

# Guía Docente

Modalidad Presencial

## Programación Orientada a Objetos II

Curso 2017/18

**G**rado en Ingeniería de  
Sistemas de Información



**UCAV**

[www.ucavila.es](http://www.ucavila.es)





<b>Nombre:</b>	PROGRAMACIÓN ORIENTADA A OBJETOS II
<b>Carácter:</b>	OBLIGATORIA
<b>Código:</b>	30203GH
<b>Curso:</b>	3º
<b>Duración (Semestral/Anual):</b>	SEMESTRAL
<b>Nº Créditos ECTS:</b>	6
<b>Prerrequisitos:</b>	NINGUNO
<b>Responsable docente:</b>	MIGUEL ÁNGEL GUTIÉRREZ GARCÍA DOCTOR EN INFORMÁTICA ESPECIALIZADO EN LENGUAJES Y SISTEMAS INFORMÁTICOS.
<b>Email:</b>	miguel.gutierrez@ucavila.es
<b>Departamento (Área Departamental):</b>	TECNOLÓGICO
<b>Lengua en la que se imparte:</b>	CASTELLANO
<b>Módulo:</b>	CONTENIDOS COMUNES A LA INGENIERÍA INFORMÁTICA
<b>Materia:</b>	PROGRAMACIÓN

La programación orientada a objetos es, actualmente, el paradigma de programación más utilizado. En esta asignatura se pretende profundizar en los conceptos de la programación orientada a objetos y en su aplicación en el desarrollo de aplicaciones.

### 2.1. 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.
- G1. 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.

- G3. 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.
- G6. Capacidad para concebir y desarrollar sistemas o arquitecturas informáticas centralizadas o distribuidas integrando hardware, software y redes.
- G8. Conocimiento de las materias básicas y tecnologías, que capaciten para el aprendizaje y desarrollo de nuevos métodos y tecnologías, así como las que les doten de una gran versatilidad para adaptarse a nuevas situaciones.
- G9. Capacidad para resolver problemas con iniciativa, toma de decisiones, autonomía y creatividad. Capacidad para saber comunicar y transmitir los conocimientos, habilidades y destrezas de la profesión de Ingeniero Técnico en Informática, tanto en lengua nativa como en lengua inglesa.

## 2.2. COMPETENCIAS ESPECÍFICAS

- E4. Conocimiento de los fundamentos del uso y programación de los computadores, los sistemas operativos, las bases de datos y, en general, los programas informáticos con aplicación en ingeniería.
- E7. Capacidad para diseñar, desarrollar, seleccionar y evaluar aplicaciones y sistemas informáticos, asegurando su fiabilidad, seguridad y calidad, conforme a principios éticos y a la legislación y normativa vigente.
- E9. Conocimiento y aplicación de los procedimientos algorítmicos básicos de las tecnologías informáticas para diseñar soluciones a problemas, analizando la idoneidad y complejidad de los algoritmos propuestos.
- E11. Capacidad para analizar, diseñar, construir y mantener aplicaciones de forma robusta, segura y eficiente, eligiendo el paradigma y los lenguajes de programación más adecuados.
- E16. Conocimiento y aplicación de las herramientas necesarias para el almacenamiento, procesamiento y acceso a los Sistemas de información, incluidos los basados en web.

- E33. Capacidad para participar activamente en la especificación, diseño, implementación y mantenimiento de los sistemas de información y comunicación.

## 2.3. RESULTADOS DE APRENDIZAJE

Diseñar e implementar programas utilizando el paradigma de la programación orientada a objetos.

3



Contenidos de la asignatura

## 3.1. PROGRAMA

### UNIDAD 1. INTRODUCCIÓN

- 1.1. ¿Qué es java?
- 1.2. Sencillo
- 1.3. Orientado a objetos
- 1.4. Distribuido, seguro y multihilo
- 1.5. Interpretado y multiplataforma
- 1.6. Entorno de desarrollo

### UNIDAD 2. EL ENTORNO DE DESARROLLO

- 2.1. El entorno de desarrollo
  - 2.1.1. Variables de entorno
  - 2.1.2. Compilador de java
  - 2.1.3. Lanzador de aplicaciones
  - 2.1.4. Depurador de aplicaciones
  - 2.1.5. Generador de documentación
- 2.2. Introducción a eclipse

