



Tecnología y Paradigmas de Programación

1. Identificación de la asignatura

NOMBRE	Tecnología y Paradigmas de Programación		CÓDIGO	GIISOF01-2-004
TITULACIÓN	Grado en Ingeniería Informática del Software	CENTRO	Escuela de Ingeniería Informática	
TIPO	Obligatoria	Nº TOTAL DE CRÉDITOS	6 ECTS	
PERIODO	Semestral	IDIOMA	Bilingüe (Castellano e Inglés)	
COORDINADOR		TELÉFONO /EMAIL		UBICACIÓN
José Quiroga Álvarez		3375 / quirogajose@uniovi.es		Edif. De Ciencias - Desp. 189 - Campus de Llamaquique
PROFESORADO		TELÉFONO /EMAIL		UBICACIÓN
José Otero Rodríguez		2500 / jotero@uniovi.es		Despacho 1..15, sedes departamentales oeste - Gijón
Miguel García Rodríguez		4134 / garciaRmiguel@uniovi.es		Edif. De Ciencias - Desp. 203 - Campus de Llamaquique
Miguel Sánchez Santillán		985 103156 / sanchezsmiguel@uniovi.es		Facultad de Geología - Desp. 5-1
Javier Escalada Gómez		3153 / escaladajavier@uniovi.es		Facultad de Geología - Desp. 3-23

2. Contextualización (en el caso de asignaturas compartidas se contextualizará, si existen diferencias, para cada una de las titulaciones donde se comparte).

En el desarrollo de aplicaciones informáticas, los lenguajes de programación han sufrido numerosas evoluciones y se han definido distintos paradigmas de programación en base a las abstracciones principales que los lenguajes utilizan para permitir la programación de ordenadores. El conocimiento de los paradigmas y elementos ofrecidos por los lenguajes de programación es importante para saber elegir las tecnologías de programación adecuadas a la hora de resolver un problema particular. Esta asignatura afronta este reto, describiendo los paradigmas de programación más utilizados en la actualidad y los elementos principales de cada uno de ellos, para que el alumno sea capaz de elegir el más apropiado en la resolución de un problema.

La materia en la que se enmarca esta asignatura dentro del Grado en Ingeniería Informática del Software es Programación, siendo una asignatura Obligatoria. Su carácter es mayoritariamente práctico, teniendo como principal objetivo hacer que el alumno sea capaz de resolver problemas



reales planteados mediante la utilización de elementos de lenguajes de programación y paradigmas adecuados.

Se imparte como una asignatura de segundo curso, dentro del segundo semestre. Cuenta con 60 créditos ECTS, que suponen un total de 150 horas de trabajo: 60 horas presenciales y 90 horas no presenciales. Se impartirán 2 horas de clases expositivas a la semana hasta un total de 21; una hora de seminario hasta un total de 7; 2 horas de prácticas de laboratorio cada semana y 2 horas totales de tutorías grupales.

Dentro de las competencias a adquirir relacionadas con el desarrollo de software, es una continuación de las asignaturas Introducción a la Programación y Metodología de la Programación. Sirve como complemento a asignaturas relacionadas como Estructuras de Datos, Algoritmia y Sistemas Operativos y como base para asignaturas como Diseño del Software, Arquitectura del Software y Diseño de Lenguajes de Programación.

3. Requisitos

Los requisitos para cursar esta asignatura son:

- Tener conocimientos de programación imperativa basada en procedimientos
- Destreza en el desarrollo de programas en un lenguaje imperativo basado en procedimientos, preferentemente el lenguaje C
- Conocimiento del paradigma orientado a objetos
- Experiencia en el desarrollo de programas siguiendo el paradigma orientado a objetos, preferiblemente en Java
- Nociones básicas de programación concurrente

Expresando los requisitos como asignaturas del grado en Ingeniería Informática del Software, el alumno deberá haber cursado y superado las siguientes asignaturas del mismo:

- Fundamentos de Informática
- Introducción a la Programación
- Metodología de la Programación

4. Competencias y resultados de aprendizaje

Se indican las competencias de la titulación con las que la presente asignatura tiene alguna relación. Cuando no cubre la totalidad de la misma, se indica la parte de ésta que la asignatura aporta en el currículum del alumno.

