



GUÍA DOCENTE

PROGRAMACIÓN WEB II, SERVIDOR

DOBLE GRADO EN MATEMÁTICA COMPUTACIONAL E INGENIERÍA DEL SOFTWARE

MODALIDAD: PRESENCIAL

CURSO ACADÉMICO: 2026-2027

Denominación de la asignatura:	Programación Web II, Servidor
Titulación:	Doble Grado en Matemática Computacional e Ingeniería del Software
Facultad o Centro:	Centro Universitario de Tecnología y Arte Digital
Materia:	Programación Web
Curso:	3
Cuatrimestre:	2
Carácter:	OB
Créditos ECTS:	6
Modalidad/es de enseñanza:	Presencial
Idioma:	Castellano
Profesor/a - email	Javier Zorzano Mier javier.zorzano@ext.live.u-tad.com
Página Web:	http://www.u-tad.com/

DESCRIPCIÓN DE LA ASIGNATURA

Descripción de la materia

Esta materia dota de los conocimientos y competencias necesarios por parte de un ingeniero del software para el desarrollo de un proyecto web en sus componentes de servidor y cliente y su potencial exportación como aplicaciones nativas o híbridas.

Descripción de la asignatura

Los navegadores web se comunican con los servidores web mediante el Protocolo de transferencia de hipertexto (HTTP). Cuando hace clic en un enlace en una página web, envía un formulario o ejecuta una búsqueda, se envía una solicitud HTTP desde su navegador al servidor de destino. El código del lado del servidor se puede escribir en cualquier número de lenguajes de programación; ejemplos de lenguajes web populares del lado del servidor incluyen PHP, Python, Ruby, C # y NodeJS (JavaScript). El código del lado del servidor tiene acceso completo al sistema operativo y la base de datos del servidor. La programación del lado del servidor es muy útil porque nos permite entregar de manera eficiente información personalizada para usuarios individuales y, por lo tanto, crear una experiencia de usuario mucho mejor.

El objetivo de este curso es entrar al fondo en el mundo de Programación Web Servidor, entender y practicar temas tan diversos como el almacenamiento y entrega de información de manera eficiente, experiencia de usuario personalizada, acceso controlado al contenido, control información de sesión y estado, notificaciones y comunicación.

COMPETENCIAS Y RESULTADOS DE APRENDIZAJE DE LA MATERIA

Competencias (genéricas, específicas y transversales)

COMPETENCIAS BÁSICAS Y GENERALES

CB1: Que los estudiantes hayan demostrado poseer y comprender conocimientos en un 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.

CB2: Que los estudiantes sepan aplicar sus conocimientos a su trabajo o vocación de una forma profesional y posean las competencias que suelen demostrarse por medio de la elaboración y defensa de argumentos y la resolución de problemas dentro de su área de estudio.

CB4: Que los estudiantes puedan transmitir información, ideas, problemas y soluciones a un público tanto especializado como no especializado.

CB5: Que los estudiantes hayan desarrollado aquellas habilidades de aprendizaje necesarias para emprender estudios posteriores con un alto grado de autonomía

CG1 - Capacidad para entender, planificar y resolver problemas a través del desarrollo de soluciones informáticas.

CG2 - Desarrollo de soluciones informáticas respetuosas con el medio ambiente, los deberes sociales y los recursos naturales, además de cumplir con la legislación y la ética

CG9 - Capacidad para aprender, modificar y producir nuevas tecnologías informáticas

CG10 - Uso de técnicas creativas para la realización de proyectos informáticos

COMPETENCIAS ESPECÍFICAS

CE5 - Capacidad para diseñar e implementar aplicaciones web tanto en lado del cliente como del servidor con tecnologías estándar escalables

CE6 - Conocimiento del uso de mecanismos de comunicación asíncrona entre cliente-servidor y empaquetado de estas aplicaciones web para plataformas móviles para el desarrollo de aplicaciones web dinámicas

CE10 - Capacidad para manejar un gestor de versiones de código y generar la documentación de una aplicación de forma automática.

Resultados de aprendizaje

Al acabar la titulación, el graduado o graduada será capaz de:

- Entender el concepto de programación full stack
- Ser capaz de desarrollar aplicaciones front end en el navegador usando Javascript y sus frameworks
- Entender la familia de protocolos HTTP
- Conocer y aplicar los web services
- Desarrollar aplicaciones de back end con NodeJS y Python
- Conocer el entorno de desarrollo de aplicaciones móviles Android
- Desarrollar una aplicación móvil sencilla en Java/Kotlin

CONTENIDO

HTTP

Frameworks de desarrollo web

Web Services

Escalabilidad de sistemas

Lenguaje de programación servidor (NodeJS o similar)

