



## FICHA TÉCNICA DE LA ASIGNATURA

Datos de la asignatura	
NombreCompleto	Programación de Aplicaciones Telemáticas
Código	DSI-TEL-374
Título	<a href="#">Grado en Ingeniería en Tecnologías de Telecomunicación</a>
Impartido en	Grado en Ingeniería en Tecnologías de Telecomunicación y Grado en ADE [Tercer Curso] Grado en Ingeniería en Tecnologías de Telecomunicación [Tercer Curso]
Nivel	Reglada Grado Europeo
Cuatrimestre	Semestral
Créditos	6,0
Carácter	Obligatoria (Grado)
Departamento / Área	Departamento de Telemática y Computación

Datos del profesorado	
<b>Profesor</b>	
Nombre	Jose Álvarez de Perea Llamas
Departamento / Área	Departamento de Telemática y Computación
Correo electrónico	jalvarezperea@icai.comillas.edu
<b>Profesor</b>	
Nombre	José Miguel Ordax Cassá
Departamento / Área	Departamento de Telemática y Computación
Correo electrónico	jmordax@comillas.edu

## DATOS ESPECÍFICOS DE LA ASIGNATURA

Contextualización de la asignatura
<b>Aportación al perfil profesional de la titulación</b>
<p>El objetivo principal de la asignatura es proporcionar al alumno la capacidad de desarrollar aplicaciones web mediante el lenguaje de programación Java.</p> <p>Alcanzar estos objetivos implica conocer los lenguajes básicos de programación del lado del cliente, como son HTML, CSS y Javascript. Además, se deberá dominar criterios de diseño para la implementación óptima de la capa de lógica de negocio.</p> <p>Al finalizar el curso el alumno adquirirá los conocimientos suficientes para desarrollar aplicaciones de acceso a base de datos en entornos web mediante el uso de protocolos TCP/IP.</p>



## Prerrequisitos

Ninguno.

## Competencias - Objetivos

### Competencias

#### GENERALES

<b>CG01</b>	Capacidad para redactar, desarrollar y firmar proyectos en el ámbito de la ingeniería de telecomunicación que tengan por objeto, de acuerdo con los conocimientos adquiridos según lo establecido en el apartado 5 de esta Orden (CIN/352/2009), la concepción y el desarrollo o la explotación de redes, servicios y aplicaciones de telecomunicación y electrónica.
<b>CG04</b>	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.

#### ESPECÍFICAS

<b>CETM04</b>	Capacidad de describir, programar, validar y optimizar protocolos e interfaces de comunicación en los diferentes niveles de una arquitectura de redes.
<b>CETM07</b>	Capacidad de programación de servicios y aplicaciones telemáticas, en red y distribuidas.

## Resultados de Aprendizaje

<b>RA1</b>	Realizar aplicaciones de escritorio avanzadas basadas en Java SE. Concurrencia. Almacenamiento en Base de datos.
<b>RA2</b>	Diseñar y programar páginas web empleando tecnología del lado del cliente. Implementar interfaces web mediante HTML y CSS. Desarrollar páginas web amigables con funcionalidad Javascript.
<b>RA3</b>	Diseñar e implementar aplicaciones web empleando tecnología del lado del servidor. Realizar programas basados en JSP y Servlet entendiendo la diferencia entre ambos. Diseñar y programar soluciones web basadas en el patrón MVC basadas en Servlets y JSP. Conocer y saber aplicar los lenguajes EL y JSTL.
<b>RA4</b>	Comprender el funcionamiento del protocolo HTTP. Conocer las diferencias existentes entre peticiones GET y POST. Gestionar información persistente en el lado del cliente: cookies. Conocer técnicas y principios que permitan mejorar el rendimiento de una aplicación web en el lado del cliente y servidor.



<b>RA5</b>	Mejorar la funcionalidad de una aplicación web. Diferenciar los tres contextos que existen en el lado del servidor: request, session y application. Saber utilizar cuando corresponda los elementos que permiten gestionar el flujo de navegación en el servidor: forward vs redirect. Realizar desarrollos basados en peticiones asíncronas basadas en XML y JSON. Conocer y utilizar frameworks y librerías de ayuda al desarrollo de aplicaciones web en el lado del cliente y servidor.
<b>RA6</b>	Optimizar el rendimiento de una aplicación web. Conocer que elementos afectan al rendimiento y las soluciones que existen para minimizar los tiempos de respuesta.
<b>RA7</b>	Conocer y saber resolver los tipos de ataques web más conocidos. Saber en qué consisten los ataques Form Tampering, SQL Injection, XSS y CSRF. Conocer los principios básicos de la metodología OWASP
<b>RA8</b>	Diseñar e implementar soluciones móviles web. Desarrollar aplicaciones web teniendo en cuenta los dispositivos móviles. Conocer las soluciones existentes para el desarrollo de aplicaciones móviles nativas.

