

**Proyecto/Guía docente de la asignatura**

Asignatura	Programación Orientada a Objetos		
Materia	Entorno de Software/Informática		
Módulo	Comunes a la informática		
Titulación	Grado en Ingeniería Informática/Programa de estudios conjunto de Grado en Estadística y de Grado en Ingeniería Informática (INdat)/Grado en Estadística/Máster en Informática		
Plan	545/551/549/510	Código	46914/46914/ 47081/53160
Periodo de impartición	1 <sup>er</sup> Cuatrimestre	Tipo/Carácter	Obligatoria/ Complementos de Informática/ Complementos de Formación
Nivel/Ciclo	Grado	Curso	2º/1º
Crédigos ECTS	6		
Lengua en que se imparte	Castellano		
Profesor responsable	Félix Prieto (Coordinador de teoría), Margarita Gonzalo (Coordinadora de Prácticas),		
Datos de contacto	<i>Teléfono:</i> 983 423 000 ext. 5617 5612 <i>email:</i> [felix marga]@infor.uva.es		
Departamento	Departamento de Informática		



## **1. Situación/Sentido de la asignatura**

---

### **1.1. Contextualización**

---

En la actualidad, la mayoría del desarrollo de software se realiza desde el paradigma Orientado a Objetos. En el contexto de la materia Entorno Software, y con los conocimientos previos adquiridos en otras asignaturas de las materias de Fundamentos Básicos de Informática y Entorno Software, el objetivo de esta asignatura es fijar de una forma clara los conceptos, técnicas y herramientas necesarias para realizar un desarrollo de software de calidad dentro del paradigma Orientado a Objetos.

### **1.2. Relación con otras materias**

---

La asignatura se basa y amplía conceptos presentados en algunas asignaturas de la materia Fundamentos Básicos de Informática, como Fundamentos de Programación, o de la materia Entornos Software como Paradigmas de Programación.

### **1.3. Prerequisitos**

---

Aunque no se han establecido prerequisites, es recomendable que el alumno posea conocimientos básicos de programación, en particular haber cursado con aprovechamiento las asignaturas de Fundamentos de Programación y Paradigmas de Programación.



## 2. Competencias

### 2.1. Generales

Código	Descripción
CG1	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.
CG2	Capacidad para dirigir las actividades objeto de los proyectos del ámbito de la informática.
CG3	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.
CG5	Capacidad para concebir, desarrollar y mantener sistemas, servicios y aplicaciones informáticas empleando los métodos de la ingeniería de software como instrumento para el aseguramiento de su calidad.
CG6	Capacidad para concebir y desarrollar sistemas o arquitecturas informáticas centralizadas o distribuidas integrando hardware, software y redes.
CG10	Conocimientos para la realización de mediciones, cálculos, valoraciones, tasaciones, peritaciones, estudios, informes, planificación de tareas y otros trabajos análogos de informática.

### 2.2. Específicas

Código	Descripción
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.
CI5	Conocimiento, administración y mantenimiento de sistemas, servicios y aplicaciones informáticas.



### 3. Objetivos

Código	Descripción
RA1	Comprender el paradigma de la programación orientada a objeto, su fundamentación teórica y las pautas de su aplicación práctica.
RA2	Emplear correctamente el concepto de objeto y de clase, las relaciones de genericidad y herencia y los mecanismos asociados al polimorfismo en la construcción de programas correctos y fáciles de mantener
RA3	Entender los fundamentos de programación bajo contrato y las ventajas que aporta.
RA4	Ser capaz de proyectar y realizar pruebas de programas en entornos específicos de objetos.
RA5	Saber aplicar bibliotecas y frameworks de objetos al desarrollo de aplicaciones



## 4. Contenidos y/o Bloques temáticos

---

### Bloque 1: "Programación Orientada a Objetos"

---

Carga de trabajo en créditosECTS: 

#### a. Contextualización y justificación

---

Tras introducir los principios del paradigma Orientado a Objetos, abordaremos los conceptos básicos utilizados en el paradigma: Clase y Objeto.

Posteriormente se abordan conceptos más avanzados del paradigma Orientado a Objetos, como son la Genericidad y la Herencia, imprescindibles para la elaboración de sistemas Orientados a Objetos de dificultad moderada.

La aplicación de técnicas Orientadas a Objetos no excluye la necesidad de garantizar la calidad del software construido, más bien al contrario, puesto que las técnicas Orientadas a Objetos permiten abordar problemas de programación más complejos, es más necesario utilizar estrategias que garanticen en la medida de lo posible la calidad del software construido. En este contexto, el tema tres aborda las técnicas de diseño bajo contrato, apropiadas para la verificación de Software Orientado a Objetos, y la forma de adaptar las técnicas de Validación a este paradigma de programación.

