



FICHA TÉCNICA DE LA ASIGNATURA

Datos de la asignatura	
Nombre completo	Programación Orientada a Objetos
Código	DTC-GITT-223
Título	Grado en Ingeniería de Tecnologías de Telecomunicación por la Universidad Pontificia Comillas
Impartido en	Grado en Ingeniería en Tecnologías de Telecomunicación [Segundo Curso] Grado en Ingeniería de Tecnologías de Telecom. y Grado en Análisis de Negocios/Business Analytics [Segundo Curso]
Nivel	Reglada Grado Europeo
Cuatrimestre	Semestral
Créditos	6,0 ECTS
Carácter	Obligatoria (Grado)
Departamento / Área	Departamento de Telemática y Computación
Responsable	David Contreras Bárcena
Horario de tutorías	Enviar email para concertar cita

Datos del profesorado	
Profesor	
Nombre	David Contreras Bárcena
Departamento / Área	Departamento de Telemática y Computación
Despacho	Alberto Aguilera 25 [D-410]
Correo electrónico	davidcb@comillas.edu

DATOS ESPECÍFICOS DE LA ASIGNATURA

Contextualización de la asignatura
Aportación al perfil profesional de la titulación
<p>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.</p> <p>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.</p> <p>Al finalizar el curso el alumno adquirirá los conocimientos suficientes para diseñar y desarrollar aplicaciones de escritorio orientadas a objetos, visuales y con acceso a ficheros.</p>



Prerequisitos

Conocimientos básicos de programación.

Competencias - Objetivos

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.

ESPECÍFICAS

CETM04	Capacidad de describir, programar, validar y optimizar protocolos e interfaces de comunicación en los diferentes niveles de una arquitectura de redes.
CETM07	Capacidad de programación de servicios y aplicaciones telemáticas, en red y distribuidas.
CRT07	Conocimiento y utilización de los fundamentos de la programación en redes, sistemas y servicios de telecomunicación.

Resultados de Aprendizaje

RA1	Conocer la plataforma de desarrollo de software Java. Diferenciar entre las plataformas Java SE y Java EE. Conocer el funcionamiento de la Java Virtual Machine.
RA2	Conocer y comprender los conceptos más importantes del paradigma orientado a objetos. Aplicar correctamente los conceptos de polimorfismo, abstracción y encapsulamiento. Diseñar programas basados en relaciones de herencia, diferenciando entre clases, clases abstractas e interfaces.
RA3	Realizar el análisis y el diseño detallado de las aplicaciones informáticas a partir de patrones de diseño orientados a objetos. Comprender los diferentes tipos de relaciones de los diagramas de clase UML. Conocer algunos de los patrones de diseño más importantes como MVC, DAO o Singleton. Diseñar aplicaciones separando claramente sus módulos entre las diferentes capas que intervienen (presentación, lógica, comunicaciones, entrada/salida, etc.).
	Conocer las características básicas del lenguaje de programación. Utilizar con fluidez la



RA4	documentación de la API del lenguaje. Conocer y utilizar las clases esenciales del lenguaje. Aplicar una estructura de paquetes adecuado a los programas. Entender la utilidad de los distintos modificadores de acceso.
RA5	Conocer las colecciones más importantes que ofrece la API de Java y decidir cuál utilizar en cada caso concreto. Diferenciar entre arrays y colecciones dinámicas. Conocer las características básicas de las estructuras tipo List, Set y Map.
RA6	Diseñar y codificar aplicaciones visuales utilizando componentes de especificaciones futuras. Realizar diseños usables y amigables de aplicaciones basadas en ventanas. Implementar aplicaciones visuales con componentes más complejos no explicados en el aula. Entender y aplicar los diferentes tipos existentes de materializar la gestión de eventos. Implementar soluciones web basadas en tecnología Java del lado del cliente.
RA7	Entender la gestión de errores. Tratar correctamente los errores lanzados en una aplicación. Conocer los mecanismos de creación de excepciones propias.
RA8	Aplicar a los programas desarrollados los mecanismos de persistencia requeridos. Conocer los filtros para operar con ficheros más importantes que ofrece la API de Java y decidir cuál utilizar en cada caso concreto.

BLOQUES TEMÁTICOS Y CONTENIDOS

Contenidos – Bloques Temáticos

Temas

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 STANDARD EDITION

- 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.

5.2 Compilación y ejecución.

TEMA 6: MODIFICADORES DE ACCESO

6.1 Materialización del paradigma de ocultamiento.

TEMA 7. COLECCIONES

7.1 Interfaz Collection

7.2 List, Set y Map.

TEMA 8. RELACIÓN DE OBJETOS

8.1 Análisis y diseño orientado a objetos de problemas reales.

8.2 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 Streams y serialización de objetos.

11.2 Readers y Writers.

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

- Lección expositiva:** El profesor desarrolla el tema que previamente los alumnos han leído, explicándolo en la pizarra y en el ordenador. Una vez desarrollados los conceptos teóricos se aplican a un caso real mediante un pequeño programa, en el que participan los alumnos aportando y sugiriendo ideas. Para el desarrollo de este programa se utilizan herramientas colaborativas. El código resultante de cada clase es facilitado a los alumnos a través de la web de la asignatura.
- Trabajos colaborativos, resolución de ejercicios y pruebas:** En estas sesiones se resolverán las dudas surgidas mediante la codificación colaborativa de programas para facilitar la comprensión de los conceptos teóricos y realización de alguna prueba corta de programación en clase. Se valorará la participación y actitud en clase por parte del alumno.
- Prácticas de laboratorio.** A lo largo del curso los alumnos realizarán prácticas individuales de cada uno de los temas expuestos. Empezarán a trabajar en casa sobre la práctica y en el laboratorio resolverán las dudas que puedan tener finalizando la práctica.

