

FICHA TÉCNICA DE LA ASIGNATURA

Datos de la asignatura	
Nombre	Programación Orientada a Objetos
Código	DTC-GITT-223
Titulación	Grado en Ingeniería de Tecnologías de Telecomunicación
Curso	2º
Cuatrimestre	2º
Créditos ECTS	6 ECTS
Carácter	Obligatorio
Departamento	Telemática y Computación
Área	Ingeniería Telemática
Universidad	Universidad Pontificia Comillas
Horario	
Profesores	David Contreras Bárcena
Descriptor	

Datos del profesorado	
Profesor	
Nombre	David Contreras Bárcena
Departamento	Telemática y Computación
Área	Ingeniería Telemática
Despacho	D-410
e-mail	davidcb@comillas.edu
Teléfono	91 542 28 00 ext. 4235
Horario de Tutorías	Petición de cita previa mediante email

DATOS ESPECÍFICOS DE LA ASIGNATURA

Contextualización de la asignatura

Aportación al perfil profesional de la titulación

El objetivo principal de la asignatura es proporcionar al alumno la capacidad de analizar un problema identificando los requisitos software, realizar un diseño orientado a objetos del mismo y codificarlo utilizando de forma eficiente las clases de la API que proporciona Java.

Alcanzar estos objetivos implica aprender el paradigma de la orientación a objetos, conocer el alcance de la especificación del lenguaje (documentación de la API) y la utilización de forma óptima de las clases de la citada especificación.

Al finalizar el curso el alumno adquirirá los conocimientos suficientes para diseñar y desarrollar aplicaciones orientadas a objetos, visuales o embebidas en navegadores y con acceso a ficheros.

Prerrequisitos

Ninguno

Competencias - Objetivos

Competencias

Competencias Generales

CG04. Capacidad de resolver problemas con iniciativa, toma de decisiones, creatividad, y de comunicar y transmitir conocimientos, habilidades y destrezas, comprendiendo la responsabilidad ética y profesional de la actividad del ingeniero técnico de telecomunicación.

CG06. Facilidad para el manejo de especificaciones, reglamentos y normas de obligado cumplimiento.

CG08. Conocer y aplicar elementos básicos de economía y de gestión de recursos humanos, organización y planificación de proyectos, así como de legislación, regulación y normalización en las telecomunicaciones.

Competencias Específicas

CRT7. Conocimiento y utilización de los fundamentos de la programación en redes, sistemas y servicios de telecomunicación.

CETM4. Capacidad de describir, programar, validar y optimizar protocolos e interfaces de comunicación en los diferentes niveles de una arquitectura de redes.

CETM7. Capacidad de programación de servicios y aplicaciones telemáticas, en red y distribuidas.

Aplicar las técnicas de análisis y diseño orientado a objetos mediante el lenguaje de modelado UML.

- RA1. Propone soluciones orientadas a objeto correctamente diseñadas a diferentes problemas de software propuestos.
- RA2. Aplica correctamente los conceptos de polimorfismo, abstracción y encapsulamiento.
- RA3. Diseña programas basados en relaciones de herencia, sabiendo diferenciar entre clases, clases abstractas e interfaces.
- RA4. Comprende los diferentes tipos de relaciones de los diagramas de clase UML.
- RA5. Conoce algunos de los patrones de diseño más importantes como son: MVC, DAO, GRASP o Singleton.
- RA6. Diseña aplicaciones separando claramente sus módulos entre las diferentes capas que intervienen (presentación, lógica, comunicaciones, entrada/salida, etc.).

Programar aplicaciones básicas en el lenguaje de programación Java.

- RA7. Conoce el funcionamiento de la Java Virtual Machine y las plataformas SE y EE.
- RA8. Utiliza con fluidez la documentación de la API del lenguaje.
- RA9. Conoce y utiliza las clases esenciales del lenguaje.
- RA10. Conoce las características básicas de las estructuras List, Set y Map.
- RA11. Aplica una estructura de paquetes adecuado a los programas.
- RA12. Entiende el funcionamiento y la importancia de trabajar correctamente con los distintos modificadores de acceso.