#### b. Objetivos de aprendizaje

---

Consulte la sección 3 de este documento

#### c. Contenidos

---

##### TEMA 1 : Clases y Objetos

1. Introducción a la OO
2. Principios de la OO
3. Clases
4. Objetos

##### TEMA 2 : Pruebas

1. Asertos en el código: Contratos Software
2. Tratamiento de excepciones
3. Pruebas en OO
4. Pruebas de caja negra

##### TEMA 3: Genericidad y Herencia

1. Genericidad
2. Herencia
3. Polimorfismo
4. Ligadura dinámica
5. Bibliotecas y Frameworks



#### **d. Métodos docentes**

---

Consulte la sección 5 de este documento

#### **e. Plan de trabajo**

---

Consulte la sección j

#### **f. Evaluación**

---

Consulte la tabla y los criterios del punto 7 de este documento

#### **g. Bibliografía Básica**

---

- Bertrand Meyer  
Construcción de software orientado a objetos, 2ª. ed.,  
Prentice-Hall, 2002 ISBN:8483220407
- Bruce Eckel  
Piensa en Java 4º Ed.  
Prentice-Hall, 2007 ISBN: 9788489660342
- Myers, Glenford J.  
The art of software testing,  
John Wiley & Sons 2004 ISBN: 0471469122
- Wirth, Niklaus  
Introducción a la programación sistemática  
El Ateneo, 1986, ISBN: 9500252341.

#### **h. Bibliografía complementaria**

---

- Bertrand Meyer  
Touch of class: learning to program well with objects and contracts  
Springer, 2009. ISBN 9783540921448
- Harvey Deitel  
Cómo programar en Java  
Pearson 2008,  
isbn: 9789702611905
- Binder, Robert V.  
Testing object-oriented systems: models, patterns, and tools  
Addison-Wesley, 2000

#### **i. Recursos necesarios**

---

Herramientas de programación instaladas en los laboratorios docentes y descargables a partir del aula virtual y/o la página web de los laboratorios de la Escuela.

**j. Temporalización**

Sem.	Teoría	Prácticas	Entrega de trabajos	Evaluación
1	Tema 1	–		
2		Clases y objetos		
3				
4				
5	Tema 2	Pruebas y automatización		
6		Trabajo en el supuesto práctico		
7				
8			Práctica 1	
9	Tema 3	Defensa de la práctica 1		
10		Genericidad y herencia		
11				
12				
13		Trabajo en el supuesto práctico		
14				
15	Práctica 2			



## 5. Métodos docentes y principios metodológicos

Actividad	Metodología
Clase de teoría	Clase magistral participativa
	Estudio de casos en el aula
	Resolución de problemas
Clase práctica	Clase magistral participativa
	Realización, en grupos de dos o tres personas, de dos pequeños sistemas de software que utilicen adecuadamente las técnicas presentadas en la asignatura
	Videos sobre funcionalidades complementarias de las herramientas utilizadas





## 6. Tabla de dedicación del estudiante a la asignatura

Actividades Presenciales	Horas	Actividades no Presenciales	Horas
Clases teórico-prácticas (T/M)	30	Estudio y trabajo autónomo individual	60
Laboratorios (L)	26	Estudio y trabajo autónomo grupal	30
Evaluación (Fuera del periodo oficial de exámenes)	4		
Total Presencial	60	Total no Presencial	90

## 7. Sistema y características de la evaluación

Instrumento/Procedimiento	Peso en la nota final	Observaciones
Entrega de la primera práctica	25 %	Resolución de problemas prácticos realizada en grupos de dos o tres personas
Entrega de la 2ª práctica	25 %	Las entregas individuales sólo serán admitidas en casos excepcionales debidamente justificados
Examen de problemas	50 %	



## Criterios de calificación

■ **Convocatoria ordinaria:**

- Suma ponderada de las prácticas en grupo ( $p_1, p_2$ ) y examen ( $e$ ). Para aprobar será necesario obtener una calificación mínima de 4 en el examen. En concreto:

$$\text{calificación final} = \begin{cases} \sum_{i=1}^2 \frac{25}{100} p_i + \frac{50}{100} e & \text{si } e \geq 4 \\ \min\left(\sum_{i=1}^2 \frac{25}{100} p_i + \frac{50}{100} e, 4,5\right) & \text{si } e < 4 \end{cases}$$

- La primera práctica requiere una defensa realizada en el laboratorio por **todos** los miembros del grupo de prácticas. Esta influye en la nota de prácticas de cada uno de los miembros del grupo. La defensa de la segunda práctica podría ser sustituida por una pregunta en el examen ordinario.

■ **Convocatoria extraordinaria:** Se mantendrá la ponderación de la convocatoria ordinaria con las siguientes puntualizaciones:

- Obligatoriamente se realizará el examen de problemas
- Si no se han entregado las prácticas de la asignatura se podrá optar a una entrega extraordinaria de las mismas, en las condiciones de la convocatoria ordinaria. En este caso la defensa podría ser sustituida por sendas preguntas en el examen extraordinario.

## 8. Consideraciones finales

El trabajo presentado (individual, o en grupo), debe ser fruto del esfuerzo de las personas que lo firman. No se permitirán copias. En caso de dudas, los profesores pondrán en marcha mecanismos adicionales de revisión de los trabajos entregados.

Las competencias y actividades formativas que figuran en esta guía corresponden a las de la memoria del Grado en Informática. Éstas se pueden considerar asimilables a las competencias que se alcanzarían en el Grado en Estadística.