



GUÍA DOCENTE

PROGRAMACIÓN WEB I, CLIENTE

DOBLE GRADO EN MATEMÁTICA COMPUTACIONAL E INGENIERÍA DEL SOFTWARE

MODALIDAD: PRESENCIAL

CURSO ACADÉMICO: 2026-2027

Denominación de la asignatura:	Programación Web I, Cliente
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:	1
Carácter:	OB
Créditos ECTS:	6
Modalidad/es de enseñanza:	Presencial
Idioma:	Castellano
Profesor/a - email	Juan Arechabala Anaya juan.arechabala@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

En esta asignatura se parte de los conceptos aprendidos en Fundamentos del Desarrollo Web, poniendo foco en tecnologías más avanzadas del lado del cliente: profundizando en JavaScript, dando un salto a React para la gestión de interfaces, con Tailwind CSS para los estilos y herramientas como React Hook Form, Zod y TanStack Query para formularios y datos.

Se cerrará con Next.js y con testing de componentes (Vitest y React Testing Library). Se elige este stack porque hoy es el mejor posicionado en la comunidad web.

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

Javascript

AJAX/JQuery

Librerías de CSS y Javascript de abstracción

Otras tecnologías web frontend

TEMARIO

- JavaScript

- o Repaso de fundamentos del desarrollo web: arquitectura web, peticiones, HTML y CSS

- o Introducción a JavaScript: variables, tipos, DOM, control de flujo, funciones

- o JavaScript avanzado: closures, this/call/apply/bind, prototipos y clases, Map/Set

- o Asincronía: event loop, promesas, async/await

- o Peticiones fetch y JSON

- o Módulos ES y herramientas modernas: npm, Vite

- Frameworks avanzados: React

- o Conceptos básicos: componentes, props, estado y ciclo de vida

- o JSX

- o Hooks: useState, useEffect, hooks personalizados, hooks de Actions (React 19)

- o Desarrollos prácticos en React

- Formularios y peticiones HTTP

- o React Hook Form y validación con Zod

- o Formularios accesibles

- o TanStack Query: fetching y caché de datos

- o CORS, variables de entorno y seguridad en el cliente

- Tailwind CSS
 - o Utility-first y comparación con CSS tradicional
 - o Tailwind v4 (CSS-first): sistema de diseño, responsive, modo oscuro
 - o Reutilización de estilos: @apply, componentes
- Frameworks avanzados: Next.js
 - o Introducción y estrategias de renderizado (CSR/SSR/SSG/ISR)
 - o App Router y rutas
 - o React Server Components vs Client Components
 - o Server Actions y Route Handlers
 - o Despliegue (Vercel)
- Testing
 - o Vitest: fundamentos de un test
 - o React Testing Library: queries e interacciones
 - o Testing de hooks personalizados y formularios

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

DESARROLLO TEMPORAL

UNIDADES DIDÁCTICAS / TEMAS PERÍODO TEMPORAL

JavaScript (incluye repaso de fundamentos) Semanas 1–2

React Semanas 3–5

Formularios y peticiones HTTP Semanas 6–7

Tailwind CSS Semanas 8–9

Next.js Semanas 10–12

Testing Semana 13

Práctica final y defensa Semana 14

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>	60	60
<i>Prueba Objetiva</i>	30	30

Consideraciones generales acerca de la evaluación

La nota final de la asignatura se calculará de la siguiente manera:

$$\text{NOTA} = \text{SE1} \times 10\% + \text{SE2} \times 60\% + \text{SE3} \times 30\%$$

SE1 — Participación en clase (10 %): asistencia y dinámicas presenciales (boss fights sin IA, code review en vivo, prompt battle).

SE2 — Ejercicios intermedios (60 %): mejor intento por misión en Webl Arena, nota provisional hasta la defensa.

SE3 — Test y Boss Final (30 %): examen final (test) + implementación y defensa del Boss Final (práctica final).

Consideraciones:

Nota ≥ 5 tanto en SE2 como en SE3. La participación (SE1) no tiene nota mínima, pero afecta a la nota total.

Calificación numérica final de 0 a 10; será requisito indispensable alcanzar una calificación mínima de 5 puntos para poder obtener un aprobado.

No se admitirán entregas fuera de forma y fecha sin causa justificada; si se aceptan, será con una reducción considerable en la nota. Cada misión se entiende como un examen y tendrá derecho a revisión — la defensa oral ya cumple esta función, ya que la nota de cada misión es provisional hasta que se defiende (§3).

Para la nota del Boss Final (práctica final) se tendrá en cuenta tanto la defensa oral y la implementación de la práctica en sí, como las respuestas dadas en el test de Blackboard. El retraso en la entrega conllevará penalización en la nota.

Convocatoria extraordinaria

Se conserva la nota de cualquier componente ya aprobado (≥ 5) en ordinaria: no se repite, ni siquiera si el alumno querría subir nota — extraordinaria es para recuperar lo suspenso, no para mejorar.

SE1 (participación) se congela tal cual quedó en ordinaria, aprobada o no: no tiene nota mínima (§9) y no hay manera de "repetir" clases ya impartidas. Sigue pesando su 10 % en la nota final.

Lo suspenso se recupera con un único instrumento: Test + Práctica final con defensa. Su peso en la nota de extraordinaria pasa a ser el de la(s) parte(s) que sustituye:

Si solo SE3 estaba suspenso → examen de extraordinaria al 30 %, SE1 y SE2 se mantienen con su peso original (10 % y 60 %).

Si SE2 también estaba suspenso (no se repite la intermedia) → su peso se acumula al del examen de extraordinaria, que pasa a valer 90 % (30 % + 60 %); SE1 se mantiene al 10 %.

Sigue aplicando el mínimo de 5 en cada parte recuperada y en la nota final para aprobar.

BIBLIOGRAFÍA / WEBGRAFÍA

U1: MDN, Eloquent JavaScript, javascript.info

U2: MDN, Vite, TypeScript

U3: react.dev

U4: React Hook Form, Zod, TanStack Query, OWASP XSS

U5: Tailwind docs, WebAIM Contrast Checker

U6: Next.js docs, Vercel

U7: Vitest, Testing Library, Kent C. Dodds

MATERIALES, SOFTWARE Y HERRAMIENTAS NECESARIAS

Tipología del aula

Aula teórica

Equipo de proyección y pizarra

Materiales:

VS Code, Node.js, Git (GitHub), Chrome, ESLint, Tailwind CSS IntelliSense

Software:

Navegador Google Chrome.

IDE para desarrollo web (Visual Studio Code).