

# **Guía Docente**

***ASIGNATURA PROGRAMACIÓN II***

***CURSO SEMESTRE 2***

***GRADO (S) INGENIERÍA DE SISTEMAS DE  
INFORMACIÓN***

***MODALIDAD: PRESENCIAL***

***CURSO 2014-2015***

***ESCUELA POLITÉCNICA SUPERIOR***

## 1. IDENTIFICACIÓN DE LA ASIGNATURA

### 1.- ASIGNATURA:

Nombre: Programación II	
Código: c109	
Curso(s) en el que se imparte: Primero	Semestre(s) en el que se imparte: Segundo
Carácter: Obligatoria	ECTS: 6
Idioma: Español	Modalidad: Presencial
Grado(s) en que se imparte la asignatura: Ingeniería de Sistemas de Información	
Facultad en la que se imparte la titulación: Escuela Politécnica Superior	

### 2.- ORGANIZACIÓN DE LA ASIGNATURA:

Departamento: Tecnologías de la Información
Área de conocimiento: Lenguajes y Sistemas Informáticos

## 2. PROFESORADO DE LA ASIGNATURA

### 1.- IDENTIFICACIÓN DEL PROFESORADO:

PROFESOR(ES)	DATOS DE CONTACTO
Mariano Fernández López	Tfno.: 913726435    Ext.: 4960    e-mail: mfernandez.eps@ceu.es    despacho: D.2.3.1

### 2.- ACCIÓN TUTORIAL:

Para todas las consultas relativas a la materia, los alumnos pueden contactar con el/los profesores a través del e-mail, del teléfono y en el despacho a las horas de tutoría que se harán públicas, en el portal del alumno.

### 3. OBJETIVOS DE LA ASIGNATURA

#### RESULTADOS DE APRENDIZAJE ESPECÍFICOS

Los resultados de aprendizaje específicos de la memoria modifica que se toman como base para esta asignatura son: RAEB 11, 12,14 y 15.

Al terminar con éxito esta asignatura, los estudiantes serán capaces de:

- Razonar apoyándose en los conceptos básicos de algorítmica y complejidad computacional (RAEB-11).
- Realizar análisis de complejidad de algoritmos (RAEB-12).
- Programar en un lenguaje de programación utilizando estructuras de datos comunes de programación (RAEB-14).
- Implementar algoritmos básicos y estructuras de datos comunes (RAEB-15).

### 4. METODOLOGÍA DOCENTE DE LA ASIGNATURA

#### ACTIVIDADES FORMATIVAS

- **Clase magistral:** exposición de conceptos teóricos por parte del profesor, exigiendo al alumno preparación previa y/o estudio posterior. Se fomentará la participación del estudiante para la aclaración de conceptos y resolución de dudas.
- **Seminario:** exposición por parte del profesor de la aplicación de los conceptos teóricos a problemas prácticos. La participación del estudiante tendrá un papel relevante para la discusión de soluciones, análisis de distintas alternativas, escenarios what-if, etc.
- **Taller:** Trabajo realizado por los alumnos individualmente y en grupo que puede ser:
  1. **Taller de problemas:** donde se realizarán problemas a partir de enunciados.
  2. **Taller de programación:** donde se diseñarán e implementarán diferentes programas a partir de unos enunciados. Utilizarán un entorno integrado de desarrollo.

Los alumnos podrán ser requeridos para exponer las soluciones alcanzadas, sirviendo como base para la discusión en común sobre la idoneidad de dichas soluciones.

- **Proyecto:** Los alumnos trabajarán en un proyecto que abarcará la duración de toda la asignatura y que se desarrollará por equipos y, parcialmente, fuera del horario de clases. Dicho proyecto consistirá en el diseño e implementación de un programa a partir de un conjunto de requisitos. La diferencia esencial con respecto al taller es que el proyecto es global, mientras que los talleres sirven para trabajar conceptos específicos tratados en la asignatura.

### 5. EVALUACIÓN DEL APRENDIZAJE

## 1.- ASISTENCIA A CLASE:

- Para poder acogerse al sistema de evaluación continua es precisa la asistencia al 75% de las clases de teoría (se realizarán controles de asistencia). Ya que el alumno puede faltar el 25% del total de las clases, no se admitirán justificaciones de ausencia.

## 2.- SISTEMAS Y CRITERIOS DE EVALUACIÓN:

### CONVOCATORIA ORDINARIA. EVALUACIÓN CONTINUA:

Prueba a realizar	Sistema de evaluación	Ponderación sobre la nota final de la asignatura	Umbral	Material a entregar
Examen intermedio	[20%-60%] de preguntas de razonamiento	20%	0%	Examen
	[40%-80%] de ejercicios, problemas, supuestos, etc.			
Examen de capacitación	[20%-60%] de preguntas de razonamiento	45%	50%	Examen
	[40%-80%] de ejercicios, problemas, supuestos, etc.			
Evaluación del Proyecto	Evaluación "in situ".	15%	50%	El proyecto completo desarrollado en el entorno de programación.
	Memoria del proyecto (cuadernos de prácticas)	5%	10%	Memoria en PDF.
	Informes quincenales	5%	0%	Cada informe quincenal en PDF.
	Defensa de la práctica (presentaciones y exposiciones orales).	5%	0%	Presentación en PDF.
Evaluación del trabajo individual	Trabajos individuales.	5%	0%	Memoria en PDF.

