



FICHA TÉCNICA DE LA ASIGNATURA

Datos de la asignatura	
Nombre completo	Programación de Aplicaciones Telemáticas
Código	DTC-GITT-325
Título	Grado en Ingeniería en Tecnologías de Telecomunicación por la Universidad Pontificia Comillas
Impartido en	Grado en Ingeniería en Tecnologías de Telecom. y Grado en Análisis de Negocios/Business Analytics [Tercer Curso] Grado en Ingeniería en Tecnologías de Telecomunicación [Tercer Curso]
Nivel	Reglada Grado Europeo
Cuatrimestre	Semestral
Créditos	6,0 ECTS
Carácter	Obligatoria (Grado)
Departamento / Área	Departamento de Telemática y Computación
Responsable	Juan Antonio Breña y Wuilmer Peñaloza

Datos del profesorado	
Profesor	
Nombre	Juan Antonio Breña Moral
Departamento / Área	Departamento de Telemática y Computación
Correo electrónico	jabrena@icai.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 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>
Prerequisitos



Ninguno.

Competencias - Objetivos

Competencias

GENERALES

CG01	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.
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.

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.

Resultados de Aprendizaje

RA1	Realizar aplicaciones de escritorio avanzadas basadas en Java SE. Concurrencia. Almacenamiento en Base de datos.
RA2	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.
RA3	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.
RA4	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.
RA5	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.
RA6	Optimizar el rendimiento de una aplicación web. Conocer que elementos afectan al rendimiento y las soluciones que existen para minimizar los tiempos de respuesta.
RA7	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
RA8	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 LA WEB EN LA INDUSTRIA

1. Ciclo de vida del software
2. Paradigmas de programación
3. Sistemas distribuidos
4. Ecosistema Frontend
5. Ecosistema Backend
6. Arquitecturas de software
7. Tendencias

Tema 2: HTTP

1. Introducción World Wide Web
2. ¿Qué es el protocolo HTTP?
3. Evolución del protocolo HTTP
4. HTTP Status Code

Tema 3: HTML

1. Introducción
2. ¿Que es un documento web?
3. HTML
4. Estructura de un documento web
5. Contenido
6. Layout
7. Interacción
8. Seguridad



Tema 4: CSS

1. CSS Selectores
2. CSS Pseudo-classes
3. Pseudo-elements
4. Unidades de medida CSS
5. Contenido
6. Layout
7. Seguridad
8. Responsive Web Design
9. Frameworks CSS
10. Web Components
11. Preprocesadores CSS

Tema 4: JAVASCRIPT

1. Introducción
2. Core Language
3. Javascript & Web Browser
4. Formularios Web
5. Referencias

Tema 6: ESTANDARES JAKARTA EE

1. Jakarta EE
2. Jakarta Contexts and Dependency Injection
3. Bean Validation
4. Jakarta Servlet
5. Jakarta RESTful Web Services
6. Jakarta Persistence
7. Jakarta Messaging Referencias

Tema 7: SPRING BOOT

1. Spring Framework
2. Spring Boot
3. Mi primera aplicación
4. Spring Scaffolding de un proyecto
5. Especificaciones Jakarta EE
6. Spring Core
7. Spring Web
8. Modelos de Concurrencia
9. Error Handling
10. Consumiendo HTTP Endpoints
11. Validacion de Bean
12. Configuracion
13. Logging
14. Scheduling



15. Actuator
16. Spring Security
17. Arquitectura Netflix
18. Arquitectura K8S
19. Servicios auxiliares

TEMA 8: Testing

1. Introduccion
2. Piramide de testing
3. Test funcionales
4. Test no funcionales
5. Naming de tests
6. Tests unitarios
7. Mockito
8. AssertJ
9. Test de Integración
10. Wiremock
11. TestContainers
12. Java Microbenchmark Harness
13. JMeter

TEMA 9: Introduccion a la persistencia de datos

1. Tipos de base de datos
2. Bases de datos relacionales
3. Bases de datos no relacionales
4. SQL

TEMA 10: Bases de datos relacionales

1. Introducción
2. Estandares
3. Conceptos
4. Mapping Objects
5. Repositorios
6. Consultas
7. Transaccionalidad
8. Testing