Competencias técnicas o específicas:

Com.8: Capacidad para analizar, diseñar, construir y mantener aplicaciones de forma robusta, segura y eficiente, eligiendo el paradigma y los lenguajes de programación más adecuados.

Com.14: Conocimiento y aplicación de los principios fundamentales y técnicas básicas de la programación paralela, concurrente, distribuida y de tiempo real.

De esa competencia, la propia a adquirir por el alumno en la presente asignatura es:



Conocimiento y aplicación de los principios fundamentales y técnicas básicas de la programación paralela y concurrente.

Bas.5: 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.

Com.1: Capacidad para diseñar y desarrollar aplicaciones y sistemas informáticos, asegurando su fiabilidad, seguridad y calidad, conforme a principios éticos y a la legislación y normativa vigente.

De esa competencia, la propia a adquirir por el alumno en la presente asignatura es:

Capacidad para diseñar y desarrollar aplicaciones, asegurando su fiabilidad, seguridad y calidad.

Com.16: Conocimiento y aplicación de los principios, metodologías y ciclos de vida de la ingeniería de software.

De esa competencia, la propia a adquirir por el alumno en la presente asignatura es:

Conocimiento de los principios de la ingeniería de software.

Competencias transversales:

CG-3: Capacidad de abstracción

CG-4: Análisis y síntesis

CG-17: Competencia para una actitud positiva ante las futuras nuevas tecnologías y situaciones

CG-10: Destreza en la comunicación oral y escrita en inglés.

Esta competencia sólo se aborda en la comunicación escrita en inglés, en particular en la lectura.

El alumno deberá ser capaz de (resultados de aprendizaje):

- **RA.P-2.** Conocer la estructura, organización y funcionamiento de los sistemas informáticos y sistemas inteligentes, los fundamentos de su programación, y su aplicación para la resolución de problemas propios de la ingeniería. [Bas.5].
- **RA.P-3.** Analizar, diseñar, desarrollar, seleccionar, evaluar y mantener aplicaciones y sistemas informáticos, asegurando su fiabilidad y calidad aplicando las teorías, principios, métodos y prácticas de la Ingeniería del Software, eligiendo el paradigma y los lenguajes de programación más adecuados, considerando las limitaciones derivadas del coste, del tiempo, de la existencia de sistemas ya desarrollados y de las propias organizaciones. [Com.1] [Com.8] [Com16] [CG-3] [CG-4].
- **RA.P-4.** Conocer y aplicar procedimientos algorítmicos básicos, tipos y estructuras de datos más adecuados a la resolución de un problema, analizando la idoneidad y complejidad de los mismos. [CG-3].
- **RA.P-5.** Conocer y aplicar los principios fundamentales y técnicas básicas de la programación paralela, concurrente, distribuida y de tiempo real. [Com.14].
- **RA.P-6.** Capacidad de dar solución a problemas de integración en función de las estrategias, estándares y tecnologías disponibles. [CG-4].
- **RA.P-7.** Documentar y exponer la solución a un problema a través de textos y diagramas, cumpliendo normas y estándares del diseño y desarrollo del software en español/inglés [CG-10].



- **RA.P-8.** Conocer las nuevas tecnologías empleadas por los lenguajes de programación, análisis de su tendencia y capacidad de juicio para el análisis de nuevas tecnologías y paradigmas [CG-17].
- **RA.SFH-5.** Comprender la influencia de la arquitectura en la programación, especialmente en arquitecturas paralelas [Com.14].
- **RA.SI-4.** Analizar las implicaciones del uso de hilos en una aplicación, comprendiendo las implicaciones en la compartición de recursos y utilizando los mecanismos básicos de comunicación y sincronización para solventar los posibles problemas a que dicha compartición pueda dar lugar. [Bas.4], [Com.14]
- **RA.IS-3.** Aplicar distintas técnicas de construcción de software en el diseño de bajo nivel [Com.8].

