



## **GUÍA DOCENTE**

### **PROGRAMACIÓN ORIENTADA A OBJETOS**

#### **DOBLE GRADO EN INGENIERÍA DEL SOFTWARE**

***MODALIDAD: PRESENCIAL***

***CURSO ACADÉMICO: 2025-2026***

Denominación de la asignatura:	<b>Programación Orientada a Objetos</b>
Titulación:	Doble Grado en Ingeniería del Software
Facultad o Centro:	Centro Universitario de Tecnología y Arte Digital
Materia:	Programación
Curso:	2
Cuatrimestre:	1
Carácter:	OB
Créditos ECTS:	6
Modalidad/es de enseñanza:	Presencial
Idioma:	Castellano
Profesor/a - email	Miguel Angel Mesas Uzal / miguel.mesas@u-tad.com
Página Web:	<a href="http://www.u-tad.com/">http://www.u-tad.com/</a>

## DESCRIPCIÓN DE LA ASIGNATURA

### Descripción de la materia

Esta asignatura pertenece a la materia de programación. Esta materia se dedica al estudio de las técnicas y los lenguajes de programación en los que se fundamentarán los estudios del grado de ingeniería del software.

### Descripción de la asignatura

Esta asignatura le permite al alumno adquirir los conocimientos clave sobre el paradigma de programación orientado a objetos, realizar programas estructurados en clases y dotados de mecanismos como la herencia y el polimorfismo, así como trabajar con un lenguaje de programación orientado a objetos.

## COMPETENCIAS Y RESULTADOS DE APRENDIZAJE DE LA MATERIA

## Competencias (genéricas, específicas y transversales)

### COMPETENCIAS BÁSICAS Y GENERALES:

CG1 - Capacidad para entender, planificar y resolver problemas a través del desarrollo de soluciones informáticas

CG3 - Conocimiento de los fundamentos científicos aplicables a la resolución de problemas informáticos

CG4 - Capacidad para simplificar y optimizar los sistemas informáticos atendiendo a la comprensión de su complejidad

CG9 - Capacidad para aprender, modificar y producir nuevas tecnologías informáticas

CG10 - Uso de técnicas creativas para la realización de proyectos informáticos

CB1 - Que los estudiantes hayan demostrado poseer y comprender conocimientos en un área de estudio que parte de la base de la educación secundaria general, y se suele encontrar a un nivel que, si bien se apoya en libros de texto avanzados, incluye también algunos aspectos que implican conocimientos procedentes de la vanguardia de su campo de estudio

CB2 - Que los estudiantes sepan aplicar sus conocimientos a su trabajo o vocación de una forma profesional y posean las competencias que suelen demostrarse por medio de la elaboración y defensa de argumentos y la resolución de problemas dentro de su área de estudio

CB3 - Que los estudiantes tengan la capacidad de reunir e interpretar datos relevantes (normalmente dentro de su área de estudio) para emitir juicios que incluyan una reflexión sobre temas relevantes de índole social, científica o ética

CB4 - Que los estudiantes puedan transmitir información, ideas, problemas y soluciones a un público tanto especializado como no especializado

CB5 - Que los estudiantes hayan desarrollado aquellas habilidades de aprendizaje necesarias para emprender estudios posteriores

con un alto grado de autonomía

### COMPETENCIAS ESPECÍFICAS

CE1 - Conocimiento de la estructura de las computadoras, de los conceptos de codificación, manipulación, tratamiento de la información y uso de lenguajes de bajo nivel

CE7 - Conocimiento de los principales tipos de estructuras de datos y utilización de las librerías y de los técnicas algorítmicas asociadas a dichas estructuras junto con los órdenes de complejidad que caracterizan a dichas técnicas

CE8 - Conocimiento de los distintos paradigmas detrás de los lenguajes de programación

CE9 - Conocimiento de las estructuras de control, variables, sintaxis de programación y gestión del uso de la memoria de manera eficaz en el desarrollo de una aplicación informática

CE10 - Capacidad para manejar un gestor de versiones de código y generar la documentación de una aplicación de forma automática.

CE15 - Conocimiento de la tolerancia a los fallos, la adaptabilidad, el balance de carga y la predictividad del sistema para el desarrollo de aplicaciones distribuidas

CE17 - Conocimiento de las características de paralelización de tarjetas gráficas y de arquitecturas de altas prestaciones para el desarrollo de aplicaciones.