### UNIDAD 3. SÍNTAXIS BÁSICA

- 3.1. Sintaxis básica
  - 3.1.1. Comentarios
  - 3.1.2. Sentencias y bloques

3.1.3. Identificadores

3.1.4. Variables

3.1.5. Operadores

3.2. Tipos de datos primitivos

3.3. Estructuras de control de flujo

3.3.1. Estructuras de control selectivas

3.3.2. Estructuras de control iterativas

3.3.3. Sentencias de ruptura

#### UNIDAD 4. CLASES

4.1. Definición de una clase

4.2. Referencias y objetos

4.3. Atributos

4.3.1. Atributos de clase

4.4. Métodos

4.4.1. Paso de argumentos

4.4.2. Métodos sobrecargados

4.4.3. Métodos de clase

4.4.4. El método main

4.5. Constructores

4.6. Finalizadores

4.7. La clase Array

4.8. La clase String

#### UNIDAD 5. HERENCIA

5.1. Herencia

5.1.1. Redefinición de métodos heredados

5.1.2. Clases y métodos abstractos

5.2. Las referencias super y this

#### UNIDAD 6. POLIMORFISMO

6.1. Polimorfismo

6.1.1. Colecciones genéricas

6.1.2. Métodos genéricos

6.2. Castings

#### UNIDAD 7. INTERFACES

7.1. Interfaces

7.1.1. Polimorfismo en interfaces

## 7.2. Interfaces vs clases abstractas

### UNIDAD 8. PAQUETES

#### 8.1. Paquetes

##### 8.1.1. Visibilidad en los paquetes

##### 8.1.2. Relación con el árbol de directorios

#### 8.2. Ejemplos

### UNIDAD 9. COLECCIONES

#### 9.1. Colecciones

##### 9.1.1. Interfaces

##### 9.1.2. Clases

##### 9.1.3. Tipos genéricos

##### 9.1.4. Iteradores

#### 9.2. Clases del framework

##### 9.2.1. Vector

##### 9.2.2. LinkedList

##### 9.2.3. HashSet

##### 9.2.4. TreeMap

### UNIDAD 10. EXCEPCIONES

#### 10.1. Excepciones

##### 10.1.1. Clases y excepciones

#### 10.2. Creación de excepciones

#### 10.3. Bloques try-catch-finally

#### 10.4. Lanzamiento de excepciones

##### 10.4.1. Excepciones y polimorfismo

##### 10.4.2. Excepciones y herencia

### UNIDAD 11. EMPAQUETAR APLICACIONES

#### 11.1. Separar código fuente y bytecode

#### 11.2. Empaquetado

#### 11.3. Ficheros jar ejecutables

### UNIDAD 12. SERIALIZACIÓN (I)

#### 12.1. Ficheros y directorios

#### 12.2. Streams

##### 12.2.1. Streams de entrada

##### 12.2.2. Streams de salida

#### 12.3. Entrada/Salida estándar



#### 12.4. Serialización

##### 12.4.1. Proteger información sensible

### UNIDAD 13. THREADS

#### 13.1. Threads

##### 13.1.1. La clase Thread

##### 13.1.2. La interfaz Runnable

#### 13.2. Ciclo de vida de un thread

##### 13.2.1. Creación de un thread

##### 13.2.2. Arrancar un thread

##### 13.2.3. Parar un thread

##### 13.2.4. Reanudar un thread

##### 13.2.5. Terminar un thread

##### 13.2.6. Estado de un thread

#### 13.3. Interrupción de threads

##### 13.3.1. Interrupción de threads en ejecución

##### 13.3.2. Interrupción de threads suspendidos

#### 13.4. Grupos de threads

#### 13.5. Prioridades

#### 13.6. Sincronización

##### 13.6.1. Synchronized

##### 13.6.2. Señales

### UNIDAD 14. SERIALIZACIÓN (II)

#### 14.1. La clase URL

#### 14.2. La clase Socket

##### 14.2.1. Servidor multihilo

### UNIDAD 15. ACCESO A BASES DE DATOS

#### 15.1. Drivers JDBC

#### 15.2. El paquete java.sql

##### 15.2.1. La clase DriverManager

##### 15.2.2. La interfaz Connection

##### 15.2.3. La interfaz Statement

##### 15.2.4. La interfaz ResultSet

#### 15.3. Excepciones

#### 15.4. Metainformación

#### 15.5. Transacciones

### 3.2. BIBLIOGRAFÍA

- Cay S. Horstmann, Gary Cornell. Core Java. Volume I. Fundamentals. Sun Microsystems Press.
- Cay S. Horstmann, Gary Cornell. Core Java. Volume II. Advanced Features. Sun Microsystems Press.
- Bruce Eckel. Thinking in Java. Prentice Hall.
- David Flanagan. Java in a Nutshell. O'Reilly Media Inc.
- David Flanagan. Java Examples in a Nutshell. O'Reilly Media Inc.
- Herbert Schildt. Java: a beginner's guide. McGraw-Hill.

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

- **Exposición del profesor:** mediante la clase magistral el profesor de la asignatura expondrá y explicará a los alumnos los contenidos principales de la misma, fomentando la participación y la opinión crítica de los alumnos. Estas sesiones tienen como objetivo transmitir conocimiento y activar procesos cognitivos en el estudiante. Durante las sesiones de teoría se podrán plantear preguntas o situaciones problemáticas sobre un tema, ocasionar debate individual o en grupo y resolver dudas que se puedan plantear.
- **Estudio del alumno:** el alumno acometerá de forma individual el estudio de la asignatura de modo que le permita adquirir las competencias de la misma.
- **Ejercicios y problemas prácticos:** consistirán en la resolución por parte del alumno, individualmente, de problemas u otros ejercicios propios de la disciplina correspondiente y que les permita adquirir las consecuentes

competencias. Podrá exigirse a los alumnos, de acuerdo con lo que se establezca en la guía docente, la entrega de dichos ejercicios con el fin de tenerlo en cuenta a la hora de evaluar.

