



Guía docente de la asignatura

## Nuevas Tecnologías de la Programación (Especialidad Ingeniería del Software) (29611BD)

Fecha de aprobación: 14/06/2022

<b>Grado</b>	Grado en Ingeniería Informática	<b>Rama</b>	Ingeniería y Arquitectura				
<b>Módulo</b>	Complementos de Ingeniería del Software	<b>Materia</b>	Complementos de Programación				
<b>Curso</b>	4 <sup>o</sup>	<b>Semestre</b>	2 <sup>o</sup>	<b>Créditos</b>	6	<b>Tipo</b>	Optativa

### PRERREQUISITOS Y/O RECOMENDACIONES

Los alumnos no habrán de tener materias o asignaturas aprobadas como requisito indispensable para superar esta materia. No obstante, se recomienda tener aprobados los contenidos y adquiridas las competencias de cuatrimestres precedentes.

### BREVE DESCRIPCIÓN DE CONTENIDOS (Según memoria de verificación del Grado)

- Técnicas avanzadas de programación: programación funcional, metadatos, reflexión.
- Tipos genéricos
- Patrones y antipatrones de diseño.
- Aplicaciones.

### COMPETENCIAS ASOCIADAS A MATERIA/ASIGNATURA

#### COMPETENCIAS GENERALES

- CG08 - 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.
- CG09 - 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.

### RESULTADOS DE APRENDIZAJE (Objetivos)

- Conocer el paradigma de programación funcional.



- Conocer la forma en que se incorpora este lenguaje de programación en el lenguaje Java.
- Abordar la solución de un programa usando programación funcional con el lenguaje de programación Scala.
- Analizar de forma crítica un diseño software, identificando puntos esenciales que supongan un problema en cuanto a su posible ampliación o modificación.
- Conocer soluciones prototípicas que han demostrado su eficacia en la resolución de problemas concretos pero usualmente presentes en los sistemas software.
- Ser capaces de identificar qué patrón de diseño puede solucionar un determinado problema identificado en un diseño software.
- Adquirir capacidades en el uso avanzado de lenguajes de programación, no cubiertas en otras materias y asignaturas, como, por ejemplo, el paradigma de programación funcional.
- Ser capaces de identificar y analizar problemas y diseñar, desarrollar, implementar, verificar y documentar soluciones software sobre la base de un conocimiento adecuado de las teorías, modelos y técnicas actuales.

## PROGRAMA DE CONTENIDOS TEÓRICOS Y PRÁCTICOS

### TEÓRICO

Tema 1. Repaso de conceptos de orientación a objetos.

1. Conceptos de clase y objeto, herencia, polimorfismo, uso de interfaces.

Tema 2. Programación funcional en Java

1. Introducción.
2. Uso de colecciones.
3. String, comparadores y filtros.
4. Diseño con expresiones lambda.
5. Trabajo con flujos.
6. Evaluación lazy.
7. Optimización de recursos.
8. Composición con expresiones lambda.

Tema 3. Programación en Scala: paradigma de programación funcional

1. Introducción.
2. Variables, estructuras de control
3. Funciones
4. Clases y objetos, herencia.

Tema 4. Patrones y antipatrones. Refactorización.

1. Introducción.
2. Ejemplos motivadores y aplicaciones de patrones: observador, decorador, factoría, factoría abstracta, singleton, plantilla, iterador, composite, estado.
3. Introducción a la refactorización.

### PRÁCTICO



- Práctica 1. Entorno de trabajo, repaso de conceptos previos de orientación a objetos. Práctica no evaluable.
- Práctica 2. Práctica sobre programación funcional en Java.
- Práctica 3. Práctica de Scala: funciones y recursividad.
- Práctica 4. Trabajo de evaluación de la asignatura. Práctica de diseño e implementación en Scala.