## BLOQUES TEMÁTICOS Y CONTENIDOS

### Contenidos – Bloques Temáticos

#### **Tema 1: INTRODUCCIÓN A JAVA EE**

1.1 Introducción.

1.2 Modelo de aplicación Java EE:

- Componentes Java EE.
- Contenedores Java EE.
- Servicios Java EE.

1.3 Especificaciones Java EE.

1.4 Diseño de aplicaciones Java EE.

1.5 Ensamblado y despliegue de aplicaciones Java EE.

1.6 Servidores de Aplicaciones Java EE.

#### **Tema 2: EL ENTORNO DE DESARROLLO**

2.1 Introducción.

2.2 Descarga, instalación y configuración del entorno:

- Java Runtime Environment (JRE).
- Eclipse IDE for Java EE Developers.
- Apache Tomcat.

2.3 Introducción a IBM Cloud.



### **Tema 3: HTML/CSS**

- 3.1 HTML y CSS.
- 3.2 Hola Mundo.
- 3.3 Etiquetas HTML.
- 3.4 Atributos CSS.
- 3.5 Pre-procesadores HTML y CSS.
- 3.6 Frameworks: Bootstrap.

### **Tema 4: JAVASCRIPT**

- 4.1 Introducción.
- 4.2 Hola Mundo.
- 4.3 Sintaxis, variables, operadores, condicionales y bucles.
- 4.4 Funciones y scope.
- 4.5 ¿Cómo trabajar con el DOM?
- 4.6 Gestión de eventos.
- 4.7 Formularios y validaciones.
- 4.8 Depurador de código (Chrome y Firefox).

### **Tema 5: PROTOCOLO HTTP**

- 5.1 Introducción.
- 5.2 Protocolo HTTP:
  - URL.
  - Conversación HTTP.
  - Sniffer HTTP (Chrome, Firefox y Eclipse).

### **Tema 6: JAVA SERVLETS**

- 6.1 Introducción.
- 6.2 Ciclo de vida de un Java Servlet.
- 6.3 Petición y respuesta de un Java Servlet.
- 6.4 Procesamiento de un formulario HTML.
- 6.5 Empaquetado y despliegue.

### **Tema 7: JAVASERVER PAGES**

- 7.1 Introducción.
- 7.2 Etiquetas.



7.3 Ciclo de vida de un JSP.

7.4 Objetos implícitos de un JSP.

7.5 Empaquetado y despliegue.

## **TEMA 8: DISEÑO DE APLICACIONES JAVA EE**

8.1 Introducción:

- El Controlador.
- La Vista.
- El Modelo.

8.2 Conectando Java Servlets y JavaServer Pages.

8.3 Otros patrones de diseño.

## **TEMA 9: ACCESO A BASE DE DATOS (JAVA SE)**

9.1 Introducción.

9.2 Componentes:

- Driver JDBC.
- Connection.
- Statement, PreparedStatement y CallableStatement.
- ResultSet.

9.3 Bases de Datos:

- Apache Derby.
- MySQL.
- Db2.

9.4 Seguridad:

- SQL Injection.
- Blind SQL Injection.

## **TEMA 10: THREADS (JAVA SE)**

10.1 Introducción.

10.2 Ciclo de vida de un Thread.

10.3 Métodos de sincronización:

- Synchronized.
- Wait/Notify.

## **TEMA 11: JAVA SERVLETS AVANZADO**

11.1 Introducción.



11.2 Configuración de un Java Servlet.

11.3 Concurrencia en un Java Servlet.

11.4 Acceso a Bases de Datos:

- Acceso directo.
- Acceso a través de un Pool de Conexiones.

11.5 Filtros.

11.6 Mantenimiento del estado del cliente:

- Cookies.
- Sesión HTTP.

11.7 Gestión de eventos.

## **TEMA 12: JAVASERVER PAGES AVANZADO**

12.1 Introducción.

12.2 Etiquetas JSP Estándar.

12.3 Lenguaje de Expresiones JSP (EL).

12.4 Librerías de Etiquetas JSP Estándar:

- Core.
- XML.
- I18N.
- Database.
- Functions.

12.5 Librerías de Etiquetas JSP personalizadas.

## **TEMA 13: OTROS CONCEPTOS JAVA EE**

13.1 Introducción.

13.2 Gestión de errores.

13.3 Gestión de seguridad.

13.4 Frameworks de desarrollo web:

- JavaServer Faces.
- Apache Struts.
- Spring Source.
- Apache Wicket.

