

**CENTRO UNIVERSITARIO DE TECNOLOGÍA Y ARTE DIGITAL**



**PLANIFICACIÓN DE LA DOCENCIA  
UNIVERSITARIA**

**GUÍA DOCENTE**

**PROGRAMACIÓN ORIENTADA A OBJETOS**

# 1. DATOS DE IDENTIFICACIÓN DE LA ASIGNATURA.

Título:	Grado en Ingeniería del Software
Facultad:	Centro Universitario de Tecnología y Arte Digital (U-TAD)
Materia:	Programación
Denominación de la asignatura:	Programación Orientada a Objetos
Curso:	2
Cuatrimestre:	1
Carácter:	Básica
Créditos ECTS:	6
Modalidad/es de enseñanza:	Híbrido Presencial
Idioma:	Castellano
Profesor/a:	Jose Jesus García Rueda Manoel Fernando Alonso Gadi
E-mail:	jose.rueda@u-tad.com manoel.alonso@u-tad.com
Teléfono:	91 640 28 11 (ext. 213)

## 2. DESCRIPCIÓN DE LA ASIGNATURA

### 2.1 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.

### 2.2 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.

### 3. COMPETENCIAS

#### Competencias Básicas y Generales

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

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

#### 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

## 4. CONTENIDOS

- **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.
  - Operadores.
  - Bifurcaciones.
  - Bucles.
  - Variables.
  - 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**
  - Jerarquías de herencia.
  - Conversión de objetos.
  - Polimorfismo.
  - Clases abstractas.
  - 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.

## 5. ACTIVIDADES FORMATIVAS Y MODALIDADES DE ENSEÑANZA

### 5.1 Modalidades de enseñanza

La asignatura se desarrollará a través de los siguientes métodos y técnicas generales, que se aplicarán diferencialmente según las características propias de la asignatura:

- **Método expositivo/Lección magistral:** el profesor desarrollará, mediante clases magistrales y dinámicas los contenidos recogidos en el temario.
- **Estudio de casos:** análisis de casos reales relacionados con la asignatura.
- **Resolución de ejercicios y problemas:** los estudiantes desarrollarán las soluciones adecuadas aplicando procedimientos de transformación de la información disponible y la interpretación de los resultados.
- **Aprendizaje basado en problemas:** utilización de problemas como punto de partida para la adquisición de conocimientos nuevos.
- **Aprendizaje orientado a proyectos:** se pide a los alumnos que, en pequeños grupos, planifiquen, creen y evalúen un proyecto que responda a las necesidades planteadas en una determinada situación.
- **Aprendizaje cooperativo:** Los estudiantes trabajan en grupo para realizar las tareas de manera colectiva.

### 5.2 Actividades formativas

Actividad Formativa	Horas	Presencialidad
AF1 Clases teóricas / Expositivas	30	100%
AF2 Clases Prácticas	24	100%
AF3 Tutorías	6	50%
AF4 Estudio independiente y trabajo autónomo del alumno	57	0%
AF5 Elaboración de trabajos (en grupo o individuales)	29	0%
AF6: Actividades de Evaluación	4	100%

## 6. 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

## 7. SISTEMA DE EVALUACIÓN

ACTIVIDAD DE EVALUACIÓN	VALORACIÓN MÍNIMA RESPECTO A LA CALIFICACIÓN FINAL (%)	VALORACIÓN MÁXIMA RESPECTO A LA CALIFICACIÓN FINAL (%)
SE1 Evaluación de la participación en clase, en prácticas o en proyectos de la asignatura	0%	30%
SE2 Evaluación de trabajos, proyectos, informes, memorias	30%	80%
SE3 Prueba Objetiva	10%	60%

## 8. CRITERIOS DE EVALUACIÓN

ACTIVIDAD DE EVALUACIÓN	CRITERIOS DE EVALUACIÓN	VALORACIÓN RESPECTO A LA CALIFICACIÓN FINAL (%)
SE1 Evaluación de la participación en clase, en prácticas o en proyectos de la asignatura	Interés, participación, autoevaluación...	0%
SE2 Evaluación de trabajos, proyectos, informes, memorias	Calidad del diseño orientado a objetos, funcionamiento del código, limpieza y claridad del código.	40%
SE3 Prueba Objetiva	Demostración de asimilación de conceptos y habilidades	60%

- Para aprobar la asignatura se deberá tener una nota final igual o superior a 5.0 (sobre 10).
- Para aprobar la asignatura se deberá tener una nota igual o superior a 5.0 (sobre 10) en cada una de las partes (proyecto y prueba objetiva).
- Las partes aprobadas en convocatoria ordinaria se guardan para la extraordinaria.

**Consideraciones generales acerca de la evaluación:**

. El alumno irá demostrando con su trabajo en el aula, sus entregas y actitud general hacia su aprendizaje si está capacitado en las competencias que de él se esperan desarrollar en esta asignatura.

. No se admitirán trabajos fuera de forma y fecha sin causa justificada. Y si se aceptasen, el retraso podría suponer una reducción considerable en la nota.



## 5. BIBLIOGRAFÍA / WEBGRAFÍA

<b>Básica:</b>
Deitel & Deitel. Java Cómo programar.
Javier García de Jalón et. Al., Aprende Java como si estuviera en primero. Universidad de Navarra.
Roger Cadenhead, Java 8. Anaya Multimedia.
<b>Recomendada:</b>
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.

## 6. MATERIAL, SOFTWARE Y HERRAMIENTAS NECESARIAS

### **MATERIALES:**

#### **Materiales necesarios del alumno:**

- Ordenador personal con Windows o Linux.
- JDK de Java.
- Eclipse.