



GUÍA DOCENTE
**PROGRAMACIÓN ORIENTADA A
OBJETOS**

Coordinación: MARTÍNEZ RODRÍGUEZ, SANTIAGO

Año académico 2020-21

Información general de la asignatura

Denominación	PROGRAMACIÓN ORIENTADA A OBJETOS			
Código	102368			
Semestre de impartición	2o Q(SEMESTRE) EVALUACIÓN CONTINUADA			
Carácter	Grado/Máster	Curso	Carácter	Modalidad
	Grado en Técnicas de Interacción Digital y de Computación	1	TRONCAL	Presencial
Número de créditos de la asignatura (ECTS)	6			
Tipo de actividad, créditos y grupos	Tipo de actividad	PRALAB	TEORIA	
	Número de créditos	3	3	
	Número de grupos	1	1	
Coordinación	MARTÍNEZ RODRÍGUEZ, SANTIAGO			
Departamento/s	INFORMATICA E INGENIERIA INDUSTRIAL			
Distribución carga docente entre la clase presencial y el trabajo autónomo del estudiante	6 ECTS = 25x6 = 150 horas de trabajo: 40% -> 60 horas semipresenciales, 60% -> 90 horas trabajo autónomo del estudiante.			
Información importante sobre tratamiento de datos	Consulte este enlace para obtener más información.			
Idioma/es de impartición	Catalán.			
Distribución de créditos	Teoría virtual síncrona: 1.5 Teoría virtual asíncrona: 1.5 Prácticas: 3			

Profesor/a (es/as)	Dirección electrónica\nprofesor/a (es/as)	Créditos impartidos por el profesorado	Horario de tutoría/lugar
MARTÍNEZ RODRÍGUEZ, SANTIAGO	santi.martinez@udl.cat	6	

Información complementaria de la asignatura

Asumimos que los alumnos han adquirido los conceptos presentados en la asignatura Algorítmica y Programación, ya que, a partir de ellos, los extendemos en dos direcciones: la programación orientada a objetos y el diseño recursivo.

Objetivos académicos de la asignatura

Los objetivos de aprendizaje del estudiante son los siguientes:

- Aplicar el paradigma de orientación a objetos a problemas simples.
- Utilizar los tipos básicos de ficheros en Java.
- Diseñar algoritmos recursivos simples.
- Utilizar la documentación estándar de la plataforma Java.
- Manejar un entorno integrado de desarrollo.

Competencias

Competencias Básicas

- **B01.** Que los estudiantes hayan demostrado poseer y comprender conocimientos en su 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.

Competencias Transversales

- **CT3.** Adquirir capacitación en el uso de las nuevas tecnologías y de las tecnologías de la información y la comunicación.
- **CT5.** Adquirir nociones esenciales del pensamiento científico.

Competencias Generales

- **CG2.** Capacidad para diseñar, desarrollar, evaluar y asegurar la accesibilidad, ergonomía, usabilidad y seguridad de los sistemas informáticos.
- **CG3.** Capacidad para utilizar plataformas hardware y software adecuadas para el desarrollo y la ejecución de aplicaciones digitales interactivas.
- **CG5.** 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.
- **CG7.** Capacidad para resolver problemas con iniciativa, toma de decisiones, autonomía y creatividad.
- **CG8.** Capacidad para la abstracción y el razonamiento crítico, lógico y matemático.

Competencias Específicas

- **CE2.** Capacidad para comprender y dominar los conceptos básicos de matemática discreta, lógica, algorítmica y complejidad computacional, y su aplicación para la resolución de problemas propios de la computación.
- **CE3.** Conocimientos básicos sobre el uso y programación de los ordenadores, sistemas operativos y bases de datos útiles para el desarrollo de aplicaciones informáticas interactivas.
- **CE4.** Capacidad de conocer, comprender y evaluar la estructura y arquitectura de los computadores, así como los componentes básicos que los conforman.
- **CE16.** Capacidad para diseñar y evaluar interfaces persona computador que garanticen la usabilidad de los sistemas, servicios y aplicaciones informáticas.
- **CE17.** Saber aplicar los conocimientos de diseño suficientes para proponer y defender un concepto de diseño de un entorno interactivo y desarrollarlo hasta que pueda ser llevado a la práctica utilizando las tecnologías creativas adecuadas a cada proyecto.
- **CE24.** Ser capaz de comprender los factores humanos que intervienen en todo proceso de interacción entre personas y

tecnología, así como saber aplicarlos de forma adecuada al diseño de productos y servicios interactivos y sus interfaces.

Contenidos fundamentales de la asignatura

Tema 1. Introducción a Java

- 1.1 De C a Java
- 1.2 La biblioteca de la ACM Java Task Force
- 1.3 El programa principal
- 1.4 Uso de funciones auxiliares
- 1.5 Tratamiento de vectores (arrays)
- 1.6 Tratamiento de cadenas de caracteres

Tema 2. Programación Orientada a Objetos

- 2.1 Objetos y referencias
- 2.2 Introducción a las clases gráficas de las bibliotecas de la ACM
- 2.3 La clase String
- 2.4 Definición de clases en Java

Tema 3. Tratamiento de ficheros

- 3.1 Clases de ficheros
- 3.2 Ficheros secuenciales de texto
- 3.3 Ficheros binarios de acceso directo
- 3.4 Algoritmo de ordenación MergeSort

Tema 4. Recursividad

- 4.1 Funcionamiento de las llamadas a funciones
- 4.2 Pensar recursivamente
- 4.3 Recursividad usando índices
- 4.4 Búsqueda binaria
- 4.5 Recursividad múltiple

Ejes metodológicos de la asignatura

Cada semana el estudiante asiste a 2 horas virtuales (1 síncrona y 1 asíncrona) con Grupo Grande y 2 horas presenciales con Grupo Medio.

