



# FUNDAMENTOS DE PROGRAMACIÓN I

## CURSO 2012/2013

<b>Asignatura:</b>	Fundamentos de Programación I
<b>Materia:</b>	Fundamentos de Programación
<b>Titulación:</b>	Grado en Ingeniería Informática
<b>Curso:</b>	Primero
<b>Créditos</b>	6
<b>Semestre:</b>	Primero
<b>Carácter:</b>	Formación básica
<b>Duración:</b>	Semestral
<b>Idioma en que se imparte:</b>	Español
<b>Profesores:</b>	Matilde Fernández Azuela ( <a href="mailto:matilde.fernandez@upsam.net">matilde.fernandez@upsam.net</a> ) Luis Rodríguez Baena ( <a href="mailto:luis.rodriguez@upsam.net">luis.rodriguez@upsam.net</a> )

## Objetivos formativos

El objetivo general de la materia es que los estudiantes sean capaces de resolver de manera eficaz distintos problemas de carácter general, con independencia del lenguaje de programación utilizado. Este objetivo requiere que los alumnos que cursen la asignatura adquieran el conocimiento teórico y práctico sobre las técnicas básicas de programación estructurada consiguiendo la *obtención de ideas intuitivas y claras de los conceptos y técnicas estudiados*, y permitirá entender fácilmente nuevos modelos facilitando la aplicación práctica de los algoritmos.

## Competencias

### Competencias transversales:

- CT1 Capacidad de análisis y síntesis
- CT2 Capacidad de organización y planificación
- CT6 Resolución de problemas
- CT7 Toma de decisiones

### Competencias básicas:

- CB5 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.



## Competencias específicas

- CI6** 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.
- CI8** Capacidad para analizar, diseñar, construir y mantener aplicaciones de forma robusta, segura y eficiente, eligiendo el paradigma y los lenguajes de programación más adecuados.
- CC1** Capacidad para tener un conocimiento profundo de los principios fundamentales y modelos de la computación y saberlos aplicar para interpretar, seleccionar, valorar, modelar, y crear nuevos conceptos, teorías, usos y desarrollos tecnológicos relacionados con la informática.
- CC3** Capacidad para evaluar la complejidad computacional de un problema, conocer estrategias algorítmicas que puedan conducir a su resolución y recomendar, desarrollar e implementar aquella que garantice el mejor rendimiento de acuerdo con los requisitos establecidos.

## Requisitos previos

---

Esta asignatura no precisa de conocimientos previos

## Contenido de la asignatura

---

### Programa de la asignatura

#### TEMA 1. RESOLUCIÓN DE PROBLEMAS POR ORDENADOR

- 1.1. Introducción al procesamiento de la información
- 1.2. Fases en la resolución de problemas
- 1.3. Análisis del problema.
- 1.4. Diseño del algoritmo.
- 1.5. Implementación y pruebas
- 1.6. Descripción de los algoritmos: herramientas de programación

#### TEMA 2. ELEMENTOS DE UN PROGRAMA

- 2.1. Datos y tipos de datos
- 2.2. Constantes y variables
- 2.3. Expresiones
- 2.4. Instrucciones
- 2.5. Estructura de un algoritmo

#### TEMA 3. ESTRUCTURAS DE CONTROL

- 3.1. Programación modular y programación estructurada
- 3.2. Flujo de control de un programa
- 3.3. Estructura secuencial
- 3.4. Estructuras selectivas
- 3.5. Estructuras repetitivas



**TEMA 4. PROGRAMACIÓN MODULAR: PROCEDIMIENTOS Y FUNCIONES**

- 4.1. Introducción a la programación modular
- 4.2. Funciones
- 4.3. Procedimientos
- 4.4. Visibilidad de las variables
- 4.5. Intercambio de información: paso de parámetros

**TEMA 5. ESTRUCTURAS DE DATOS: ARRAYS Y REGISTROS**

- 5.1. Introducción a las estructuras de datos
- 5.2. Arrays
- 5.3. Registros

**TEMA 6. CADENAS**

- 6.1. Datos alfanuméricos
- 6.2. Operaciones con cadenas y caracteres
- 6.3. Funciones de cadena

**TEMA 7. Introducción al lenguaje C**

## Materiales y recursos de enseñanza-aprendizaje

---

### Bibliografía

#### Bibliografía básica

JOYANES, L. (2008). *Fundamentos de la programación. Algoritmos y Estructura de Datos*, 4ª Edición. Madrid: McGraw-Hill.

JOYANES, L.; RODRIGUEZ, L; FERNANDEZ, M. (2003). *Fundamentos de programación Libro de problemas*. 2ª Edición. Madrid: McGraw-Hill.

#### *Programación C*

KERNIGHAN, B. W.; RITCHIE, D. M. (1991). *El lenguaje de Programación C*, 2ª Edición. México: Prentice-Hall Hispanoamericana

JOYANES, L.; ZAHOHERO, I. (2005). *Programación en C : metodología, algoritmos y estructura de datos*, 2ª Edición. Madrid: McGraw-Hill

#### Bibliografía complementaria.

AHO, Alfred V.; HOPCROFT, John E.; ULLMAN, Jeffrey D. (1998). *Estructuras de datos y algoritmos*. México: Addison Wesley.

