

**Proyecto/Guía docente de la asignatura**

<b>Asignatura</b>	Programación Orientada a Objetos		
<b>Materia</b>	Proceso de Desarrollo del Software		
<b>Módulo</b>			
<b>Titulación</b>	Grado en Ingeniería de Servicios y Aplicaciones		
<b>Plan</b>	413	<b>Código</b>	40812
<b>Periodo de impartición</b>	Semestre 3	<b>Tipo/Carácter</b>	OB
<b>Nivel/Ciclo</b>	Grado	<b>Curso</b>	2
<b>Créditos ECTS</b>	6		
<b>Lengua en que se imparte</b>	Español		
<b>Profesor/es responsable/s</b>	José Vicente Álvarez Bravo		
<b>Datos de contacto (E-mail, teléfono...)</b>	Escuela de Ingeniería Informática de Segovia Plaza de la Universidad, nº 1, 40.005 - Segovia Teléfono: (+34) 921 11 24 54 Fax: (+34) 921 11 24 01 email: jvalvarez@infor.uva.es		
<b>Departamento</b>	Informática ( ATC; CCIA, LSI)		



## 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. Esta asignatura se encuadra en la materia “Proceso de desarrollo del software”, y a partir de los conocimientos previos adquiridos en la asignatura “Fundamentos de Programación” (FP) , introduce 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

Esta asignatura se relaciona directamente con otras asignaturas de la materia “Proceso de desarrollo del software” como son, “Programación y Estructura de Datos”, y “Proceso de desarrollo del software”. También tiene conexión con la asignatura “Modelado Software” perteneciente a la materia “Sistemas de Información”.

### 1.3 Prerrequisitos

Aunque no se han establecido prerrequisitos, es recomendable que el alumno posea conocimientos básicos de programación, en particular haber cursado con aprovechamiento la asignatura “Fundamentos de Programación” y “Metodología de la Programación”.



## 2. Competencias

### 2.1 Generales

- Competencias genéricas G01, G02, G03, G04, G05, G08, G09, G10, G11, G12, G16, G18, G20, G21 y G22.
- Competencia genérica G06 opcional (Si se elige la modalidad bilingüe español-inglés).

### 2.2 Específicas

- Competencias comunes a la rama de informática: E03, E07, E10, E11, E12, E13, E20.
- Competencias de Tecnologías de la Información: E27.
- Competencias de Sistemas de Información: E34.





### 3. Objetivos

- Conocer y aplicar de forma adecuada los conceptos asociados al paradigma de programación orientada a objetos.
- Comprender y aplicar la técnica de herencia para diseñar jerarquías de clases que faciliten la reutilización y mejoren la productividad.
- Comprender el concepto de genericidad y su futuro uso en el diseño de TAD
- Utilizar entornos de programación orientados a objetos.
- Conocer y desarrollar patrones de diseño orientados a objetos.





#### 4. Contenidos y/o bloques temáticos

Bloque Teórico	Contenidos de la asignatura
	Tema 1.- Programación Orientada a Objetos. Conceptos Fundamentales
	Tema 2.- Diseño por contrato
	Tema 3.- Clases, Herencia y Polimorfismo. 3.1. Clases. Estructura estática. 3.2. Objetos. Estructura dinámica. 3.3. Herencia y polimorfismo. Jerarquía de clases.
	Tema 4.- Genericidad. Clases y métodos genéricos
	Tema 5.- Diseño Orientado a Objetos: Patrones.
Bloque Práctico	
	Lab 1.- Conceptos fundamentales de la plataforma Java
	Lab 2.- Introducción al Lenguaje de Programación Java
	Lab 3.- Colecciones en Java
	Lab 4.- I/O en Java
	Lab 5.- Manejo de excepciones

Carga de trabajo en créditos ECTS: 

##### a. Contextualización y justificación

Los alumnos ya conocen el paradigma de la programación modular y estructurada y los conceptos básicos asociados a la programación de computadores (fueron presentados en la asignatura FP). En la asignatura POO vamos a introducir las bases del paradigma de Programación orientado a objetos y pondremos en práctica estos conocimientos con ayuda del lenguaje Java. En ninguna asignatura anterior del plan de estudios se ha trabajado con este lenguaje, por lo que resulta imprescindible ofrecer a los alumnos una primera aproximación al mismo.