- **Prácticas con ordenador:** el alumno realizará individualmente las diferentes prácticas según las competencias y actividades correspondientes que deba trabajar durante el semestre. Se realizarán en el aula de ordenadores con el apoyo del profesor, posteriormente el alumno podrá repetir las prácticas en su casa.
- **Preparación y realización de trabajos:** el alumno elaborará los diferentes trabajos según las competencias y actividades correspondientes que deba trabajar durante el semestre.
- **Tutorías personalizadas:** El profesor pondrá a disposición del alumno un tiempo para que éste pueda plantear cuantas dudas le surjan en el estudio de la materia.
- **Realización de examen escrito sobre materia:** El alumno realizará un examen final que junto con los trabajos obligatorios servirán para la evaluación de éste en la materia.
- **Revisión:** El profesor establecerá un horario para que el alumno pueda ponerse en contacto con él para poder revisar la evaluación de la asignatura.



La evaluación es un componente fundamental de la formación del alumno. Está compuesta por un examen final escrito y la evaluación continua, que consta de ejercicios y actividades evaluables.

La evaluación de esta asignatura se realiza mediante la media del examen (valorado en un 60%) y la realización de un trabajo obligatorio individual (con valor del 40%).

- Examen (60 % de la nota final)

La superación de dicho examen constituye un requisito indispensable para la superación de la asignatura. El alumno deberá tener en el examen al menos un 5 para poder realizar la ponderación de notas. El alumno con nota inferior se considerará suspenso. El alumno dispondrá de dos convocatorias de examen por curso académico.

No se guardará la nota del examen, si éste estuviera aprobado, para una convocatoria posterior.

➤ Trabajo obligatorio (40% de la nota final)

La superación del trabajo constituye un requisito indispensable para la superación de la asignatura. El alumno deberá tener en el trabajo al menos un 5 para poder realizar la ponderación de notas. El alumno con nota inferior se considerará suspenso. En el caso de tener el trabajo obligatorio superado y no aprobar el examen, se guardará su nota hasta la segunda convocatoria de examen perteneciente al curso académico actual.

No se admitirán trabajos fuera de la fecha límite de entrega, que será comunicada al alumno con suficiente antelación. Con la no presentación del trabajo obligatorio se considerará suspensa la asignatura, independientemente de la nota obtenida en el examen.

### **Criterios de calificación de la evaluación continua**

---

Los criterios para la evaluación del trabajo obligatorio se presentan en la siguiente tabla, donde se resumen los aspectos a valorar y el porcentaje que representa cada uno de los mismos:

COMPONENTES EVALUABLES	PROPORCIÓN
La aplicación compila y se ejecuta correctamente	50%
Estructuración en clases, interfaces, etc.	10%
Selección y uso de la estructura de datos de la aplicación	10%
Control de excepciones	5%

Control de la concurrencia	10%
Acceso correcto a base de datos	5%
Trabajo en red	10%
<b>TOTAL</b>	<b>100%</b>

6

Apoyo tutorial

Para el apoyo tutorial, el alumno tendrá a su disposición un equipo docente encargado de acompañar al alumno durante toda su andadura en el proceso formativo, prestando una atención personalizada al alumno. Las dos figuras principales son:

**Profesor docente:** encargado de resolver todas las dudas específicas de la asignatura y de informar al alumno de todas las pautas que debe seguir para realizar el estudio de la asignatura.

**Tutor personal o de grupo:** asignado al alumno al iniciar los estudios de Grado y que orienta al alumno tanto en cuestiones académicas como personales.

**Horario de Tutorías del profesor docente: los martes de 16:00 a 18:00 h.**

7

Horario de la asignatura y calendario de temas

Horario de la asignatura: **miércoles de 9:00 a 11:00 y jueves de 16:00 a 18:00**

Las sesiones se desarrollarán según la siguiente tabla. El plan de trabajo y las semanas son orientativas, pudiendo variar ligeramente, dependiendo de la evolución del alumno durante las distintas sesiones.

UNIDADES DIDÁCTICAS	HORAS DEDICACIÓN
Unidad 1 – Introducción	7,5 HORAS
Unidad 2 – El entorno de desarrollo	7,5 HORAS
Unidad 3 – La sintaxis básica	7,5 HORAS
Unidad 4 – Clases	7,5 HORAS
Unidad 5 – Herencia	7,5 HORAS
Unidad 6 – Polimorfismo	7,5 HORAS
Unidad 7 – Interfaces	7,5 HORAS
Unidad 8 – Paquetes	7,5 HORAS
Unidad 9 – Colecciones	15 HORAS
Unidad 10 – Excepciones	15 HORAS
Unidad 11 – Empaquetar aplicaciones	15 HORAS
Unidad 12 – Serialización (I)	15 HORAS
Unidad 13 – Threads	15 HORAS
Unidad 14 – Serialización (II)	7,5 HORAS
Unidad 15 – Acceso a bases de datos	7,5 HORAS
<b>TOTAL</b>	<b>150</b>