



Técnicas de  
Programación  
Avanzada  
**Grado en Ingeniería  
Informática**



UNIVERSIDAD  
**NEBRIJA**

## GUÍA DOCENTE

**Asignatura:** Técnicas de Programación Avanzada

**Titulación:** Grado en Ingeniería Informática

**Carácter:** Obligatoria

**Idioma:** Castellano.

**Modalidad:** Presencial

**Créditos:** 6

**Curso:** 2º

**Semestre:** 1º

**Profesores/Equipo Docente:** Dra. Nieves Cubo Mateo

### 1. COMPETENCIAS Y RESULTADOS DE APRENDIZAJE

#### 1.1. Competencias

La asignatura contribuye a adquirir las siguientes competencias:

*Competencias específicas:*

CEB04. Aplicar los conocimientos básicos sobre el uso y programación de los ordenadores, sistemas operativos, bases de datos y programas informáticos con aplicación en ingeniería.

CEB05. Aplicar el conocimiento de la estructura, organización, funcionamiento e interconexión de los sistemas informáticos, los fundamentos de su programación, y su aplicación para la resolución de problemas propios de la ingeniería.

CEC01. 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.

CEC05. Conocer, administrar y mantener sistemas, servicios y aplicaciones informáticas.

CEC06. Conocer y aplicar los procedimientos algorítmicos básicos de las tecnologías informáticas para diseñar soluciones a problemas, analizando la idoneidad y complejidad de los algoritmos propuestos.

CEC07. Conocer, diseñar y utilizar de forma eficiente los tipos y estructuras de datos más adecuados a la resolución de un problema.

CEC08. Analizar, diseñar, construir y mantener aplicaciones de forma robusta, segura y eficiente, eligiendo el paradigma y los lenguajes de programación más adecuados.

CEC14. Conocer y aplicar los principios fundamentales y técnicas básicas de la programación paralela, concurrente, distribuida y de tiempo real.

CEC17. Diseñar y evaluar interfaces persona computador que garanticen la accesibilidad y usabilidad a los sistemas, servicios y aplicaciones informáticas.

*Competencias generales:*

CGT1. Analizar y sintetizar la información necesaria para realizar su trabajo plasmando los resultados en informes o en la toma de decisiones en proyectos del ámbito de la ingeniería informática.

CGT2. Organizar y planificar los recursos e ideas necesarias para realizar su trabajo ideando acciones e hitos en proyectos del ámbito de la ingeniería informática.

CGT3. Comunicar de forma oral y escrita en la lengua nativa pudiendo expresar sus opiniones de forma clara para transmitir conceptos y soluciones dentro del ámbito de la ingeniería informática.

CGT5. Aplicar conocimientos de informática relativos al ámbito de estudio al trabajar en proyectos del ámbito de la ingeniería informática.

CGT6. Gestionar la información, conociendo su importancia y la forma de procesarla generando los recursos necesarios para facilitar su acceso y provisión en el ámbito de la ingeniería informática.

CGT7. Resolver problemas en el entorno de trabajo, dentro del ámbito de la ingeniería informática, enfrentándose a situaciones complejas en cuanto a problemas técnicos y a las relaciones personales y profesionales.

CGT8. Tomar decisiones basadas en la información disponible en el ámbito de la ingeniería informática.

CGS1. Aplicar la motivación por la calidad en el desarrollo de su actividad profesional poniendo el máximo cuidado en el desarrollo de los entregables dentro de proyectos de ingeniería informática.

CGS2. Razonar de forma crítica ante los problemas que surjan en el ámbito de la ingeniería informática, contando con la información disponible, y explicar dicho razonamiento.

CGS4. Aprender de forma autónoma conceptos relativos a la profesión ingenieril para facilitar la mejora continua ya sea mediante el acceso a información disponible o cualquier otro medio.

CGS5. Adaptarse a nuevas situaciones en el entorno de la ingeniería informática, reconociendo dichas situaciones y expresando formas de afrontarlas.

CGS6. Aplicar la creatividad ante las diferentes circunstancias generando soluciones novedosas dentro del ámbito de la ingeniería informática.

CGP1. Trabajar en equipo contribuyendo de forma activa al resultado de los proyectos u operaciones del ámbito de la ingeniería informática.

CGP2. Trabajar en un equipo de carácter interdisciplinar dentro de proyectos del ámbito de la ingeniería informática, entendiendo y pudiendo explicar la división de trabajo y la integración de los diferentes miembros del mismo.

