basada en el uso de la programación estructurada.
- + Conozca las estructuras de datos, algoritmos y esquemas de uso general.
- + Documente adecuadamente los programas.
- + Realice programas correctos, claros, y eficientes.
- + Verifique el funcionamiento de los programas.
- + Conozca el lenguaje de programación C (lenguaje de alto nivel estructurado, general y extendido), de manera que sea capaz de implementar en dicho lenguaje los algoritmos diseñados previamente y, de este modo, completar adecuadamente el proceso de desarrollo de programas.
- + Sea capaz de entender y seguir el código de los programas realizados por otras personas.
- + Se habitúe a desarrollar programas siguiendo unas normas de estilo tendentes a conseguir programas de calidad. Dentro de estas normas de estilo se engloban aquellas que facilitan la comprensión del código, como pueden ser el uso de comentarios, la indentación de código, ...

### CONTENIDOS

#### 1. Contenidos teóricos

- Bloque Temático 1.- Introducción a la programación.
- Bloque Temático 2.- Esquemas condicionales
- Bloque Temático 3.- Esquemas iterativos
- Bloque Temático 4.- Funciones.
- Bloque Temático 5.- Vectores
- Bloque Temático 6.- Matrices
- Bloque Temático 7.- Cadenas de caracteres
- Bloque Temático 8.- Estructuras de datos
- Bloque Temático 9.- Misceláneo: argumentos a main y defines

#### 2. Contenidos prácticos

- Unidad Práctica 1. Introducción al Lenguaje de programación y el compilador.
- Unidad Práctica 2. Esquemas iterativos I.



## GUÍA DOCENTE

Unidad Práctica 3. Esquemas iterativos II.  
Unidad Práctica 4. Programación modular.  
Unidad Práctica 5. Programación de vectores  
Unidad Práctica 6. Programación de matrices  
Unidad Práctica 7. Programación de cadenas  
Unidad Práctica 8. Programación de estructuras de datos  
Unidad Práctica 9. Misceláneo: argumentos a main y defines

## OBJETIVOS DE DESARROLLO SOSTENIBLE RELACIONADOS CON LOS CONTENIDOS

Sin relación

## METODOLOGÍA

### Aclaraciones generales sobre la metodología (opcional)

#### Lecciones magistrales

En estas sesiones se presentarán los conceptos de cada tema a la par que se desarrollarán ejemplos que ayuden al alumno a comprender y clarificar los conceptos. Los ejemplos consistirán en la resolución de algoritmos mediante la aplicación de esquemas de creciente dificultad a lo largo del curso.

En algunos casos, se hará uso de la metodología inversa, en la que el alumno deberá estudiar el tema en casa, utilizando videos y materiales proporcionados por el profesor, y la clase presencial se empleará para resolver dudas y problemas.

#### Laboratorio

Las clases de prácticas se realizan en un aula de ordenadores con el software necesario para implementar los programas. El objetivo de estas clases consiste en la implementación en el ordenador de un conjunto de ejercicios de programación cuyo enunciado estará previamente a disposición del alumno en la página web de la asignatura.

#### Tutorías

Durante el desarrollo de la asignatura, cada alumno dispondrá, además del foro que se colocará en Moodle para la exposición y resolución de dudas de forma colectiva, de las tutorías individualizadas que desee dentro del horario establecido por el profesor. Igualmente, se han programado 2 horas de tutorías colectivas que se celebrarán a lo largo del curso. En estas sesiones se resolverán dudas de teoría de cara al examen final

#### Resolución de problemas

Se usarán, como apoyo a las clases teóricas y prácticas, y se realizarán de forma individual por cada alumno en el aula. Estas sesiones se centrarán en el análisis de problemas y su planteamiento como algoritmos o programas.

### Adaptaciones metodológicas para alumnado a tiempo parcial y estudiantes con discapacidad y necesidades educativas especiales

Para los estudiantes a tiempo parcial se tendrá en cuenta su condición y disponibilidad en la asignatura, tanto en el desarrollo de la misma como en su evaluación. La adaptación del estudiante a tiempo parcial a la asignatura se llevará a cabo de mutuo acuerdo entre el Profesor responsable de la misma y los alumnos implicados al inicio del cuatrimestre.