CE20 - Capacidad para testar el funcionamiento y funcionalidad de una aplicación informática, elaborando planes de pruebas y empleando técnicas de diseño y programación orientado a las pruebas

CE23 - Conocimiento de los principios de la inteligencia artificial y uso de algoritmos de búsqueda deterministas y máquinas de estado

### Resultados de aprendizaje

Al acabar la titulación, el graduado o graduada será capaz de:

- Entender y manejar el concepto de memoria dinámica
- Identificar clases de objetos con los datos de un problema.
- Crear clases y objetos y manipularlos.
- Entender y utilizar los mecanismos de herencia, polimorfismo y sobrecarga de operadores.
- Identificar las relaciones entre clases en distintos casos de uso.
- Dominar un lenguaje de programación orientado a objetos.
- Dominar los patrones de programación
- Conocer las distintas formas de resolución de problemas desde el punto de vista de la algoritmia, como, por ejemplo, el esquema divide y vencerás,
- programación dinámica, backtracking o algoritmos genéticos.
- Estudiar la complejidad de un determinado algoritmo, interpretar dicha complejidad y analizar posibles optimizaciones.
- Codificar un programa que sea capaz de encontrar el camino óptimo que une dos nodos de un grafo siguiendo los distintos algoritmos de pathfinding.
- Crear y entrenar redes neuronales que solucionen problemas concretos.

## CONTENIDO

Conceptos básicos de la programación orientada a objetos

Manejo de excepciones

## TEMARIO

Tema 1: Introducción a la Programación Orientada a Objetos

Programación Procedural vs. Programación Orientada a Objetos.

Objetos y clases.

Propiedades de la orientación a objetos.

Tema 2: Programación en lenguaje Java

Entorno de desarrollo.o Operadores.

Bifurcaciones.o Bucles.o Variables.o Arrays.

Listas de argumentos.

Tema 3: Clases en Java

Concepto de clase.

Objetos y métodos.

Variables miembro.

Constructores.

Ámbito de las variables.

Uso de this.

Paquetes.

Relaciones de composición.

Tema 4: Herencia o Jerarquías de herencia.

Conversión de objetos.

Polimorfismo.

Clases abstractas.o Interfaces.

Tema 5: Manejo de excepciones

Excepciones estándar en Java.

Lanzamiento y captura de excepciones.

Creación de excepciones.

Tema 6: Programación de GUIs

Ventanas.

Botones.

Manejo de eventos.

Componentes gráficos de texto.

Gestores de disposición.

Tema 7: Entrada y salida de datos en Java

Clases para lectura y escritura.

Entrada y salida estándar.

La clase Scanner.

Ficheros de texto.

Ficheros binarios.

Tema 8: Plantillas y programación genérica.

Clases genéricas.

Métodos genéricos

## ACTIVIDADES FORMATIVAS Y METODOLOGÍAS DOCENTES

### Actividades formativas

Actividad Formativa	Horas totales	Horas presenciales
<i>Clases teóricas / Expositivas</i>	35,6363636363636	35,6363636363636
<i>Clases Prácticas</i>	18,9090909090909	18,9090909090909
<i>Tutorías</i>	4	2
<i>Estudio independiente y trabajo autónomo del alumno</i>	51,8181818181818	0
<i>Elaboración de trabajos (en grupo o individuales)</i>	33,8181818181818	0
<i>Actividades de Evaluación</i>	5,81818181818182	5,81818181818182
<i>Preparación y defensa del TFG</i>	<<7- Preparación y defensa del TFG>>	<<Horas presenciales 7- Preparación y defensa del TFG>>

### Metodologías docentes

Método expositivo o lección magistral

Aprendizaje de casos

Aprendizaje basado en la resolución de problemas

Aprendizaje cooperativo o colaborativo

Aprendizaje por indagación

Metodología Flipped classroom o aula invertida

Gamificación

Just in time Teaching (JITT) o aula a tiempo

Método expositivo o lección magistral

Método del caso

Aprendizaje basado en la resolución de problemas

Aprendizaje cooperativo o colaborativo

Aprendizaje por indagación

Metodología flipped classroom o aula invertida

Gamificación

## DESARROLLO TEMPORAL

### UNIDADES DIDÁCTICAS / TEMAS PERÍODO TEMPORAL

Tema 1: Introducción a la Programación Orientada a Objetos 1 semana

Tema 2: Programación en lenguaje Java 2 semanas

Tema 3: Clases en Java 3 semanas

Tema 4: Herencia 3 semanas

Tema 5: Manejo de excepciones 1 semana

Tema 6: Programación de GUIs 2 semana

Tema 7: Entrada y salida de datos en Java 2 semana

Tema 8: Plantillas y programación genérica. 1 semana

## SISTEMA DE EVALUACIÓN

ACTIVIDAD DE EVALUACIÓN	VALORACIÓN MÍNIMA RESPECTO	VALORACIÓN MÁXIMA RESPECTO
-------------------------	-------------------------------	-------------------------------



	A LA CALIFICACIÓN FINAL (%)	A LA CALIFICACIÓN FINAL (%)
<i>Evaluación de la participación en clase, en prácticas o en proyectos de la asignatura</i>	0	30
<i>Evaluación de trabajos, proyectos, informes, memorias</i>	30	80
<i>Prueba Objetiva</i>	10	60
<i>Evaluación del TFG</i>	<<4-(MIN )Evaluación del TFG>>	0