## 1.2. Resultados de aprendizaje

La asignatura contribuye a los siguientes resultados de aprendizaje:

- Aplicar procedimientos algorítmicos básicos de las tecnologías informáticas.
- Diseñar soluciones a problemas, analizando la idoneidad y complejidad de los algoritmos propuestos.
- Determinar los tipos y estructuras de datos avanzadas más adecuados a la resolución de un problema.
- Emplear métodos de programación imperativa y de programación orientada a objetos para desarrollar soluciones a problemas concretos.
- Interpretar código realizado por terceros para su reutilización.
- Organizar la arquitectura de una aplicación a partir del uso de patrones de diseño.

- Evaluar diferentes diseños de aplicaciones para seleccionar el más apropiado para resolver un problema.

## 2. CONTENIDOS

### 2.1. Requisitos previos

Es conveniente, aunque no imprescindible haber aprobado las asignaturas de Programación I y II.

### 2.2. Descripción de los contenidos

Objetos y memoria. Polimorfismo. Programación basada en eventos e inversión de control. Interfaces gráficas de usuario. Entrada / salida. Genericidad y plantillas. Excepciones. Threads. Patrones de diseño.

### 2.3. Contenido detallado

Presentación de la asignatura y explicación de la Guía Docente.

#### **Tema 1: Objetos y memoria.**

- 1.1 Características básicas del lenguaje. Primer programa. Compilación y Ejecución. IDE.
- 1.2 Sentencias de control. Secuencia, selección e iteración.
- 1.3 Abstracción. Clases, objetos, métodos y atributos.
- 1.4 Sobrecarga de métodos y encapsulamiento.

#### **Tema 2. Otros conceptos fundamentales de la Programación Orientada a Objetos**

- 2.1 Herencia. Interfaces y clases abstractas. Agregación.
- 2.2 Polimorfismo.
- 2.3 Gestión de Excepciones.
- 2.4 Genericidad y plantillas.
- 2.5 Utilidades. Entrada y Salida.
- 2.6 Anotaciones.

#### **Tema 3. Patrones de Diseño.**

- 3.1 Concepto de Patrones de Diseño
- 3.2 Patrones de creación
- 3.3 Patrones estructurales.
- 3.4 Patrones de comportamiento.

#### **Tema 4. Programación de Interfaces.**

- 4.1 Interfaces Gráficas de Usuario.
- 4.2 Gestión de eventos.

#### **Tema 5. Temas Avanzados.**

- 5.1 Concurrencia.
- 5.2 Inversión de Control. Definición y ejemplos. Inyección de dependencias.
- 5.3 Expresiones avanzadas del lenguaje.

### 2.4. Actividades dirigidas

Como se indica en el contenido, durante el curso se desarrollarán actividades dirigidas en forma de 5 prácticas que corresponderán a:

- Práctica 1: Objetos y memoria
- Práctica 2: Herencia, polimorfismo
- Práctica 3: Patrones de diseño
- Práctica 4: Programación de interfaces

Práctica 5: Concurrencia e inversión del control

El contenido de las prácticas podrá modificarse con el fin de afianzar aquellos aspectos para los que se detecte una mayor dificultad de aprendizaje.

**2.5. Actividades formativas**

Código	Actividades formativas	Descripción
AF1	Clases de teoría, evaluación y problemas	Las clases de teoría utilizan la metodología de Lección Magistral que se desarrollará en el aula empleando la pizarra y/o el cañón de proyección. Las clases de problemas se podrán impartir en aula informática utilizando la pizarra y/o el ordenador. En función de la asignatura se dará un mayor peso a unas u otras.
AF2	Tutorías	Consulta al profesor por parte de los alumnos sobre la materia en los horarios de tutorías o empleando mecanismos de tutoría telemática (correo electrónico, uso del campus virtual de la Universidad o herramientas de telepresencialidad como Blackboard Collaborate)
AF3	Prácticas	Se desarrollarán en un aula informática o en el laboratorio correspondiente, con ordenadores para todos los alumnos y los materiales apropiados. El profesor enseñará a los alumnos a utilizar programas informáticos o herramientas electrónicas para la asignatura indicada en cada caso. Los alumnos realizarán las prácticas aplicando los conocimientos adquiridos en las clases de teoría y problemas, ayudándoles a afianzarlos.
AF4	Estudio individual	Trabajo individual del alumno utilizando los apuntes de clase, libros de la biblioteca, o apuntes del profesor disponibles en el campus virtual. Se le encargará al alumno la realización y entrega de trabajos individuales o en grupo. Algunos de ellos se expondrán oralmente a lo largo del curso por parte de los alumnos, lo que facilitará alcanzar la competencia comunicativa en mayor grado. Algunos trabajos requerirán el manejo de programas informáticos que estarán disponibles en los ordenadores de la universidad. Otros requerirán un trabajo de investigación sobre los contenidos de la materia o similares y aplicaciones.