CG04, CG06,
CRT07, CETM04,
CETM07

Metodología No presencial: Actividades

- Estudio teórico:**
 - Estudio y preparación de los temas o conceptos que vayan a ser expuestos por el profesor.
 - Estudio individual y personal por parte del alumno de los conceptos ya expuestos en las lecciones expositivas.
- Casos prácticos:** Preparación y comienzo del desarrollo de las prácticas de laboratorio propuestas semanalmente por el profesor.
- Proyecto:** Desarrollo de la práctica final de curso sobre la cual se trabajará incrementalmente todo el curso. Se realizarán entregas parciales al profesor.

CG04, CG06,
CG08, CRT07,
CETM04, CETM07



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		
Clase magistral y presentaciones generales	Prácticas de laboratorio	Resolución de problemas de carácter práctico o aplicado
35.00	15.00	10.00
HORAS NO PRESENCIALES		
Prácticas de laboratorio	Prácticas de diseño y desarrollo de un proyecto	Estudio de conceptos teóricos fuera del horario de clase por parte del alumno
45.00	40.00	35.00
CRÉDITOS ECTS: 6,0 (180,00 horas)		

EVALUACIÓN Y CRITERIOS DE CALIFICACIÓN

Actividades de evaluación	Criterios de evaluación	Peso
Exámenes: <ul style="list-style-type: none"> • Prueba Intersemestral. • Examen Final. 	<ul style="list-style-type: none"> • Prueba Intersemestral (15%): comprensión de los conceptos teóricos de la programación orientada a objetos mediante una prueba abierta de análisis, diseño y programación mediante el lenguaje Java. • Examen Final (55%): se evaluará la capacidad de análisis, diseño y programación de una solución software completa mediante el lenguaje de programación Java. También se evaluará la limpieza, orden y estilo del código escrito acorde a los patrones y estilos de codificación vistos en clase. 	70 %
Evaluación continua del rendimiento: <ul style="list-style-type: none"> • Proyectos desarrollados por los alumnos. 	<ul style="list-style-type: none"> • La práctica final de la asignatura que constará en el desarrollo de una aplicación gráfica comercial con interfaz gráfico de usuario. Comercial significa que la aplicación estará 	



<ul style="list-style-type: none"> • Trabajos de carácter práctico individual o en grupo. • Ejercicios o problemas resueltos de manera individual o en grupo. • Pruebas cortas de evaluación continua. • Participación en clase. 	<p>al nivel de los pequeños programas que se suelen utilizar hoy en día (Bloc de Notas, Calculadora, etc.): 20%.</p> <ul style="list-style-type: none"> • El desarrollo de pruebas cortas, problemas resueltos en clase (resolución de errores, retos, etc.) y la actitud y participación en clase: 5%. 	25 %
<p>Evaluación del trabajo experimental:</p> <ul style="list-style-type: none"> • Participación en el laboratorio. • Informes de prácticas de laboratorio individual o en grupo. 	<ul style="list-style-type: none"> • La actitud, participación y realización de las prácticas semanales durante las clases de laboratorio: 5% 	5 %

Calificaciones

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

- Un 55% el examen final de la asignatura.
- Un 15% el examen intersemestral.
- Un 20% la práctica final.
- Un 5% las pruebas cortas.
- Un 5% las prácticas semanales.

La calificación en la **convocatoria extraordinaria**:

- Un 55% el examen final de la asignatura.
- Un 15% el examen intersemestral.
- Un 20% la práctica final.
- Un 5% las pruebas cortas.
- Un 5% las prácticas semanales.

Para aprobar la asignatura los alumnos tienen que obtener 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	Fecha de realización	Fecha de entrega
Lectura y estudio de los contenidos teóricos en el libro de texto, apuntes y código facilitado.	Después de cada clase	
	Antes y después de cada	El mismo día de la



Desarrollo de las prácticas individuales semanales	Antes y después de cada sesión	El mismo día de la práctica
Entregas parciales de la práctica final		Semanas 10 y 13
Pruebas que se realizarán durante las horas de clase	Semanas 3, 6 y 12	
Realización de los exámenes	Marzo, mayo y junio	
Práctica final		El mismo día del examen

BIBLIOGRAFÍA Y RECURSOS

Bibliografía Básica

- Moodle.
- Web específica de la asignatura desarrollada por el profesor con ejemplos y enunciados de prácticas con sus soluciones: <https://github.com/DavidContrerasICAI/javaCourseExamples>

Bibliografía Complementaria

- Horstmann. "Core Java 2 Volumen I - Fundamentos 7º Edición", 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>

En cumplimiento de la normativa vigente en materia de **protección de datos de carácter personal**, le informamos y recordamos que puede consultar los aspectos relativos a privacidad y protección de datos que ha aceptado en su matrícula entrando en esta web y pulsando "descargar"

[https://servicios.upcomillas.es/sedelectronica/inicio.aspx?csv=02E4557CAA66F4A81663AD10CED66792](https://servicios.upcomillas.es/sedeelectronica/inicio.aspx?csv=02E4557CAA66F4A81663AD10CED66792)