

**Proyecto/Guía docente de la asignatura**

Se debe indicar de forma fiel cómo va a ser desarrollada la docencia. Esta guía debe ser elaborada teniendo en cuenta a todos los profesores de la asignatura. Conocidos los espacios y profesorado disponible, se debe buscar la máxima presencialidad posible del estudiante siempre respetando las capacidades de los espacios asignados por el centro y justificando cualquier adaptación que se realice respecto a la memoria de verificación. Si la docencia de alguna asignatura fuese en parte online, deben respetarse los horarios tanto de clase como de tutorías). La planificación académica podrá sufrir modificaciones de acuerdo con la actualización de las condiciones sanitarias.

Asignatura	Lenguajes de Programación Multiparadigma		
Materia	Plataformas Tecnológicas		
Módulo			
Titulación	Grado en Ingeniería de Servicios y Aplicaciones		
Plan	413	Código	40826
Periodo de impartición	Semestre 7	Tipo/Carácter	OB
Nivel/Ciclo	Grado	Curso	4
Créditos ECTS	6		
Lengua en que se imparte	Español		
Profesor/es responsable/s	José Vicente Álvarez Bravo		
Datos de contacto (E-mail, teléfono...)	Escuela de Ingeniería Informática de Segovia Plaza de la Universidad, nº 1, 40.005 - Segovia Teléfono: (+34) 921 11 24 54 Fax: (+34) 921 11 24 01 email: jvalvarez@infor.uva.es		
Departamento	Informática (ATC; CCIA, LSI)		



1. Situación / Sentido de la Asignatura

Esta asignatura, que se encuadra dentro de la materia “Plataformas Tecnológicas”, pretende mostrar todos los aspectos más relevantes de aquellas nuevas plataformas que empiezan a despuntar y que pueden aportar una visión más aproximada de lo que es actualidad en el ámbito de las tecnologías de la Información.

1.1 Contextualización

En la actualidad, la mayoría del desarrollo de software se realiza desde el paradigma Orientado a Objetos. Sin embargo, cada vez más se imponen nuevos entornos multiparadigma, es decir, que integran varios paradigmas en una única plataforma de desarrollo. Éste es el caso de Scala, un lenguaje orientado a objetos pero que también sigue la filosofía de la programación funcional. La idea es extraer lo mejor de estos dos paradigmas e integrarlo en un único lenguaje.

1.2 Relación con otras materias

Esta asignatura se relaciona directamente con las asignaturas de la materia en la que se engloba. También tiene conexión con todas aquellas en las que se abordan la adquisición de habilidades en técnicas de programación.

1.3 Prerrequisitos

No se establecen prerrequisitos.

2. Competencias

2.1 Generales

- Competencias genéricas G01, G02, G03, G04, G05, G07, G08, G09, G10, G11, G12, G16, G18, G19, G20 y G21.

2.2 Específicas

- Competencias comunes a la rama de informática: E07, E18, E19.
- Competencias de Tecnologías de la Información: E26, E27, E29

3. Objetivos

- Introducir los conceptos básicos de la Programación funcional empleando Scala como lenguaje de programación.
- Hacer un uso generalizado de funciones de alto orden y polimorfismo para la obtención de programas modulares y fácilmente reutilizables.
- Definición y uso de la evaluación perezosa.
- Programación concurrente en Scala.
- Integración de todos estos conceptos en una plataforma multiparadigma como Scala.

4. Contenidos y/o bloques temáticos

Bloque Único	Contenidos de la asignatura
	Tema 1.- Programación Funcional. Conceptos Fundamentales
	Tema 2.- Introducción al lenguaje de programación Scala
	Tema 3.- Sintáxis y Elementos básicos de Scala
	Tema 4.- Colecciones de Datos en Scala
	Tema 5.- Funciones en Scala
	Tema 6.- Clases y Objetos en Scala
	Tema 7.- Genericidad en Scala
	Tema 8.- Mónadas y monoides
	Tema 9.- Parámetros implícitos y conversiones

Carga de trabajo en créditos ECTS: 6



a. Contextualización y justificación

Los alumnos ya conocen el paradigma de la programación orientada a objetos y su uso en distintos entornos de desarrollo. En la asignatura de LPM vamos a introducir las bases del paradigma de Programación Funcional con la idea de que finalmente se integren con el paradigma anterior a través del lenguaje Scala. En ninguna asignatura anterior del plan de estudios se ha trabajado con este lenguaje, por lo que resulta imprescindible ofrecer a los alumnos una primera aproximación al mismo.

b. Objetivos de aprendizaje

- ✓ Conocer los conceptos básicos asociados al paradigma de Programación Funcional.
- ✓ Conocer las características principales del lenguaje de programación Scala.
- ✓ Conocer cómo se integra este paradigma con el Orientado a Objetos dentro de esta plataforma.
- ✓ Trabajar en el laboratorio con un IDE de este lenguaje.