CÓDIGO	ACTIVIDAD FORMATIVA	HORAS	PORCENTAJE DE PRESENCIALIDAD
AF1	Clases de teoría, evaluación y problemas	45	100
AF2	Tutorías	12,5	100
AF3	Prácticas	15	100
AF4	Estudio individual	77,5	0

**3. SISTEMA DE EVALUACIÓN**

**3.1. Sistema de calificaciones**

El sistema de calificaciones finales se expresará numéricamente del siguiente modo:

- 0 - 4,9 Suspenso (SS)
- 5,0 - 6,9 Aprobado (AP)

7,0 - 8,9 Notable (NT)  
9,0 - 10 Sobresaliente (SB)

La mención de "matrícula de honor" podrá ser otorgada a alumnos que hayan obtenido una calificación igual o superior a 9,0.

### 3.2. Criterios de evaluación

#### Convocatoria ordinaria

Sistemas de evaluación	Porcentaje
Evaluación de la participación del alumno	5%
Actividades dirigidas, prácticas y memorias de prácticas, trabajos (obligatorios y voluntarios) y proyectos a realizar.	15%
Prueba escrita parcial	15%
Prueba escrita final	65%

#### Convocatoria extraordinaria

Sistemas de evaluación	Porcentaje
Actividades dirigidas, prácticas y memorias de prácticas, trabajos (obligatorios y voluntarios) y proyectos a realizar.	10%
Prueba escrita final	90%

### 3.3. Restricciones

#### Calificación mínima

La ponderación tanto del examen parcial como de los conceptos de participación y trabajos escritos/prácticas, solo se aplicará si el alumno obtiene al menos un 5 en el examen final. Esta ponderación también se aplica solo en el caso de que el alumno obtenga al menos un 5 en este examen final extraordinario.

La no superación de las prácticas supone el suspenso automático de la asignatura en la convocatoria ordinaria y extraordinaria. Se conservará la nota de prácticas aprobadas para posteriores convocatorias.

Las prácticas que no hayan sido aprobadas pueden, en su caso, ser entregadas de nuevo para ser evaluadas en la convocatoria extraordinaria, previa consulta al profesor y siempre antes del examen de la convocatoria ordinaria.

El examen parcial no libera materia.

#### Asistencia

El alumno que, injustificadamente, deje de asistir a más de un 25% de las clases presenciales podrá verse privado del derecho a examinarse en la convocatoria ordinaria.

#### Normas de escritura

Se prestará especial atención en los trabajos, prácticas y proyectos escritos, así como en los exámenes tanto a la presentación como al contenido, cuidando los aspectos gramaticales y ortográficos. El no cumplimiento de los mínimos aceptables puede ocasionar que se resten puntos en dicho trabajo.

### 3.4. Advertencia sobre plagio

La Universidad Antonio de Nebrija no tolerará en ningún caso el plagio o copia. Se considerará plagio la reproducción de párrafos a partir de textos de auditoría distinta a la del estudiante (Internet, libros, artículos, trabajos de compañeros...), cuando no se cite la fuente original de la que provienen. El uso de las citas no puede ser indiscriminado. El plagio es un delito.

En caso de detectarse este tipo de prácticas, se considerará Falta Grave y se podrá aplicar la sanción prevista en el Reglamento del Alumno.

#### 4. BIBLIOGRAFÍA

##### Bibliografía básica

- B. Eckel. *Piensa en Java*. Pearson. 4ª edición. 2007.
- P. Deitel and H. Deitel. *Java How to Program*. Early objects. Pearson. 11th Edition. 2018.
- E. Gamma, R. Helm, R. Johnson and J. Vlissides. *Patrones de diseño*. Pearson 2003.

##### Bibliografía recomendada

- The Java tutorials. <https://docs.oracle.com/javase/tutorial/>
- P. Deck. *Spring MVC: A Tutorial*. Brainy Software. 2nd Edition. 2016