El instrumento principal para determinar la consecución de los resultados de aprendizaje es el **examen de capacitación**, que cubre toda la asignatura. En caso de no alcanzar el nivel mínimo establecido para cada resultado de aprendizaje, la calificación de la asignatura será inferior a 5 (suspense). En caso de que sea necesario, las pruebas pueden dividirse en sesiones o incluso en días diferentes.

### Sobre los informes quincenales

Como mínimo, el grupo de proyecto deberá entregar un informe de seguimiento de forma quincenal, desde la fecha de inicio del proyecto hasta su entrega. Dicho informe deberá contener:

- Tareas realizadas durante la quincena anterior, indicando quién se ha encargado de cada una de ellas así como tiempo aproximado dedicado por cada miembro del equipo.
- Tareas previstas pero no realizadas para dicha quincena, indicando los motivos.
- Tareas a realizar durante la quincena en curso, indicando quién se va a encargar de cada una de ellas.

### Requisitos para aprobar la asignatura:

1. Media ponderada de todas las pruebas mayor o igual que el 50%.
2. Satisfacer todos los umbrales.
3. Satisfacer los requisitos de asistencia a clase.

### Penalización por retrasos

Se admitirán entregas de proyectos y trabajos fuera de plazo hasta el día del examen de capacitación. Para el retraso en una entrega habrá una penalización del 10% de la nota máxima posible de la actividad correspondiente.

#### CONVOCATORIA EXTRAORDINARIA:

El alumno que no supere la asignatura en la convocatoria ordinaria tendrá la opción de presentarse a la convocatoria extraordinaria. Esta convocatoria constará de una prueba presencial única que determinará la calificación final de la asignatura, sin tener en cuenta su rendimiento académico en la convocatoria ordinaria. Al igual que en ésta, para superar la asignatura en convocatoria extraordinaria será necesario alcanzar el nivel mínimo establecido para cada resultado de aprendizaje.

**Nota:** Considerando que una correcta presentación, redacción y ortografía son mínimos exigibles en cualquier actividad o prueba de nivel universitario, las deficiencias en estos aspectos podrán ser penalizadas con hasta 2 puntos en la calificación de cada prueba.

### 3.- VALORACIÓN FINAL DEL ALUMNO:

PRUEBA	PORCENTAJE
Examen intermedio	20%
Examen de capacitación	45%
Evaluación del proyecto	30%
Evaluación del trabajo individual	5%
<b>CALIFICACIÓN FINAL DE LA ASIGNATURA</b>	<b>100%</b>

## 6. PROGRAMA DE LA ASIGNATURA

### 1.- PROGRAMA DE LA ASIGNATURA:

#### PROGRAMA TEÓRICO:

##### INTRODUCCIÓN A LA ALGORITMIA

- Algoritmos.
- Algoritmos recursivos.

##### COSTES DE TIEMPO

- Coste constante.
- Coste logarítmico.
- Coste lineal y coste  $N \log N$ .
- Coste cuadrático y coste cúbico.
- Coste exponencial.

##### ESTRUCTURAS DE DATOS

- Pilas.
- Colas.
- Listas.
- Árboles.
- Grafos.
- Tablas hash.

##### TÉCNICAS ALGORÍTMICAS

- Fuerza bruta.
- Divide y vencerás.
- Técnica voraz.
- Programación dinámica.
- Vuelta atrás.
- Ramificación y poda.
- Algoritmos de ordenación y búsqueda.

##### TESTEO DE PROGRAMAS

- Pruebas unitarias, de integración y de sistema.
- Pruebas de caja negra y de caja blanca.
- Automatización del testeo de programas.

### PROGRAMA DE TRABAJOS Y PRÁCTICAS:

PRÁCTICA. Diseño, implementación y testeo de un programa de ordenador que haga uso de estructuras de datos tratadas en la asignatura.

TRABAJO INDIVIDUAL. Elaboración de un artículo, noticia, tutorial o similar sobre un tema relacionado con la asignatura.

## 7. BIBLIOGRAFÍA DE LA ASIGNATURA

### 1.- BIBLIOGRAFÍA BÁSICA:

Weiss, M (2012) *Data structures and algorithms in Java*. Pearson

### 2.- BIBLIOGRAFÍA COMPLEMENTARIA:

Aho AV, Hopcroft, JE; Ullman, JE *Estructura de datos y algoritmos*. Addison-Wesley.

García-Magariño García I *Estructuras de datos*. Ediciones CEF

Cormen T, Leiserson C, Rivest R, Stein C. *Introduction to Algorithms*. MIT Press.

Donald K. *The Art of Computer Programming*. Addison-Wesley

### 4.- RECURSOS WEB DE UTILIDAD:

<http://www.cs.cmu.edu/~adamchik/15-121/lectures/Algorithmic%20Complexity/complexity.html>

MIT Open Course [https://www.youtube.com/watch?v=moPtWq\\_cVH8](https://www.youtube.com/watch?v=moPtWq_cVH8)

[http://www.programacion.com/articulo/estructuras\\_de\\_datos\\_y\\_algoritmos\\_en\\_java\\_309/](http://www.programacion.com/articulo/estructuras_de_datos_y_algoritmos_en_java_309/)

<http://www.javahispano.org/>

## 8. ACTITUD DENTRO DEL AULA

### 1.- NORMAS:

Las faltas en la Integridad Académica (ausencia de citación de fuentes, plagios de trabajos o uso indebido/prohibido de información durante los exámenes), así como firmar en la hoja de asistencia por un compañero que no está en clase, implicarán la pérdida de la evaluación continua, sin perjuicio de las acciones sancionadoras que estén establecidas.