c. Contenidos

En los temas 1 y 2 se proporciona un marco teórico que permita tener una visión general del paradigma de programación funcional y del lenguaje Scala, con la intención de que el alumno pueda comprender mejor los conceptos que se van a desarrollar en los temas siguientes. A continuación, en el tema 3 se presentan la sintaxis y los conceptos básicos asociados al Lenguaje de programación Scala. En el tema 4, se introducen las Colecciones de Datos más relevantes del lenguaje Scala. En el tema 5 se aborda la definición de funciones como objetos que pueden ser argumento o resultado de otra función (First-Class Data values). En este sentido se introducirán los conceptos de función de orden superior, de función anónima, de función parcial, el proceso de Currificación o el concepto de Cierre (Closure). En el tema 6 se introduce la parte orientada a objetos específica de Scala: las clases Case y la implementación de Traits como una forma de extensión funcional. En el tema 7 se describe como se aborda la genericidad desde esta plataforma, así como las colecciones de datos más comunes. En el tema 8 se introducen los monoides y las mónadas como una poderosa herramienta para crear abstracciones de alto nivel. Por último, en el tema 9 se aborda la definición de parámetros implícitos y la forma que tiene Scala de hacer conversiones entre tipos.

d. Métodos docentes

1. Lección magistral: exposición de teoría
2. Prácticas en aula: resolución de problemas
3. Evaluación
4. Estudio autónomo por parte del alumno, incluyendo la realización de problemas, consulta bibliográfica, realización de prácticas y preparación de pruebas de evaluación

e. Plan de trabajo

- Alternar sesiones teóricas con clases de problemas y prácticas de laboratorio

f. Evaluación

Ver tabla apartado 7.

g. Bibliografía básica

- S. Thompson. The craft of Functional programming, Editorial Addison-Wesley, 3ª Edición, 2011.
- P. Chiusano & R. Bjarnason. Functional Programming in Scala. Editorial Manning. 2015
- C. S. Horstmann. Scala for the Impatient. Editorial Addison-Wesley. 2015.
- Learning Scala, Practical functional Programming for the JVM. Jason Swartz. O'Reilly (online book).

- Reactive programming with Scala and Akka. Prassanna Kumar & Suraj Atreya. Packt 2016 (Open source - online book).

h. Bibliografía complementaria

- The Scala Language Especification (V2.9). Martin Odersky. 2014.
<http://www.scala-lang.org/docu/files/ScalaReference.pdf>
- Scala by Example. Martin Odersky. 2014.
<http://www.scala-lang.org/docu/files/ScalaByExample.pdf>
- Akka documentation 2.4.
<http://doc.akka.io/docs/akka/2.4.7/AkkaScala.pdf>

i. Recursos necesarios

Aula con pizarra y ordenador con proyector, biblioteca, sala de estudio y despacho para tutorías.

j. Temporalización

CONTENIDOS Teórico/Prácticos	CARGA ECTS	PERIODO PREVISTO DE DESARROLLO
TEMA 1	0,2	Semana: 1
TEMA 2	0,2	
TEMA 3	0,8	Semanas:2-3
TEMA 4	0,8	Semanas: 4-5
TEMA 5	1	Semanas: 6-8
TEMA 6	1	Semanas: 8-10
TEMA 7	0,8	Semanas: 11-12
TEMA 8	0,8	Semanas: 13
TEMA 9	0,4	Semana: 14

5. Métodos docentes y principios metodológicos

Dada la naturaleza de la asignatura, del número de alumnos que se han matriculado en estos últimos años (menor o igual que 15), y siguiendo el principio de presencialidad segura, se ha optado preferentemente por seguir una metodología 100% presencial. Sin embargo, se deja abierta la posibilidad a cualquiera de las otras metodologías bimodales propuestas por la universidad, en el caso de que las circunstancias así lo aconsejen.

6. Tabla de dedicación del estudiante a la asignatura

ACTIVIDADES PRESENCIALES	HORAS	ACTIVIDADES NO PRESENCIALES	HORAS
Clases teórico-prácticas	15	Estudio y trabajo autónomo individual	40
Clases prácticas de aula	15	Estudio y trabajo autónomo grupal	50
Laboratorios	30		
Total presencial	60	Total no presencial	90



7. Sistema y características de la evaluación

INSTRUMENTO/PROCEDIMIENTO	PESO EN LA NOTA FINAL	OBSERVACIONES
Resolución de ejercicios prácticos en el aula – laboratorio.	100%	Resolución de ejercicios prácticos en el laboratorio que se deberán entregar para su evaluación. La calificación final en esta parte será la media ponderada de todos los ejercicios realizados.

CRITERIOS DE CALIFICACIÓN

- **Convocatoria ordinaria:**
 - Para superar la convocatoria ordinaria se deberá obtener como mínimo un 5 sobre 10 en la nota ponderada de todos los ejercicios realizados.
- **Convocatoria extraordinaria:**
 - Los alumnos que no superen la convocatoria ordinaria harán un examen práctico delante del ordenador en el que se pedirá que realicen un conjunto de ejercicios similares a los realizados durante las sesiones de laboratorio. La nota mínima para aprobar será de 5 puntos sobre 10

8. Consideraciones finales

Todos los recursos docentes de la asignatura, bibliografía incluida, estarán disponibles en el espacio dedicado a la asignatura en el campus virtual de la Uva: campusvirtual.uva.es.