

## GUÍA DOCENTE 2026-2027

### DATOS GENERALES DE LA ASIGNATURA

|                                         |                                 |  |  |
|-----------------------------------------|---------------------------------|--|--|
| <b>ASIGNATURA:</b>                      | Programación Web I              |  |  |
| <b>PLAN DE ESTUDIOS:</b>                | Grado en Ingeniería Informática |  |  |
| <b>FACULTAD:</b>                        | Escuela Politécnica Superior    |  |  |
| <b>CARÁCTER DE LA ASIGNATURA:</b>       | Obligatoria                     |  |  |
| <b>ECTS:</b>                            | 6                               |  |  |
| <b>CURSO:</b>                           | Cuarto                          |  |  |
| <b>SEMESTRE:</b>                        | Primero                         |  |  |
| <b>IDIOMA EN QUE SE IMPARTE:</b>        | Castellano                      |  |  |
| <b>PROFESORADO:</b>                     | David Pérez                     |  |  |
| <b>DIRECCIÓN DE CORREO ELECTRÓNICO:</b> | david.alvarez@uneatlantico.es   |  |  |

### DATOS ESPECÍFICOS DE LA ASIGNATURA

|                                                                                                                                                                                                                                                                                                                                                            |
|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>REQUISITOS PREVIOS:</b>                                                                                                                                                                                                                                                                                                                                 |
| No Aplica                                                                                                                                                                                                                                                                                                                                                  |
| <b>CONTENIDOS:</b>                                                                                                                                                                                                                                                                                                                                         |
| <ol style="list-style-type: none"> <li>Arquitectura de los sitios Web.</li> <li>Aplicaciones web. <ol style="list-style-type: none"> <li>Introducción a HTML5. Aprender a estructurar páginas web semánticas y accesibles.</li> <li>Diseño Web con CSS3. Estilos visuales avanzados para crear interfaces atractivas y responsivas.</li> </ol> </li> </ol> |

3. ECMAScript6 (JavaScript moderno). Introducir a las nuevas características de ECMAScript 6.
4. Manipulación del DOM con JavaScript. Controlar la estructura y el contenido del DOM.
3. Servicios y Backend.
  1. Comunicación con el servidor. AJAX, Fetch, Axios. Aprender a realizar solicitudes asíncronas al servidor.
  2. API RESTful y SOAP. Conocer las diferencias y características de API REST y SOAP.
  3. NodeJS y NPM (Backend básico). Crear servidores utilizando Node.js y gestionar dependencias con NPM
  4. Integración cliente-servidor. Aprender cómo interactúa el frontend con el backend.
4. Aplicaciones web modernas
  1. PWA. Crear aplicaciones web con características similares a las nativas.
  2. Docker para despliegue y contenedores. Aprender a contenedorizar aplicaciones con Docker.
  3. Orquestación con Kubernetes. Aprender a orquestar contenedores y gestionar aplicaciones escalables.

## COMPETENCIAS

### COMPETENCIAS GENERALES:

Que los estudiantes sean capaces de:

- Capacidad para concebir y desarrollar sistemas o arquitecturas informáticas centralizadas o distribuidas integrando hardware, software y redes.
- Capacidad de explicar y aplicar las materias básicas y tecnologías, que permitan el aprendizaje y desarrollo de nuevos métodos y tecnologías, así como las que les doten de una gran versatilidad para adaptarse a nuevas situaciones.
- Capacidad para resolver problemas con iniciativa, toma de decisiones, autonomía y creatividad. Capacidad para saber comunicar y transmitir los

conocimientos, habilidades y destrezas de la profesión de Ingeniero Técnico en Informática.