TEMARIO

Bloque 1: Fundamentos de Backend

T1 - Fundamentos Modernos de Desarrollo Backend con Node.js

T2 - Eventos y Asincronía en Node.js

T3 - HTTP y Enrutamiento

T4 - Framework Express

T5 - MVC y Bases de Datos NoSQL con MongoDB

T6 - Validación Avanzada y Manejo de Errores

T7 - Autenticación y Autorización con JWT

T8 - Documentación, Testing y Monitorización

Bloque 2: Tecnologías Avanzadas

T9 - Supabase y PostgreSQL con Prisma

T10 - Comunicación en Tiempo Real con WebSockets

T11 - TypeScript con Node.js

T12 - Subida de Archivos y Storage

T13 - Deploy y DevOps

ACTIVIDADES FORMATIVAS Y METODOLOGÍAS DOCENTES

Actividades formativas

Actividad Formativa	Horas totales	Horas presenciales
<i>Clases teóricas / Expositivas</i>	22	22
<i>Clases Prácticas</i>	32	32
<i>Tutorías</i>	4	2
<i>Estudio independiente y trabajo autónomo del alumno</i>	48	0
<i>Elaboración de trabajos (en grupo o individuales)</i>	38	0
<i>Actividades de Evaluación</i>	7	7

Metodologías docentes

Método expositivo o lección magistral

Aprendizaje de casos

Aprendizaje basado en la resolución de problemas

Aprendizaje cooperativo o colaborativo

Aprendizaje por indagación

Metodología Flipped classroom o aula invertida

Gamificación

Just in time Teaching (JITT) o aula a tiempo

Método expositivo o lección magistral

Método del caso

Aprendizaje basado en la resolución de problemas

Aprendizaje cooperativo o colaborativo

Aprendizaje por indagación

Metodología flipped classroom o aula invertida

Gamificación

DESARROLLO TEMPORAL

Semana1:T1 - Fundamentos Modernos de Desarrollo Backend con Node.js,

T2 - Eventos y Asincronía en Node.js

Semana2: T3 - HTTP y Enrutamiento

Semana3:T4 - Framework Express

Semana4:T5 - MVC y Bases de Datos NoSQL con MongoDB

Semana5:T6 - Validación Avanzada y Manejo de Errores

Semana6:T7 - Autenticación y Autorización con JWT

Semana7:T8 - Documentación, Testing y Monitorización

Semana8:Semana2:T9 - Supabase y PostgreSQL con Prisma

Semana9:T10 - Comunicación en Tiempo Real con WebSockets

Semana10:T11 - TypeScript con Node.js

Semana11:T12 - Subida de Archivos y Storage

Semana12:T13 - Deploy y DevOps

Semana13,14: práctica final y defensas.

SISTEMA DE EVALUACIÓN

ACTIVIDAD DE EVALUACIÓN	VALORACIÓN MÍNIMA RESPECTO A LA CALIFICACIÓN FINAL (%)	VALORACIÓN MÁXIMA RESPECTO A LA CALIFICACIÓN FINAL (%)
-------------------------	---	---

<i>Evaluación de la participación en clase, en prácticas o en proyectos de la asignatura</i>	10	30
<i>Evaluación de trabajos, proyectos, informes, memorias</i>	30	60
<i>Prueba Objetiva</i>	30	60

CRITERIOS ESPECÍFICOS DE EVALUACIÓN

ACTIVIDAD DE EVALUACIÓN	CONVOCATORIA ORDINARIA	CONVOCATORIA EXTRAORDINARIA
<i>Evaluación de la participación en clase, en prácticas o en proyectos de la asignatura</i>	10	10
<i>Evaluación de trabajos, proyectos, informes, memorias</i>	30	30
<i>Prueba Objetiva</i>	60	60

Consideraciones generales acerca de la evaluación

La prueba objetiva consistirá en la defensa ante el profesor de una práctica final y de un cuestionario tipo test. En el cuestionario será necesario obtener una puntuación mínima de 4 sobre 10.

Para aprobar la asignatura será obligatorio una nota mínima de 5 en ambas prácticas y una media de 5 en el total de la asignatura.

BIBLIOGRAFÍA / WEBGRAFÍA

Bibliografía Básica:

<https://nodejs.org/es/docs/>

<https://www.mongodb.com/docs/>

<https://dev.mysql.com/doc/refman/8.0/en/>

Bibliografía Recomendada:

<https://devdocs.io/express/>

<https://mongoosejs.com/docs/>

<https://decidirv2.api-docs.io/1.0/guia-de-inicio>

<https://expressjs.com/en/guide/routing.html>

<https://kinsta.com/es/blog/nodejs-vs-python/>

MATERIALES, SOFTWARE Y HERRAMIENTAS NECESARIAS

Tipología del aula

Aula teórica

Equipo de proyección y pizarra

Materiales:

Ordenador personal

Software:

VS Code

Node.js