Programar aplicaciones gráficas en Java con almacenamiento de objetos.

- RA13. Realiza diseños usables y amigables de aplicaciones basadas en ventanas las cuales responden a necesidades de software reales.
- RA14. Implementa aplicaciones visuales con componentes más complejos no explicados en el aula.
- RA15. Entiende y aplica los distintos modelos de gestión de eventos.
- RA16. Conoce las soluciones web basadas en tecnología Java del lado del cliente.
- RA17. Gestiona correctamente los errores lanzados por las clases de la API y las creadas propias.
- RA18. Conoce los mecanismos de creación de excepciones propias
- RA19. Implementa soluciones de almacenamiento de objetos a partir de las clases que ofrece la API de Java.

¹ Los resultados de aprendizaje son indicadores de las competencias que nos permiten evaluar el grado de dominio que poseen los alumnos. Las competencias suelen ser más generales y abstractas. Los R.A. son indicadores observables de la competencia

BLOQUES TEMÁTICOS Y CONTENIDOS

Contenidos – Bloques Temáticos
TEMA 1: PLATAFORMA DE PROGRAMACIÓN JAVA
1.1 Java Virtual Machine. 1.2 Plataforma de desarrollo JDK. 1.3 Características y herramientas de la plataforma Java Standard Edition.
TEMA 2: PROGRAMACIÓN ORIENTADA A OBJETOS
2.1 Conceptos básicos: objetos, mensajes e interfaz. 2.2 Definición de una clase: constructores, getters, setters. 2.3 Paradigmas de la POO. 2.4 Clases abstractas e interfaces.
TEMA 3: INTRODUCCIÓN A JAVA SE
3.1 Referencias a objetos: Garbage Collector. 3.2 Upcasting y Downcasting. 3.3 Arrays. 3.4 Interfaz de la clase Object: métodos toString() y equals(). 3.5 Tipos de datos y Wrappers.
TEMA 4: CLASES BÁSICAS
4.1 API del JDK. 4.2 Clases String, Math, Date y StringBuffer/StringBuilder. 4.3 Empaquetado de aplicaciones: JAR.
TEMA 5: PAQUETES
5.1 Diseño de una aplicación basada en paquetes.
TEMA 6: MODIFICADORES DE ACCESO
6.1 Materialización del paradigma de ocultamiento.
TEMA 7. COLECCIONES
7.1 List, Set y Map.
TEMA 8. RELACIÓN DE OBJETOS
8.1 Análisis y diseño orientado a objetos de problemas reales. Lenguaje de modelado UML.
TEMA 9. DESARROLLO DE APLICACIONES VISUALES
9.1 Bibliotecas AWT y Swing. Componentes básicos. 9.2 Gestión de eventos. 9.3 Desarrollo de aplicaciones cliente en entornos web: Applets. 9.4 Diseño y evaluación de interfaces de usuario.
TEMA 10. EXCEPCIONES
10.1 Gestión de errores en Java. 10.2 Definición de excepciones propias.
TEMA 11. ENTRADA Y SALIDA
11.1 Diferencias entre Streams y Readers/Writers. 11.2 Serialización.

METODOLOGÍA DOCENTE

Aspectos metodológicos generales de la asignatura

Con el fin de conseguir el desarrollo de competencias propuesto, la materia se desarrollará teniendo en cuenta la actividad del alumno como factor prioritario. Ello implicará que tanto las sesiones presenciales como las no presenciales promoverán la implicación activa de los alumnos en las actividades de aprendizaje.