5. Contenido.

1. Lenguajes y paradigmas de programación

Lenguaje de Programación
Características de los Lenguajes de Programación
Paradigmas de Programación
Tecnología de Programación

2. Paradigma orientado a objetos

Abstracción
Encapsulamiento
Modularidad
Sobrecarga
Herencia
Polimorfismo
Clases Abstractas e Interfaces
Excepciones
Asertos
Diseño por Contratos
Genericidad
Inferencia de Tipos

3. Fundamentos del paradigma funcional

Cálculo Lambda
Isomorfismo Curry-Howard
Funciones como Entidades Primer Orden
Cláusulas
Currificación
Aplicación Parcial
Continuaciones
Evaluación Perezosa
Transparencia Referencial
Pattern Matching
Funciones de Orden Superior



Listas por comprensión

4. Fundamentos de la programación concurrente y paralela

Programación Concurrente y Paralela
Proceso e Hilo
Paralelización de Algoritmos
Paso Asíncrono de Mensajes
Creación Explícita de Hilos
Condición de Carrera
Context Switching & Thread Pooling
Foreground & Background Threads
Sincronización de Hilos y Procesos
Interbloqueo
Estructuras de Datos Thread-Safe
Paralelización mediante tareas y Task Parallel Library
Paradigma Funcional en la Paralelización de Algoritmos

5. Tipado Dinámico y Meta-programación

Tipado Dinámico
Duck Typing
Multiple Dispatch
Reflexión Computacional
Anotaciones o Atributos
Generación Dinámica de Código
MetaProgramación

A continuación, se detalla una tabla describiendo cómo las distintas competencias están relacionadas con el contenido de la asignatura:

Temas	Competencias Específicas	Competencias transversales
Tema 1	Com.8, Com.16, Bas.5	CG-10, CG-17
Tema 2	Com.8, Com.1, Com.16, Bas.5	CG-3, CG-4, CG-10, CG-17
Tema 3	Com.8, Com.14, Com.1, Bas.5	CG-3, CG-4, CG-10, CG-17
Tema 4	Com.8, Com.1, Bas.5	CG-3, CG-4, CG-10, CG-17
Tema 5	Com.8, Com.1, Bas.5	CG-3, CG-4, CG-10, CG-17

6. Metodología y plan de trabajo.

Las actividades y metodología a utilizar son:



- Clases expositivas dialogadas en las que se exponen las características de los lenguajes de programación en cada uno de los paradigmas descritos. El objetivo de estas clases es crear la base para que puedan, posteriormente, resolver los casos prácticos que se le planteen, eligiendo el paradigma y tecnología de programación adecuados (competencias Com.8, Com.14, Bas.5, Com.16 y CG-17).
- Actividades a realizar de forma autónoma por parte de alumno previamente a la asistencia a los laboratorios, para reforzar las competencias de las asignaturas previas a ésta (competencias Com.8, Bas.5, Com.1, CG-3, CG-4 y CG-10).
- Prácticas de aula y de laboratorio se plantearán problemas reales en los que se analizarán la adecuación de los elementos de cada paradigma para la resolución de los problemas planteados (competencias Com.8, Com.14, Bas.5, Com.1, CG-3 y CG-4)
- Deberá analizarse el material ofrecido en el que se resuelven problemas utilizando de forma apropiada los elementos de cada uno de los paradigmas (competencias Com.8, CG-3 y CG-4)
- El alumno deberá resolver problemas planteados a partir de lo aprendido, utilizando un lenguaje de programación y paradigma determinado (competencias Com.8, Com.14, Com.1, Bas.5, Com.16 y CG-4)
- Como trabajo autónomo, el alumno deberá ser capaz de leer material referenciado por el profesor, pudiendo está este escrito en el idioma inglés (competencias CG-4 y CG-10).

La siguiente tabla desglosa por temas y horas las actividades anteriormente descritas:

Temas	Horas totales	TRABAJO PRESENCIAL								TRABAJO NO PRESENCIAL	
		Clase Expositiva	Prácticas de aula /Seminarios/ Talleres	Prácticas de laboratorio /campo /aula de informática/ aula de idiomas	Prácticas clínicas hospitalarias	Tutorías grupales	Prácticas Externas	Sesiones de Evaluación	Total	Trabajo grupo	Trabajo autónomo
Tema 1	2	2							2		
Tema 2	52	6	2	8					16		36
Tema 3	42	5	2	10				1	18		24
Tema 4	46	6	3	10		2		1	22		24
Tema 5	8	2							2		6
Total	150	21	7	28		2		2	60		90