## **TEMA 14: COMUNICACIONES TCP/IP (JAVA SE)**

14.1 Introducción.

14.2 Desarrollo HTTP de bajo nivel:



- URL.
- URLConnection.
- URLEncoder y URLDecoder.

14.3 Frameworks existentes: Apache HTTP Components.

14.4 Desarrollo TCP/IP de bajo nivel:

- Socket.
- ServerSocket.

## **TEMA 15: AJAX**

15.1 Introducción.

15.2 Formatos JSON y XML.

15.3 Componente XMLHttpRequest.

## **TEMA 16: SEO**

16.1 Introducción.

16.2 Reglas de Google.

16.3 Accelerated Mobile Pages.

## **TEMA 17: HABLAMOS CON UN HACKER**

17.1 Conceptos básicos de seguridad.

17.2 Tipos de ataques:

- Form Tampering.
- Inyecciones SQL.
- Cross-site Scripting (XSS).
- Cross-site Request Forgery (CSRF).

17.3 Desarrollo seguro de aplicaciones.

17.4 Retos de Hacking.

## **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. 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.

**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.

**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.

**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

### 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.

## RESUMEN HORAS DE TRABAJO DEL ALUMNO

HORAS PRESENCIALES		
Clase magistral y presentaciones generales	Resolución en clase de problemas prácticos	Prácticas de laboratorio, preparación y trabajo posterior
37,00	8,00	15,00
HORAS NO PRESENCIALES		





Trabajo autónomo sobre contenidos teóricos por parte del alumno	Trabajo autónomo sobre contenidos prácticos por parte del alumno	Práctica final de la asignatura
60,00	30,00	30,00
<b>CRÉDITOS ECTS: 6,0 (180,00 horas)</b>		

## EVALUACIÓN Y CRITERIOS DE CALIFICACIÓN

Actividades de evaluación	Criterios de evaluación	Peso
Examen final.	<ul style="list-style-type: none"><li>• Comprensión de conceptos mediante pruebas abiertas, enfocado a casos prácticos.</li><li>• Aplicación de los conceptos adquiridos en forma de un desarrollo de software.</li><li>• Presentación y comunicación escrita.</li></ul>	70 %
Realización de pruebas de seguimiento, debates y ejercicios: <ul style="list-style-type: none"><li>• Examen parcial.</li></ul>	<ul style="list-style-type: none"><li>• Comprensión de conceptos mediante pruebas objetivas y one minute papers.</li><li>• Aplicación de conceptos al desarrollo de pequeños programas temáticos sobre los nuevos conceptos mediante pruebas prácticas abiertas.</li><li>• Análisis e interpretación de los resultados obtenidos en la resolución de problemas mediante debates.</li></ul>	20 %
Prácticas: <ul style="list-style-type: none"><li>• Casos a desarrollar en la clase de prácticas.</li><li>• Proyecto: práctica Final.</li></ul>	<ul style="list-style-type: none"><li>• Comprensión de conceptos.</li><li>• Aplicación de conceptos al desarrollo de casos (pequeños programas temáticos) sobre los nuevos conceptos planteados.</li><li>• Aplicación de todos los conocimientos adquiridos durante el curso en forma de un desarrollo de un proyecto de software incremental, mediante entregas periódicas.</li><li>• Realización de pruebas en el laboratorio para comprobar la destreza en un entorno real de</li></ul>	10 %



desarrollo.

## Calificaciones

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

- Un 70% la calificación del examen final.
- Un 20% será la calificación de las pruebas de seguimiento.
- Un 10% será la calificación de las prácticas.

**Nota: Para hacer media los alumnos tienen que tener al menos 5 puntos sobre 10 en el examen final de la asignatura.**

La calificación en la convocatoria extraordinaria:

- Un 100% la calificación del examen final.

## BIBLIOGRAFÍA Y RECURSOS

### Bibliografía Básica

- Web específica de la asignatura desarrollada por el profesor con documentación, ejemplos y enunciados de prácticas en la plataforma Moodle.
- Professional Java for Web Applications. Nicholas S. Williams.

### Bibliografía Complementaria

- Deepak Alur, Dan Malks, John Crupi. "Core J2EE Patterns: Best Practices and Design Strategies (2nd Edition)". Prentice Hall. 2003.
- Eric Jendrock, Ian Evans. "The Java EE 6 Tutorial: Basic Concepts (4th Edition)". Java Series. 2010.
- Web oficial de Java EE: <http://docs.oracle.com/javaee/6/tutorial/doc/>