##### b. Objetivos de aprendizaje

- ✓ Conocer los conceptos básicos asociados al paradigma de POO.
- ✓ Conocer las características principales del lenguaje de programación Java
- ✓ Trabajar en el laboratorio con un IDE y el paradigma de POO

##### c. Contenidos

En el Tema 1 se proporciona una visión general del paradigma de la programación orientada a objetos, con la intención de que el alumno pueda comprender mejor los conceptos que se van a desarrollar en los temas siguientes. A continuación, se presentan los conceptos básicos asociados a la tecnología asociada al Lenguaje de programación Java así como su sintaxis básica (Anexos 1 y 2). En el Tema 2 se introduce la metodología del Diseño por contrato como una herramienta que permite diseñar y documentar adecuadamente la interfaz pública de una clase. En el Tema 3 se ahonda en las ideas de encapsulamiento, herencia y Polimorfismo a través de ejemplos implementados en Java. En el Anexo 3 se presentan algunas colecciones de Java que permitan al alumno una mayor capacidad en la resolución de problemas y se introduce el concepto de genericidad (Tema 4). Para finalizar, presentamos los conceptos fundamentales que definen el diseño Orientado a Objetos y los patrones básicos de diseño que facilitan el proceso de desarrollo del software bajo este paradigma (tema 5). Como complemento adicional se abordará la Entrada/Salida en Java (API I/O) así como el tratamiento de excepciones (Anexos 4 y 5).





#### d. Métodos docentes

1. Lección magistral: exposición de teoría
2. Prácticas en aula: resolución de problemas
3. Evaluación
4. Estudio autónomo por parte del alumno, incluyendo la realización de problemas, consulta bibliográfica, realización de prácticas y preparación de pruebas de evaluación

#### e. Plan de trabajo

- Alternar sesiones teóricas con clases de problemas y prácticas de laboratorio

#### f. Evaluación

Ver tabla apartado 7.

#### g. Bibliografía básica

- B. Meyer . *Construcción de software orientado a objetos*. 2ª. ed., Prentice-Hall, 2002
- H.M. Deitel y P.J. Deitel. *Java: Cómo programar*. 9ª ed - Ed. Pearson Education, 2012
- L. Joyanes Aguilar. I. Zahonero Martínez. *Programación en Java*. Ed. McGraw-Hill, 2011.

#### h. Bibliografía complementaria

- T. Budd. *Introducción a la programación orientada a objetos*. Addison-Wesley Iberoamericana, 1994
- L. Joyanes Aguilar y L. Zahonero Martínez . *Programación en C, C++, Java y UML*. Ed. McGraw-Hill, 2010.
- F. Durán y otros. *Programación orientada a objetos con Java*. Ed. Thomson , 2007.
- B. Eckel. *Piensa en Java* . 4º Ed. Prentice-Hall, 2007

#### i. Recursos necesarios

Aula con pizarra y ordenador con proyector, biblioteca, sala de estudio y despacho para tutorías.

#### j. Temporalización

BLOQUE TEÓRICO		
CONTENIDOS	CARGA ECTS	PERIODO PREVISTO DE DESARROLLO
TEMA 1	0,2	Semana: 1
TEMA 2	0,2	Semana: 2
TEMA 3.1	0,6	Semanas:3-6
TEMA 3.2	0,7	Semanas: 6-9
TEMA 3.3	0,7	Semanas: 9-12
TEMA 4	0,4	Semanas: 13-14
TEMA 5	0,2	Semana: 15



BLOQUE PRÁCTICO		
CONTENIDOS	CARGA ECTS	PERIODO PREVISTO DE DESARROLLO
LAB 1	0,2	Semana: 1
Práctica Guiada	LAB 2	Semana: 2-4
	LAB 3	Semanas: 5
	LAB 4	Semanas: 6
	LAB 5	Semanas: 7
Proyecto final	1,6	Semanas: 8-15

## 5. Métodos docentes y principios metodológicos



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

ACTIVIDADES PRESENCIALES	HORAS	ACTIVIDADES NO PRESENCIALES	HORAS
Clases teórico-prácticas (T/M)	30 h	Estudio y trabajo autónomo individual (conocer, comprender, plantear dudas, experimentar)	35 h
Laboratorios (L)	30 h	Estudio y trabajo autónomo individual (preparación de prácticas)	55 h
Total presencial	<b>60</b>	<b>Total no presencial</b>	<b>90</b>

**7. Sistema y características de la evaluación**

INSTRUMENTO/PROCEDIMIENTO	PESO EN LA NOTA FINAL	OBSERVACIONES
Examen de Teoría y Problemas	70%	Uno por cada convocatoria (Junio, Julio)
Realización y defensa de prácticas y/o proyectos	30%	Evaluación continua (todo el semestre) La nota final consistirá: <ul style="list-style-type: none"><li>• Evaluación continua (10%)</li><li>• Proyecto Final (20%)</li></ul>

**CRITERIOS DE CALIFICACIÓN**

- **Convocatoria ordinaria:**
  - Para aprobar la asignatura es necesario obtener, como mínimo, un 5 sobre 10 en cada parte (teórica y práctica).
- **Convocatoria extraordinaria:**
  - Para aprobar la asignatura es necesario obtener, como mínimo, un 5 sobre 10 en cada parte (teórica y práctica).

**8. Consideraciones finales**

Todos los recursos docentes de la asignatura, bibliografía incluida, estarán disponibles en el espacio dedicado a la asignatura en el campus virtual de la Uva: [campusvirtual.uva.es](http://campusvirtual.uva.es).