Lo siguiente es un resumen de horas en función del método utilizado:

MODALIDADES		Horas	%	Totales
Presencial	Clases Expositivas	21	14%	60
	Práctica de aula / Seminarios / Talleres	7	4,7%	
	Prácticas de laboratorio / campo / aula de informática / aula de idiomas	28	18,7%	
	Prácticas clínicas hospitalarias			
	Tutorías grupales	2	1,3%	
	Prácticas Externas			
	Sesiones de evaluación	2	1,3%	
No presencial	Trabajo en Grupo			90
	Trabajo Individual	90	60%	
Total		150	100%	

De forma excepcional, si las condiciones sanitarias lo requieren, se podrán incluir actividades de docencia no presencial. En cuyo caso, se informará al estudiantado de los cambios efectuados.



7. Evaluación del aprendizaje de los estudiantes.

La asignatura tendrá una evaluación de la parte más teórica y conceptual, en la que se evaluará el grado de adquisición de conceptos, abstracciones y habilidades propios de las competencias Com.8, Com.14, Com.1, Bas.5, Com.16 y CG-3. Para esta evaluación se realizará una prueba objetiva de respuesta corta al alumno.

Debido al carácter práctico de la asignatura, la parte con mayor peso en la evaluación será práctica, utilizando para ello un ordenador por alumno. Las competencias a evaluadas son Com.8, Com.14, Com.1, Bas.5, CG-3, CG-4, CG-17 y CG-10. Ésta consistirá en una serie de controles a realizar en las prácticas de laboratorio, consistentes en la ampliación o modificación del trabajo realizado por el alumno en las clases anteriores y de una forma autónoma.

Dependiendo de la convocatoria, la evaluación se realizará de forma distinta:

Convocatoria de Mayo

Esta convocatoria posee evaluación continua. La nota se compone de la calificación en el examen de teoría más la nota obtenida en los controles de prácticas. Para poder optar a esta calificación, se requiere un porcentaje de asistencia superior o igual al 80% de los seminarios y prácticas. En ese caso, el factor de ponderación será 40% para teoría y 60% para prácticas, siendo obligatorio haber obtenido una nota mínima de 4 puntos sobre 10 en teoría y una nota mínima de 3 puntos sobre 10 en cada uno de los exámenes de prácticas. Además, la nota ponderada de teoría y prácticas tendrá que ser de al menos 5 puntos sobre 10 para aprobar la asignatura. La nota será No Presentado cuando las pruebas de evaluación realizadas no superen el 50% de la evaluación total. Si el alumno no ha cumplido el porcentaje de asistencia requerido y su nota fuese superior o igual a 5, ésta pasará a ser 4.

Convocatorias de Enero y Julio

La evaluación se trata de un único examen teórico-práctico con el ordenador. Dicha prueba consistirá en preguntas de teoría y resolución de problemas con el ordenador.

Modelo de Evaluación Diferenciado

Para aquellos alumnos a los que se les conceda el sistema de evaluación diferenciado, no deberán satisfacer el porcentaje de asistencia del 80% para la convocatoria de mayo. El resto del sistema de evaluación se aplicará exactamente igual.

De forma excepcional, si las condiciones sanitarias lo requieren, se podrán incluir métodos de evaluación no presencial. En cuyo caso, se informará al estudiantado de los cambios efectuados.

8. Recursos, bibliografía y documentación complementaria.

Recursos Bibliográficos

- Oliver Sturm. Functional Programming in C#: Classic Programming Techniques for Modern Projects. Wiley. 2011.
- Tomas Petricek, Jon Skeet. Real World Functional Programming: With Examples in F# and C#. Manning Publications; 2009.
- Joseph Albahari. Threading in C#. Ebook disponible en <http://www.albahari.com/threading>



- Mark Michaelis. Essential C# 4.0 (3rd Edition). Addison-Wesley Professional; 3 edition. 2010.
- Andrew Troelsen. Pro C# 2010 and the .NET 4 Platform, Fifth Edition. Apress. 2010.

Recursos Software

- Microsoft Visual Studio