



**GUÍA DOCENTE**  
Curso 2011-2012

<b>Titulación:</b>	Grado en Ingeniería Informática			<b>Código :</b>	801G
<b>Centro:</b>	FCEAI				
<b>Dirección:</b>	Edificio CCT C / Madre de Dios, 51			<b>Código postal:</b>	26006
<b>Teléfono:</b>	+34 941 29 9607	<b>Fax:</b>	+34 941 299 611	<b>Correo electrónico:</b>	decanato.cai@unirioja.es
<b>Director del Grado:</b>	Ángel Luis Rubio García				
<b>Teléfono:</b>	+34 941 299 449	<b>Correo electrónico:</b>	direstudios.informatica@unirioja.es		
<b>Despacho:</b>	231	<b>Edificio:</b>	Vives		

Fdo.: Ángel Luis Rubio García

En Logroño a 1 de julio de 2011

## GUÍA DOCENTE

Curso 2011-2012

<b>Titulación:</b>	Grado en Ingeniería Informática			801G	
<b>Asignatura:</b>	Especificación y desarrollo de sistemas de software			801207013	
<b>Materia:</b>	Programación				
<b>Módulo:</b>	M3 Programación				
<b>Carácter:</b>	Obligatorio	<b>Curso:</b>	2º	<b>Semestre:</b>	2º
<b>Créditos ECTS:</b>	6	<b>Horas presenciales:</b>	60	<b>Horas de trabajo autónomo estimadas:</b>	90
<b>Idiomas en los que se imparte:</b>	castellano				
<b>Idiomas del material de lectura o audiovisual:</b>	castellano, inglés				

### Departamentos responsables de la docencia:

Matemáticas y Computación			R111
<b>Dirección:</b>	Luis de Ulloa, s/n	<b>Código postal:</b>	26004
<b>Teléfono:</b>	+34 941 299 452	<b>Fax:</b>	+34 941 299 460
<b>Correo electrónico:</b>	dpto.dmc@unirioja.es		

### Profesores

<b>Profesor responsable de la asignatura:</b>	Ana Romero Ibáñez		
<b>Teléfono:</b>	+34 941 299438	<b>Correo electrónico:</b>	
<b>Despacho:</b>	235	<b>Edificio:</b>	Vives
<b>Horario de tutorías:</b>			
<b>Nombre profesor:</b>	Laureano Lambán Pardo		
<b>Teléfono:</b>	+34 941 299446	<b>Correo electrónico:</b>	lalamban@unirioja.es
<b>Despacho:</b>	223	<b>Edificio:</b>	Vives
<b>Horario de tutorías:</b>			

**Descripción de contenidos:**

- Abstracción de datos. Especificación /implementación de tipos de datos. Relación con la orientación a objetos. Aplicaciones al diseño orientado a objetos.
- Especificación de algoritmos.
- Conceptos sobre lenguajes: sintaxis y semántica.
- Análisis de algoritmos: eficiencia y corrección.

**Requisitos previos:**

Se aconseja tener conocimientos y competencias relacionadas con la programación, especialmente con los fundamentos básicos del paradigma orientado a objetos.

**Relación de asignaturas que proporcionan los conocimientos y competencias requeridos:**

Metodología de la Programación  
Tecnología de la Programación  
Programación orientada a objetos

**Contexto**

La asignatura pertenece al módulo "Programación" y aparece en el segundo semestre del segundo curso. Dentro de dicho bloque, es la continuación natural de la asignatura "Programación Orientada a Objetos". Se presupone a los alumnos cierta familiaridad con nociones básicas sobre programación orientada a objetos (clase, objeto, herencia, etc.). Por su parte, pretende revisar algunos conceptos básicos de programación desde un punto de vista más abstracto, haciendo de puente para la asignatura "Procesadores de lenguajes".

Dentro del módulo "Programación", la asignatura pretende dotar a los alumnos de las competencias necesarias para ser capaces de diseñar e implementar aplicaciones desde el paradigma orientado a objetos, viendo los elementos esenciales de la orientación a objetos como herramientas necesarias para dicho diseño. El otro punto esencial de la asignatura es aportar una perspectiva formal (mayor nivel de abstracción) sobre diferentes aspectos relacionados con la programación (sintaxis, semántica, corrección y eficiencia), buscando una mejora en los hábitos del alumno a la hora de programar, que mejore la calidad y fiabilidad de su trabajo.

Las prácticas de laboratorio de la asignatura consisten en el diseño y desarrollo (en un lenguaje que permita orientación a objetos) de los sistemas de clases necesarios para la construcción de pequeñas aplicaciones. Se pretende que, a lo largo de la asignatura, aparezcan involucrados en las prácticas los elementos centrales de la programación orientada a objetos, vistos aquí como elementos necesarios para el diseño.