- 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
- 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
- Que los estudiantes tengan la capacidad de reunir e interpretar datos relevantes (normalmente dentro de su área de estudio) para emitir juicios que incluyan una reflexión sobre temas relevantes de índole social, científica o ética
- Que los estudiantes puedan transmitir información, ideas, problemas y soluciones a un público tanto especializado como no especializado
- Que los estudiantes hayan desarrollado aquellas habilidades de aprendizaje necesarias para emprender estudios posteriores con un alto grado de autonomía

#### **COMPETENCIAS ESPECÍFICAS:**

Que los estudiantes sean capaces de:

- Capacidad para diseñar, desarrollar, seleccionar y evaluar aplicaciones y sistemas informáticos, asegurando su fiabilidad, seguridad y calidad, conforme a principios éticos y a la legislación y normativa vigente.
- Capacidad para analizar, diseñar, construir y mantener aplicaciones de forma robusta, segura y eficiente, eligiendo el paradigma y los lenguajes de programación más adecuados.
- Capacidad de conocer y aplicar las herramientas necesarias para el almacenamiento, procesamiento y acceso a los Sistemas de información, incluidos los basados en web.
- Capacidad para diseñar y evaluar interfaces persona computador que garanticen la accesibilidad y usabilidad a los sistemas, servicios y aplicaciones informáticas.

#### **RESULTADOS DE APRENDIZAJE:**

En esta asignatura se espera que los alumnos alcancen los siguientes resultados de aprendizaje:

- Conocer los principios de una arquitectura cliente - servidor
- Comprender los estándares de construcción web, tales como el metalenguaje HTML y ser capaz de construir una página web basada en dichos estándares
- Desarrollar una aplicación con lenguaje de servidor incluyendo la conexión con bases de datos

## METODOLOGÍAS DOCENTES Y ACTIVIDADES FORMATIVAS

### METODOLOGÍAS DOCENTES:

En esta asignatura se ponen en práctica diferentes metodologías docentes con el objetivo de que los alumnos puedan obtener los resultados de aprendizaje definidos anteriormente:

- MD1 Método expositivo
- MD2 Estudio y análisis de casos
- MD3 Resolución de ejercicios
- MD4 Aprendizaje basado en problemas
- MD5 Aprendizaje orientado a proyectos
- MD6 Aprendizaje cooperativo / Trabajo en grupos
- MD7 Trabajo autónomo

### ACTIVIDADES FORMATIVAS:

A partir de las metodologías docentes especificadas anteriormente, en esta asignatura, el alumno participará en las siguientes actividades formativas:

| Actividades formativas          |                                      | Horas |
|---------------------------------|--------------------------------------|-------|
| <b>Actividades dirigidas</b>    | Clases expositivas                   |       |
|                                 | Clases prácticas                     |       |
| <b>Actividades supervisadas</b> | Supervisión de actividades           |       |
|                                 | Tutorías (individual / en grupo)     |       |
| <b>Actividades autónomas</b>    | Preparación de clases                |       |
|                                 | Estudio personal y lecturas          |       |
|                                 | Elaboración de trabajos              |       |
|                                 | Trabajo individual en campus virtual |       |

El primer día de clase, el profesor/a proporcionará información más detallada al respecto.

## SISTEMA DE EVALUACIÓN

### CONVOCATORIA ORDINARIA:

En la convocatoria ordinaria de esta asignatura se aplican los siguientes instrumentos de evaluación:

| Actividades de evaluación |                                                     | Ponderación |
|---------------------------|-----------------------------------------------------|-------------|
| Evaluación continua       | Entregas de Portfolios y Ejercicios                 | 45 %        |
|                           | Interés y participación del alumno en la asignatura | 5 %         |
| Evaluación final          | Examen Teórico-Práctico                             | 50 %        |

La calificación del instrumento de la evaluación final (tanto de la convocatoria ordinaria como de la extraordinaria, según corresponda) **no podrá ser inferior, en ningún caso, a 4,0 puntos** (escala 0 a 10) para aprobar la asignatura y consecuentemente poder realizar el cálculo de porcentajes en la calificación final.

En la circunstancia que en esa Prueba o Examen Final se obtenga una calificación de 3.9 o inferior, en el Acta de Evaluación correspondiente se registrará la nota o calificación obtenida en dicho Examen Final, sin hacer media aritmética con las notas obtenidas en la Evaluación continua.

El alumno que suspenda la asignatura en la convocatoria ordinaria podrá examinarse en la convocatoria extraordinaria, donde se tendrán en cuenta las calificaciones obtenidas en la Evaluación continua a lo largo del semestre.