## CRITERIOS ESPECÍFICOS DE EVALUACIÓN

ACTIVIDAD DE EVALUACIÓN	CONVOCATORIA ORDINARIA	CONVOCATORIA EXTRAORDINARIA
<i>Evaluación de la participación en clase, en prácticas o en proyectos de la asignatura</i>	15	15
<i>Evaluación de trabajos, proyectos, informes, memorias</i>	25	25
<i>Prueba Objetiva</i>	60	60
<i>Evaluación del TFG</i>	<<4-(MIN )Evaluación del TFG>>	0

### Consideraciones generales acerca de la evaluación

Apartados oficiales de evaluación

- Participación en clase (15%)
  - o Tests por tema: se realizan al finalizar algunos temas, de forma individual y supervisada. No se repiten y su calificación se conserva para la convocatoria extraordinaria dentro del mismo curso académico.
- Evaluación de trabajos, proyectos, informes, memorias (25%)
  - o Proyecto final: se desarrolla a lo largo del curso y se entrega en la fecha indicada a través de la plataforma virtual. Se evalúa mediante entrega y demostración. No se admitirán entregas fuera de plazo; de aceptarse excepcionalmente por causa justificada, el retraso conllevará una reducción considerable de la calificación.
- Prueba objetiva (60%)
  - o Examen final con dos partes: teórica tipo test y práctica. Ambas son obligatorias.

### Cálculo de la calificación

- Nota final =  $0,15 \times (\text{Participación en clase}) + 0,25 \times (\text{Evaluación de trabajos, proyectos, informes, memorias}) + 0,60 \times (\text{Prueba objetiva})$ , esto es:

$0,15 \times \text{tests por tema} + 0,25 \times \text{proyecto} + 0,60 \times \text{examen final}$ .

- Para superar la asignatura es imprescindible alcanzar  $\geq 5,0/10$  en la nota final.
- Para que los componentes sean compensables, se exige una calificación mínima de 4,0 tanto en el proyecto como en el examen final (considerado en su conjunto).

### Convocatorias

#### Convocatoria ordinaria (evaluación habitual)

- Ámbito: evaluación normal del curso (participación en clase –tests por tema–, proyecto y examen final).
- Ponderaciones y mínimos: se aplican las establecidas en los apartados anteriores (15% participación en clase, 25% proyecto, 60% examen; mínimos de 4,0 en proyecto y examen; 5,0 final para aprobar).
- Tests por tema: no se repiten y conservan su calificación para la extraordinaria dentro del mismo curso académico.

#### Convocatoria extraordinaria

- Destinatarios: estudiantes que no hayan superado la convocatoria ordinaria o a quienes se les haya concedido convocatoria única.
- Ponderaciones y estructura: se mantienen las de la ordinaria (15% participación en clase —tests por tema— —calificación conservada, no se repiten—, 25% proyecto y 60% examen final con parte teórica tipo test y parte práctica).
- Proyecto: si no alcanzó 4,0 en ordinaria o no se realizó, podrá presentarse/reentregarse en la fecha indicada para esta convocatoria.
- Criterios de superación: se mantienen los mínimos de 4,0 en proyecto y examen final como condición de compensación y la exigencia de nota final  $\geq 5,0/10$ .

### Otras normas

- Conservación de calificaciones entre cursos: no se conservarán calificaciones de ningún tipo entre distintos cursos académicos.
- Uso de dispositivos en el aula:
  - o Teléfonos móviles: no está permitido su uso durante el período de evaluación continua, salvo indicación expresa del profesorado.
  - o Ordenadores portátiles: su uso se limita a actividades relacionadas con la asignatura. El profesorado podrá retirar el derecho de uso si se emplean para fines ajenos (p. ej., correo, noticias, redes sociales u otras asignaturas).
- Participación: se demanda una participación activa del estudiantado, necesaria para el correcto desarrollo de las clases.

- Comportamiento: se exige un comportamiento adecuado en todo momento. Las conductas que impidan el normal desarrollo de la clase podrán conllevar la expulsión temporal del aula, a criterio del profesorado

## **BIBLIOGRAFÍA / WEBGRAFÍA**

Básica:

Deitel & Deitel. Java Cómo programar.

Javier García de Jalón et. Al., Aprenda Java como si estuviera en primero. Universidad de Navarra.

Roger Cadenhead, Java 8. Anaya Multimedia.

Recomendada:

Matt Weisfeld, The Object-Oriented Thought Process.

Addison Wesley. David Etheridge, Java: The Fundamentals of Objects and Classes – An Introduction to Java Programming. Ventus Publishing ApS.

C. Thomas Wu, Object-Oriented Programming with Java. Mc Graw Hill.

Kathy Sierra & Bert Bates, Head First Java. O'Reilly.

David J. Eck, Introduction to Programming Using Java. Hobart and William Smith Colleges

## **MATERIALES, SOFTWARE Y HERRAMIENTAS NECESARIAS**

### **Tipología del aula**

Aula teórica

Equipo de proyección y pizarra

### **Materiales:**

Ordenador personal con Windows o Linux

### **Software:**

JDK de Java

Eclipse