## GUÍA DOCENTE

### Actividades presenciales

Actividad	Grupo completo	Grupo mediano	Total
Actividades de evaluación	2	-	2
Lección magistral	32	-	32
Resolución de problemas	-	24	24
Tutorías	2	-	2
<b>Total horas:</b>	<b>36</b>	<b>24</b>	<b>60</b>

### Actividades no presenciales

Actividad	Total
Análisis	15
Búsqueda de información	15
Ejercicios	15
Estudio	15
Problemas	30
<b>Total horas:</b>	<b>90</b>

## MATERIAL DE TRABAJO PARA EL ALUMNO

Casos y supuestos prácticos  
Cuaderno de Prácticas  
Ejercicios y problemas

### Aclaraciones

El alumno contará con un amplio material didáctico y de referencia para realizar su trabajo. Además, las clases serán grabadas y puestas a disposición de los alumnos en la web de la asignatura.

## EVALUACIÓN

Competencias	Diarios	Exámenes	Prácticas de laboratorio
CB2	X	X	X
CB4		X	
CEB3	X	X	X

## GUÍA DOCENTE

Competencias	Díarios	Exámenes	Prácticas de laboratorio
CEB4			X
CEB5	X	X	
<b>Total (100%)</b>	<b>10%</b>	<b>80%</b>	<b>10%</b>
<b>Nota mínima (*)</b>	<b>5</b>	<b>5</b>	<b>5</b>

(\*)Nota mínima (sobre 10) necesaria para que el método de evaluación sea considerado en la calificación final de la asignatura. En todo caso, la calificación final para aprobar la asignatura debe ser igual o superior a 5,0.

### Valora la asistencia en la calificación final:

No

### Aclaraciones generales sobre los instrumentos de evaluación:

La evaluación consistirá, por una parte en la entrega de prácticas y trabajos que se realizará a lo largo del curso, y que contarán un 20% sobre la nota final. Por otro lado, se realizará un examen práctico al final del curso que contará el 80% restante. Para los alumnos que se presenten en una convocatoria diferente a la de Enero o Febrero, el examen será el 100% de la nota.

### Aclaraciones sobre la evaluación para el alumnado a tiempo parcial y necesidades educativas especiales:

Se empleará el mismo procedimiento que para los alumnos a tiempo completo.

### Aclaraciones sobre la evaluación de la convocatoria extraordinaria y convocatoria extraordinaria de finalización de estudios:

Se empleará el mismo procedimiento que para los alumnos a tiempo completo.

### Criterios de calificación para la obtención de Matrícula de Honor:

Tener al menos un 9 en todas las partes evaluables

## BIBLIOGRAFIA

### 1. Bibliografía básica

#### Introducción a la Informática

- \* A. Prieto, A. Lloris, J.C. Torres. Introducción a la Informática. McGraw-Hill, 4ª edición, 2006.
- \* R. Medina et al. Fundamentos de Informática. Algaída, 1998. Metodología y Algorítmica
- \* J.J. García, J.L. Fernández, F.J. Montoya. Una introducción a la programación. Un enfoque algorítmico. Thomson, 2005.
- \* F.J. Cortijo, J.C. Cubero, O. Pons. Metodología de la Programación. Programas y estructuras de datos en Pascal. Proyecto Sur, 1993.
- \* J.L. Castro et al. Metodología de la Programación. Problemas resueltos. Proyecto Sur, 1993.

### 2. Bibliografía complementaria

#### Sistema Operativo Linux

- \* R. Baig, F. Auli. Sistema Operativo GNU/Linux Básico. UOC, 2003. ISBN:84-9788-028-3



## GUÍA DOCENTE

\* S.M. Sarwar, R. Koretsky, S.A. Sarwar. El libro de Linux. Addison Wesley, 2003.

## CRITERIOS DE COORDINACIÓN

Fecha de entrega de trabajos  
 Selección de competencias comunes  
 Trabajos válidos para varias asignaturas

## CRONOGRAMA

Periodo	Actividades de evaluación	Lección magistral	Resolución de problemas	Tutorías
1ª Quincena	0,0	4,0	0,0	0,0
2ª Quincena	0,0	5,0	4,0	0,0
3ª Quincena	0,0	5,0	4,0	0,0
4ª Quincena	0,0	5,0	4,0	0,0
5ª Quincena	0,0	5,0	4,0	0,0
6ª Quincena	0,0	4,0	4,0	0,0
7ª Quincena	2,0	4,0	4,0	2,0
<b>Total horas:</b>	<b>2,0</b>	<b>32,0</b>	<b>24,0</b>	<b>2,0</b>

Las estrategias metodológicas y el sistema de evaluación contempladas en esta Guía Docente serán adaptadas de acuerdo a las necesidades presentadas por estudiantes con discapacidad y necesidades educativas especiales en los casos que se requieran.