Metodología Presencial: Actividades

1. **Lección expositiva:** El profesor desarrolla el tema que previamente los alumnos han leído, explicándolo en la pizarra. Una vez desarrollados los conceptos teóricos se aplican a un caso real. Para ello el profesor y/o los propios alumnos, proponen un problema cotidiano al cual se le intenta dar solución con la participación de los alumnos. A continuación, el profesor codifica dicho programa en el ordenador con la participación de los de los alumnos. Este programa codificado por el profesor es facilitado a los alumnos a través de la web de la asignatura.
2. **Debates grupales, pruebas y resolución de ejercicios:** En estas sesiones se resolverán las dudas surgidas de las lecturas llevadas a cabo por los alumnos de los temas o contenidos propuestos por el profesor. También se realizarán debates grupales y pruebas para facilitar la comprensión de los mismos. Además, se resolverán los pequeños ejercicios propuestos por el profesor.
3. **Prácticas de laboratorio.** A lo largo del curso los alumnos se realizarán prácticas individuales de cada uno de los temas expuestos. En el laboratorio se finalizará el desarrollo de la práctica y se resolverán las dudas pertinentes.
4. **Tutorías.** Se realizarán en grupo e individualmente para resolver las dudas planteadas a los alumnos después de trabajar sobre los distintos temas. También se utilizarán para orientar al alumno en su proceso de aprendizaje.

Metodología No presencial: Actividades

1. **Estudio teórico:**
 - a. Estudio y preparación de los temas o conceptos que vayan a ser expuestos por el profesor.
 - b. Estudio individual y personal por parte del alumno de los conceptos ya expuestos en las lecciones expositivas.
2. **Casos prácticos:** Preparación y comienzo del desarrollo de las prácticas de laboratorio propuestas semanalmente por el profesor.
3. **Proyecto:** Desarrollo de la práctica final de curso sobre la cual se trabajará incrementalmente todo el curso. Se realizarán entregas parciales al profesor.
4. **Trabajo virtual en red:** Mediante la utilización de una Wiki los alumnos deberán desarrollar de forma colaborativa los temas propuestos por el profesor.

El objetivo principal del trabajo no presencial es llegar a entender y comprender los conceptos teóricos de la asignatura, así como ser capaz de poner en práctica estos conocimientos para resolver los diferentes tipos de problemas.

RESUMEN HORAS DE TRABAJO DEL ALUMNO			
HORAS PRESENCIALES			
Lección magistral	Debates y ejercicios	Prácticas laboratorio	Pruebas
32	8	15	5
HORAS NO PRESENCIALES			
Trabajo autónomo sobre contenidos teóricos	Trabajo autónomo sobre contenidos prácticos	Realización de trabajos colaborativos	Estudio
16	62	10	32
CRÉDITOS ECTS:			6 (180 horas)

EVALUACIÓN Y CRITERIOS DE CALIFICACIÓN

Actividades de evaluación	Criterios de evaluación	PESO
Realización de exámenes: <ul style="list-style-type: none"> Examen Intercuatrimestral Examen Final 	<ul style="list-style-type: none"> Comprensión de conceptos mediante pruebas abiertas y realización de un desarrollo de software por escrito. Se valorará la capacidad de análisis y diseño de la solución aportada, así como la limpieza y eficiencia del código escrito. 	65%
Realización de pruebas de seguimiento, debates y ejercicios.	<ul style="list-style-type: none"> Comprensión de conceptos mediante pruebas objetivas y one minute papers. Aplicación de conceptos al desarrollo de pequeños programas temáticos sobre los nuevos conceptos mediante pruebas prácticas abiertas. Análisis e interpretación de los resultados obtenidos en la resolución de problemas mediante debates. Capacidad para profundizar sobre nuevos temas relacionados con la programación Java mediante herramientas colaborativas. 	15%
Prácticas: <ul style="list-style-type: none"> Casos a desarrollar en la clase de prácticas. Proyecto: práctica Final 	<ul style="list-style-type: none"> Aplicación de los conceptos teóricos al desarrollo de casos (pequeños programas temáticos) sobre los nuevos conceptos planteados. Aplicación de todos los conocimientos adquiridos durante el curso en forma de un desarrollo de un proyecto de software incremental, mediante entregas periódicas. Se valorará el diseño de la interfaz visual, el diseño del software desarrollado y el grado de cumplimiento con todos los requisitos propuestos. 	20%

Calificaciones.