## BIBLIOGRAFÍA

### BIBLIOGRAFÍA FUNDAMENTAL

- Introduction to Java Programming and Data Structures. Y.D. Liang. Pearson, 2019.
- Big Java: Early objects. C. Horstman. Wiley, 2019.
- Programming in Scala, M. Odersky, L. Spoon, B. Venners. Artima, 2019.
- Programming Scala: Scalability = functional programming + objects. D. Wampler, A. Payne. O'Reilly Media, 2014.
- Introduction to the art of programming using Scala. M. C. Lewis. Chapman & Hall, 2012.
- Learning Scala: practical functional programming for the JVM. J. Swartz. O'Reilly Media, 2014.
- Get programming with Scala. D. Sfregola. Manning, 2021.

### BIBLIOGRAFÍA COMPLEMENTARIA

- Functional Programming in Java: Harnessing the power of Java 8 lambda expressions. V. Subramaniam. The pragmatic programmers, 2014.
- Object orientation, abstraction and data structures using Scala. M.C. Lewis, L.L. Lacher. Chapman & Hall, 2017.
- Head First Design Patterns: a brain-friendly guide. E. Freeman, E. Freeman, B. Bates, K. Sierra. O'Reilly Media, 2020.
- Design Patterns Explained. A. Shalloway, J.R. Trott. Addison-Wesley, 2004.

## ENLACES RECOMENDADOS

- [Programación con Scala](#)
- [Libro disponible sobre Scala](#)
- [Patrones de diseño](#)
- [Patrones de diseño con Java](#)
- [Patrones de diseño, antipatrones, refactorización](#)
- [Java programming design patterns](#)

## METODOLOGÍA DOCENTE

- MD01 - Lección Magistral (Clases Teóricas-Expositivas)
- MD02 - Actividades Prácticas (Resolución de Problemas, Resolución de Casos Prácticos, Desarrollo de Proyectos, Prácticas en Laboratorio, Taller de Programación, Aula de Informática, Prácticas de Campo).



**EVALUACIÓN (instrumentos de evaluación, criterios de evaluación y porcentaje sobre la calificación final)****EVALUACIÓN ORDINARIA**

## Porcentaje de evaluación

Evaluación de la asignatura	
Entregas de prácticas sin defensa	55%
Entrega y defensa de práctica final	35%
Evaluación continua	10%

Es obligatoria la realización de todas las prácticas propuestas en la asignatura. En caso de no entregar y defender la práctica final, se suspenderá la asignatura en la evaluación ordinaria, pero se podrá mantener la nota de prácticas para la evaluación extraordinaria.

**EVALUACIÓN EXTRAORDINARIA**

Los estudiantes serán evaluados con un examen único consistente en la resolución de varios cuestiones teórico-prácticas relacionadas con la materia impartida. El peso de este examen dependerá de las actividades realizadas por el alumno durante la fase de evaluación continua: 100% en caso de no haber realizado ninguna entrega; 35 % en caso de haber entregado las prácticas sin defensa.

**EVALUACIÓN ÚNICA FINAL**

De acuerdo a lo establecido en la Normativa de evaluación y de calificación de los estudiantes de la Universidad de Granada vigente, la evaluación será preferentemente continua. No obstante, el estudiante que no pueda acogerse a dicho sistema por motivos laborales, estado de salud, discapacidad, programas de movilidad o cualquier otra causa debidamente justificada podrá acogerse a la evaluación única final. Para ello deberá solicitarlo al Director del Departamento o al Coordinador del Máster en las dos primeras semanas de impartición de la asignatura o, excepcionalmente, en las dos primeras semanas tras la matriculación en la asignatura.

Esta modalidad de evaluación se realizará en un único acto académico en la fecha establecida por el Centro y consistirá en un examen escrito (evaluado de 0 a 10) que incluirá preguntas tanto de tipo teórico como práctico que garanticen que el alumno ha adquirido la totalidad de las competencias descritas en esta misma guía docente.

**INFORMACIÓN ADICIONAL**

La asistencia a las clases teóricas y prácticas no será obligatoria, aunque la participación activa en clase y la entrega de ejercicios planteados por el profesor se tendrá en cuenta dentro del sistema de evaluación continua de la asignatura. Sí será obligatoria la defensa de prácticas si así se requiere por parte del profesor (en caso de no realizarse la defensa la práctica correspondiente se considerará como no entregada).