### CONVOCATORIA EXTRAORDINARIA:

La convocatoria extraordinaria tendrá lugar durante el mes de julio (consúltase el calendario académico fijado por la universidad). Esta consistirá en la realización de uno o dos Exámenes Teórico-Prácticos con un valor de hasta el 50% de la nota final de la asignatura. El resto de la nota se complementará con la calificación obtenida en la evaluación continua de la convocatoria ordinaria.

## BIBLIOGRAFÍA Y RECURSOS DE REFERENCIA GENERALES

### BIBLIOGRAFÍA BÁSICA:

Las siguientes referencias son de consulta obligatoria:

- Eloquent JavaScript, 3rd Edition - Marijn Haverbeke

📄 URL: <https://eloquentjavascript.net/>

- Descripción: Un excelente libro interactivo en línea para aprender JavaScript moderno (ES6) y conceptos avanzados del lenguaje.
- Temas: JavaScript, DOM, programación funcional.
- You Don't Know JS (Yet), 2nd Edition - Kyle Simpson
  - URL: <https://github.com/getify/You-Dont-Know-JS>
  - Descripción: Una serie de libros que profundizan en los aspectos fundamentales y avanzados de JavaScript.
  - Temas: JavaScript (ES6+), programación asíncrona, promesas, fetch.
- MDN Web Docs - Mozilla Foundation
  - URL: <https://developer.mozilla.org/>
  - Descripción: Documentación oficial y tutoriales de HTML5, CSS3, y JavaScript. Es una de las referencias más importantes para desarrolladores web.
  - Temas: HTML5, CSS3, JavaScript, APIs web.
- Node.js Official Documentation - Node.js Foundation
  - URL: <https://nodejs.org/en/docs/>
  - Descripción: Documentación oficial para aprender sobre Node.js, el entorno de ejecución JavaScript en el backend.
  - Temas: Node.js, manejo de servidores, APIs, npm.
- Express.js Documentation - Express.js
  - URL: <https://expressjs.com/>

- Descripción: La guía oficial para aprender Express.js, un framework ligero para desarrollar servidores y APIs RESTful con Node.js.
- Temas: Node.js, Express, API REST, middleware

### **BIBLIOGRAFÍA COMPLEMENTARIA:**

Las siguientes referencias no se consideran de consulta obligatoria, pero su lectura es muy recomendable para aquellos estudiantes que quieran profundizar en los temas que se abordan en la asignatura.

- Learning Progressive Web Apps - Google Developers
  - URL: <https://developers.google.com/web/ilt/pwa>
  - Descripción: Una colección de recursos y tutoriales ofrecidos por Google para aprender a crear Progressive Web Apps (PWA).
  - Temas: PWA, service workers, caché, manifest.
- RESTful Web APIs - Leonard Richardson, Mike Amundsen, Sam Ruby
  - URL: <https://restfulapi.net/>
  - Descripción: Un sitio web completo que ofrece recursos sobre cómo diseñar APIs RESTful, con tutoriales y mejores prácticas.
  - Temas: APIs REST, JSON, HTTP.
- Docker Documentation - Docker Inc.
  - URL: <https://docs.docker.com/>
  - Descripción: Guía oficial de Docker, ideal para aprender cómo contenedorizan aplicaciones y manejar la infraestructura basada en contenedores.
  - Temas: Docker, contenedores, imágenes, docker-compose.
- Kubernetes Documentation - Kubernetes

- 🌐 URL: <https://kubernetes.io/docs/>
- 🌐 Descripción: Documentación oficial de Kubernetes, el sistema de orquestación de contenedores más popular.
- 🌐 Temas: Kubernetes, pods, servicios, despliegue de aplicaciones.

#### **WEBS DE REFERENCIA:**

- “Mozilla Developer Network web docs: Learn web development”  
<https://developer.mozilla.org/en-US/docs/Learn>
- Documentación oficial de NodeJS: <https://nodejs.org/en/docs/>

#### **OTRAS FUENTES DE CONSULTA:**

- “Aprende desarrollo web”. <https://developer.mozilla.org/es/docs/Learn>