Calificaciones

La calificación en la **convocatoria ordinaria** de la asignatura se obtendrá como:

- Un 65% la calificación del examen. La nota del examen final supondrá un 50% de la calificación final en la asignatura y un 15% de la calificación será la del examen intercuatrimestral.
- Un 15% será la calificación de las pruebas de seguimiento.
- Un 20% será la calificación de las prácticas. La práctica final supondrá un 15% y las prácticas realizadas en clase un 5%.

La calificación en la **convocatoria extraordinaria**:

- Un 65% la calificación del exámen. La nota del examen final supondrá un 50% de la calificación de esta convocatoria extraordinaria y un 15% de la calificación será la del examen intercuatrimestral.
- Un 15% será la calificación de las pruebas de seguimiento.
- Un 20% será la calificación de las prácticas. La práctica final supondrá un 15% y las prácticas realizadas en clase un 5%.

Para aprobar la asignatura los alumnos tienen que tener al menos 5 puntos sobre 10 en el examen final de la asignatura, tanto en la convocatoria ordinaria como en la extraordinaria.

PLAN DE TRABAJO Y CRONOGRAMA

Actividades No presenciales	Fecha de realización	Fecha de entrega
<ul style="list-style-type: none">• Lectura y estudio de los contenidos teóricos en el libro de texto	Después de cada clase	
<ul style="list-style-type: none">• Resolución de los problemas propuestos o debates	Semanalmente	
<ul style="list-style-type: none">• Entregas parciales de la práctica final		Semanas 10 y 13
<ul style="list-style-type: none">• Aportaciones realizadas mediante trabajo virtual en red		Semanas 4, 6, 9 y 11
<ul style="list-style-type: none">• Pruebas que se realizarán durante las horas de clase	Semanas 3 y 12	
<ul style="list-style-type: none">• Preparación de la prueba intercuatrimestral y examen final	Abril y junio	
<ul style="list-style-type: none">• Entrega de la práctica final		Día del examen

Semana	Presencial			No Presencial				Total Horas
	Temas	Aula	Lab.	Trab. Autón.	Conten. Prácticos	Trabajos Colab.	Estudio	
1	Tema 1 y 2	3	1			2		7
2	Tema 2	3	1		1		2	7
3	Tema 2	3	1		1		3	7
4	Tema 3 y 4	3	1	4	4	2	2	16
5	Tema 5 y 6	3	1		4		2	10
6	Tema 7	3	1	4	5	2	2	17
7	Prueba Intercuatrim.	3			4		2	6
8	Tema 8	3	1		4		2	10
9	Tema 8	3	1	4	4	2	2	16
10	Tema 9	3	1		5		2	12
11	Tema 9	3	1	4	5	2	2	18
12	Tema 10	3	1		7		3	13
13	Tema 11	3	1		7		2	13
14	Tema 12	3	1		7		2	13
	Examen Final	5			4		3	10
	Total Horas	47	13	16	62	10	32	180

BIBLIOGRAFÍA Y RECURSOS

Bibliografía Básica

- Web específica de la asignatura desarrollada por el profesor con documentación, ejemplos y enunciados de prácticas con sus soluciones: <http://www.labcom.upcomillas.es/poo/>

Bibliografía Complemetaria

- Horstmann. "Core Java 2 Volumen I - Fundamentos 7° Edicion", Prentice Hall. 2005.
- Horstmann. "Core Java 2 Volumen II", Prentice Hall. 2005.
- Bishop. "Java. Fundamentos de Programación", Addison Wesley, Madrid 1999
- Bruce Eckel. "Piensa en JAVA", Addison Wesley, Madrid 2002
- Web oficial de Java: <http://www.oracle.com/technetwork/java/index.html>
- Tutorial Java de Oracle (on-line): <http://www.oracle.com/technetwork/java/index-jsp-135888.html>
- Bruce Eckel. "Thinking in Java" (on-line): <http://www.mindview.net/Books/TIJ/>
- <http://www.programacion.com/java/>
- <http://www.javahispano.org>