**Competencias:****Competencias generales**

- **CG2:** Estar capacitado para, utilizando el nivel adecuado de abstracción, establecer y evaluar modelos que representen situaciones reales.
- **CG4:** Estar capacitado para transmitir información, ideas, planteamiento de problemas y soluciones, tanto a otros profesionales tecnológicos y científicos, como a personas ajenas a esas disciplinas.
- **CG7:** Haber desarrollado aquellas habilidades de aprendizaje necesarias para continuar su formación.

**Competencias específicas**

- **CE1:** Capacidad para concebir, redactar, organizar, planificar, desarrollar y firmar proyectos en el ámbito de la ingeniería en informática que tengan por objeto, la concepción, el desarrollo o la explotación de sistemas, servicios y aplicaciones informáticas.

- CE3: Capacidad para diseñar, desarrollar, evaluar y asegurar la accesibilidad, ergonomía, usabilidad y seguridad de los sistemas, servicios y aplicaciones informáticas, así como de la información que gestionan.
- CE10: Conocimientos para la realización de mediciones, cálculos, valoraciones, tasaciones, peritaciones, estudios, informes, planificación de tareas y otros trabajos análogos de informática.

#### Resultados del aprendizaje:

- Aplicar los elementos característicos de la programación orientada a objetos en el diseño y la programación de aplicaciones.
- Utilizar los conceptos relacionados con la abstracción de datos para facilitar el desarrollo de aplicaciones desde un enfoque orientado a objetos.
- Comprender y aprender a establecer descripciones formales de algoritmos.
- Conocer algunos conceptos básicos sobre sintaxis y semántica de lenguajes de programación.
- Aprender a realizar pequeñas verificaciones formales de corrección de algoritmos, conocimiento enfocado a adquirir mejores hábitos de programación.
- Conocer y aplicar conceptos y herramientas relacionados con el análisis de la eficiencia.
- Usar con solvencia algún lenguaje que permita un enfoque orientado a objetos.

#### Temario

1. Introducción: "Abstracción y formalismo en Programación".
2. Especificación/Implementación de tipos de datos.
3. Tipos abstractos y orientación a objetos.
4. Especificación de algoritmos.
5. Sintaxis y semántica de un lenguaje imperativo sencillo.
6. Nociones sobre eficiencia. Verificación formal de la corrección de algoritmos.

#### Bibliografía

A. Tucker, R. Noonan.

"Lenguajes de programación: principios y paradigmas".

Mc Graw Hill, 2003.

R. D. Tennent.

"Specifying software".

Cambridge University Press, 2002.

R. Peña.

"Diseño de programas: formalismo y abstracción".

Prentice Hall, 1997.

C. Muñoz, A. Niño, A. Vizcaíno.

"Introducción a la programación con orientación a objetos".

Prentice Hall, 2002.

C.T. Wu.

"Introducción a la programación orientada a objetos con Java".

Mc Graw Hill, 2001.

J. Bishop.

"Java: Fundamentos de programación".

Addison-Wesley, 1999.

T. Budd.

"Object-oriented programming".

Addison-Wesley, 2002.

F. Gutiérrez, F. Durán, E. Pimentel.

"Programación orientada a objetos con Java"

Thomson, 2007.

J. Lewis, J. Chase.

"Estructuras de datos con Java. Diseño de estructuras y algoritmos".

Pearson Addison Wesley, 2006.

Página oficial de Java.

[java.sun.com](http://java.sun.com)

## Metodología

### Modalidades organizativas:

- MO1: Clases teóricas
- MO2: Seminarios y talleres
- MO3: Clases prácticas
- MO5: Tutorías
- MO6: Estudio y trabajo autónomo del alumno

### Métodos de enseñanza:

- ME1: Lección magistral
- ME2: Aprendizaje basado en problemas
- ME3: Resolución de ejercicios y problemas
- ME4: Utilización de recursos informáticos

## Organización

Actividades presenciales:	Horas
- Clases teóricas	30
- Clases prácticas de aula	28
- Pruebas presenciales de evaluación	2
<b>Total horas presenciales</b>	<b>60</b>
Actividades no presenciales (trabajo autónomo):	Horas estimadas
- Estudio autónomo individual o en grupo	25

- Resolución individual de ejercicios, cuestiones u otros trabajos, actividades en biblioteca o similar	50
- Preparación en grupo de trabajos, presentaciones (orales, debates,...), actividades en biblioteca o similar	15

<b>Total horas estimadas de trabajo autónomo</b>	90
--	----

<b>Total horas</b>	150
--------------------	-----

### Evaluación

Sistemas de evaluación:	% sobre total	Recuperable/ No Recuperable
<i>Examen final</i>	70	Recuperable
<i>Prácticas en grupo.</i>	10	Recuperable
<i>Prácticas individuales.</i>	20	No recuperable

### Comentario:

*Para los estudiantes a tiempo parcial (reconocidos como tales por la Universidad), las actividades de evaluación no recuperable podrán ser sustituidas por otras, a especificar en cada caso. Esta posibilidad se habilitará siempre y cuando la causa que le impida la realización de la actividad de evaluación programada sea la que ha llevado al reconocimiento de la dedicación a tiempo parcial.*

### Criterios críticos para superar la asignatura:

--