Las sesiones virtuales síncronas son videoconferencias, las asíncronas son videos. Las sesiones con Grupo Medio son de prácticas.

Grupo Grande: Clases Teoría (3 créditos)

- Parte teórica: clases soportadas por apuntes.
- Parte de aplicación práctica: se trabaja siempre sobre ejemplos.

Grupo Medio: Clases Prácticas (3 créditos)

- Clases dirigidas a la resolución de problemas por parte de los alumnos (se dispone de una colección de problemas y de exámenes de cursos anteriores).
- Resolución de dificultades y seguimiento personalizado de las prácticas.
- Uso de un entorno integrado de desarrollo.

Trabajo Autónomo (no presencial):

- Las prácticas se realizarán en horas no presenciales.
- Se recomienda que el alumno resuelva por su cuenta los problemas de la colección de problemas para practicar y obtener *feedback* por parte del profesor.

Plan de desarrollo de la asignatura

PROGRAMACIÓN ORIENTADA A OBJETOS 2020-21

Sem.	Actividad Presencial GG	Actividad Presencial GM	Trabajo autónomo
1	Presentación + De C a Java (1, 2, 3)	javac y java. Problema 1	Estudio y resolución de problemas
2	De C a Java (resto)	Problemas 2, 3, 4	Estudio y resolución de problemas
3	Introducción OO (1, 2)	Problemas 5, 6, 7	Estudio y resolución de problemas Práctica 1
4	Introducción OO (3, 4)	Problemas 1, 2	Estudio y resolución de problemas Práctica 1
5	Introducción OO (5, 6, 7)	Problemas 3, 4, 5	Estudio y resolución de problemas Práctica 1
6	Ampliación OO (8, 9)	Problemas 6, 7, 8	Estudio y resolución de problemas Práctica 2
7	Ampliación OO (10, 11)	Problemas 9, 10, 11	Estudio y resolución de problemas Práctica 2
8	Ampliación OO (12, 13, 14)	Exámenes anteriores	Estudio y resolución de problemas
9	Evaluación		
10	Tratamiento ficheros (1, 2, 3)	Sol. examen. Problema 1	Práctica 2
11	Tratamiento ficheros (4, 5, 6)	Problemas 2, 3	Estudio y resolución de problemas Práctica 2
12	Tratamiento ficheros (7, 8)	Problemas 4, 5	Estudio y resolución de problemas Práctica 3
13	Recursividad (1, 2, 3)	Problemas 9, 10	Estudio y resolución de problemas Práctica 3
14	Recursividad (4, 5, 6)	Problemas 1, 2	Estudio y resolución de problemas Práctica 3
15	Recursividad (9, 10)	Exámenes anteriores	Estudio y resolución de problemas
16	Evaluación		
17	Evaluación		
18			Estudio y resolución de problemas Práctica 3
19	Recuperación		

- Los números al lado del nombre del tema indican las secciones de los apuntes de la asignatura.
- Los números al lado de los problemas indican el número de los problemas dentro de la lista.

Sistema de evaluación

Actividades de evaluación

Acronimo	Actividad de Evaluación	Peso	Nota Mínima	En grupo	Obligatoria	Recuperable
EP1	1.º Examen Parcial	25%	4	No	Sí	Sí (con segundo parcial)
EP2	2.º Examen Parcial	25%	4	No	Sí	Sí (examen de recuperación, con nota máxima de 8)
PR1	Práctica 1	15%	No	Sí (≤ 2)	No	Sí (al final de curso, con nota máxima de 8)
PR2	Práctica 2	20%	No	Sí (≤ 2)	No	Sí (al final de curso, con nota máxima de 8)
PR3	Práctica 3	15%	No	Sí (≤ 2)	No	No
Nota Final = $0,25 \cdot EP1 + 0,25 \cdot EP2 + 0,15 \cdot PR1 + 0,2 \cdot PR2 + 0,15 \cdot PR3$						

Observacions:

- La asignatura se aprueba con una nota final igual o superior a 5.
- Un segundo parcial superado permite recuperar el primer parcial.
- La nota del primer parcial sólo se tiene en cuenta si es superior a la del segundo parcial (si no, se coge como nota la del segundo parcial).
- En caso de que el alumno haya de realizar la recuperación, la nota del primer parcial no se tendrá en cuenta y la nota máxima de la parte de teoría será un 8.
- Las dos primeras prácticas son recuperables, pero en caso de reentrega, la nota máxima será un 8.
- Una práctica detectada como copia (o trabajo no original del alumno) tendrá una nota de 0 y no será recuperable.

En caso de que un confinamiento impidiera hacer algún examen de manera presencial, éste se sustituiría por un par de ejercicios de programación individuales.

Bibliografía y recursos de información

Básica

- Apuntes de la asignatura.
- Eric S. Roberts: The Art & Science of Java: An Introduction to Computer Science, Pearson Education, 2008 (hay una versión preliminar disponible en pdf).
- Eric S. Roberts: Thinking Recuersively with Java, John Wiley & Sons, 2006.

Complementaria

- Documentación de la biblioteca ACM Java Task Force <http://jtf.acm.org/>
- Kathy Sierra, Bert Bates: Head First Java, O'Reilly, 2003.
- Jorge A. Villalobos, Rubby Casallas: Fundamentos de Programación. Aprendizaje Activo Basado en Casos. Pearson Pentice-Hall, 2006.