TEMA 11: Bases de datos no relacionales

1. Introducción
2. Estandares
3. Conceptos
4. Mapping Objects
5. Repositorios
6. Consultas



7. Transaccionalidad

8. Testing

TEMA 12: Marketing Digital

1. SEO

2. Analítica web

3. Accesibilidad

4. Herramientas

TEMA 13: Programación funcional en Java

1. Conceptos funcionales

2. Functional composition

3. Type driven development

4. Functional Error Handling

TEMA 14: Event Driven Architectures

1. Introducción

2. Conceptos

3. Patrones de diseño

4. CQRS

5. Source, Processor & Sink

6. Implementing EDA

7. Testing

TEMA 15: DISEÑO DE APLICACIONES JAVA EE

1. Introducción

2. DDD

3. LEAN

4. SOLID

5. Onion Architecture

6. Patrones de diseño

7. C4 Model & UML

TEMA 16: React

1. Introducción

2. Conceptos

3. Creando tu primera aplicación en React

4. Componentes

TEMA 17: Seguridad

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 de problemas de carácter práctico o aplicado	Prácticas de laboratorio
30.00	15.00	15.00
HORAS NO PRESENCIALES		
Estudio de conceptos teóricos fuera del horario de clase por parte del alumno	Prácticas de laboratorio	Prácticas de diseño y desarrollo de un proyecto
35.00	45.00	40.00
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 (15%) • Examen final (55%) 	<ul style="list-style-type: none"> • Comprensión de conceptos mediante pruebas abiertas, enfocado a casos prácticos. • Aplicación de los conceptos adquiridos en forma de un desarrollo de software. • Presentación y comunicación escrita. 	70 %
Evaluación continua del rendimiento: <ul style="list-style-type: none"> • Proyecto final incremental desarrollado por los alumnos. 	<ul style="list-style-type: none"> • Desarrollo de un proyecto final de la asignatura. 	15 %
Evaluación del trabajo experimental: <ul style="list-style-type: none"> • Pruebas de evaluación del trabajo experimental. • Participación en el laboratorio. 	<ul style="list-style-type: none"> • Aplicación de conceptos al desarrollo de casos (pequeños programas temáticos) sobre los nuevos conceptos planteados. • Realización de pruebas en el 	15 %



<ul style="list-style-type: none">• Informes de prácticas de laboratorio individual o en grupo	laboratorio para comprobar la destreza en un entorno real de desarrollo.
--	--

Calificaciones

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

- Un 55% la calificación del examen final.
- Un 15% será la calificación de la prueba intersemestral.
- Un 15% será el rendimiento en el laboratorio.
- Un 15% la práctica final.

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 55% la calificación del examen final.
- Un 15% será la calificación de la prueba intersemestral.
- Un 15% será el rendimiento en el laboratorio.
- Un 15% la práctica final.

BIBLIOGRAFÍA Y RECURSOS

Bibliografía Básica

- <https://developer.mozilla.org/es/docs/Web/HTTP>
- <https://openwebinars.net/blog/que-es-el-modelo-osi/>
- <https://developer.mozilla.org/enUS/docs/Web/HTML>
- https://www.w3schools.com/html/html_layout.asp
- https://www.w3schools.com/html/html_forms.asp
- <https://html5sec.org/>
- https://developer.mozilla.org/enUS/docs/Web/CSS/Class_selectors
- <https://caniuse.com/>
- <https://www.ecma-international.org/publications/standards/Ecma-262.htm>
- <https://developer.mozilla.org/en-US/docs/Web/JavaScript/Guide>
- <https://www.oreilly.com/library/view/javascript-the-definitive/9781491952016/>
- <https://eloquentjavascript.net/>
- <https://js.do/>
- <https://12factor.net>
- <https://spring.io/cloud>
- <https://docs.spring.io/springboot/docs/current/reference/htmlsingle/>



COMILLAS

UNIVERSIDAD PONTIFICIA

ICAI

ICADE

CIHS

**GUÍA DOCENTE
2020 - 2021**

- <https://landscape.cncf.io>
- <https://github.com/cncf/landscape/blob/master/README.md>

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/>

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>