BRASSARD, G.; BRATLEY, P. (1997). *Fundamentos de Algoritmia*. Madrid: Prentice-Hall.

COLLADO MACHUCA, M.; MORALES FERNÁNDEZ, R.; MORENO NAVARRO, J. J. (1987). *Estructuras de datos. Realización en Pascal*. Madrid: Ediciones Díaz de Santos.



GARCÍA MOLINA, J. J.; MONTOYA DATO, F. J.; FERNÁNDEZ ALEMÁN, J. L.; MAJADO ROSALES, M. J. (2005). *Una introducción a la programación. Un enfoque algorítmico*. Madrid: Thomson-Paraninfo.

JOYANES, L. (1990). *Problemas de Metodología de la Programación*. Madrid: McGraw-Hill.

JOYANES, L.; ZAHOHERO, I. (2005). *Programación en C : metodología, algoritmos y estructura de datos*, 2ª Edición. Madrid: McGraw-Hill

NAVIA, J. (2012). *Introduction to C using lcc-win* [en línea]. Jacob Navia, Logiciels/Informatique: Villetaneuse (France). Disponible en <http://www.cs.virginia.edu/~lcc-win32/C-Tutorial.pdf> [Consultado el 13/05/2012]

WIRTH. N. (1986). *Algoritmos + Estructuras de datos = programas*. Madrid: Ediciones del Castillo S. A.

## Direcciones web

*lcc-win32: A Compiler system for windows*

Sitio web desde donde se puede descargar el compilador lcc-win y la documentación asociada.

<http://www.cs.virginia.edu/~lcc-win32/>

*lcc Win32 Compiling Tutorial*

Tutorial para la instalación, compilación y manejo de lcc.

<http://www.quake2.com/dll/tutorials/lcc/index.html>

*Librerías ANSI C*

Recopilación de las funciones que aparecen en las librerías estándares de ANSI C. Por cada función explica su formato, su utilidad y un ejemplo.

<http://c.conclase.net/librerias/index.php>

*C standard questions*

Recopilatorio de direcciones web sobre C

[http://home.datacomm.ch/t\\_wolf/tw/c/](http://home.datacomm.ch/t_wolf/tw/c/)

*comp.lang.c Frequently Asked Questions*

Recopilatorio de mensajes del grupo comp.lang.c de usenet

<http://c-faq.com/>

## Otros recursos documentales

- Apuntes y documentación aportados por el profesor
- Selección de ejercicios propuestos

## Materiales y recursos de apoyo



- Plataforma virtual UPSAM (MOODLE) en campus.upsam.org
- Aula de teoría con equipada con ordenador conectado a un dispositivo de proyección, pizarra, etc.

## Metodología docente

---

### Actividad presencial

Aproximadamente, un 60% de la actividad docente será presencial en el aula. Entre las actividades presenciales se incluyen:

- Exposición por parte del profesor (40%)
- Resolución de ejercicios y problemas propuestos (20%)
- Prácticas en laboratorio informático (30%)
- Tutorías y seminarios en grupo (5%)
- Actividades de evaluación (5%)

La asistencia a las actividades presenciales será **obligatoria**, siendo necesario asistir al menos al **80%** de las mismas.

### Actividad no presencial

El 40% restante será trabajo autónomo de los estudiantes, entre el que se incluye:

- Estudio individual y preparación de evaluaciones
- Resolución de ejercicios y problemas propuestos
- Elaboración de trabajos y prácticas
- Búsqueda de información y documentación

## Tutorías

---

A lo largo del periodo docente y de exámenes los profesores de la asignatura darán apoyo tutorial presencial a los estudiantes que lo necesiten. La disponibilidad horaria de los profesores se publicará al comienzo del periodo docente.

## Método general de evaluación

---

### Evaluación continua a lo largo del semestre



A lo largo del periodo lectivo se realizará un seguimiento continuo de los resultados de aprendizaje. Las actividades de evaluación que se realizarán serán:

- Pruebas de carácter teórico-práctico realizadas en el aula de teoría o en el laboratorio con problema similares a los realizados en el aula (45% de la nota final).
- Un conjunto de ejercicios, trabajos de curso o prácticas realizados fuera del aula de forma individual o en grupo (25% de la nota final)
- Prueba objetiva al final del periodo lectivo con cuestiones teóricas y prácticas que los estudiantes deberán realizar de forma individual (30% de la nota final)

## **Convocatoria ordinaria de febrero**

Los estudiantes que no hayan superado alguna de las partes de las que consta la evaluación continua podrán realizar una prueba ordinaria de carácter teórico-práctico al final del periodo lectivo con materia correspondiente.

## **Convocatoria extraordinaria de julio**

Los estudiantes que no hayan superado alguna de las partes de las que consta la evaluación continua podrán realizar una prueba ordinaria de carácter teórico-práctico en la convocatoria extraordinaria de julio.

## **Calificación final**

La evaluación concluye con un reconocimiento sobre el nivel de aprendizaje conseguido por el estudiante y se expresa en calificaciones numéricas, de acuerdo con lo establecido en la legislación vigente.