

TS5C4: Programación Web

Nombre del programa académico	Tecnología en Desarrollo de Software
Nombre completo de la asignatura	Programación Web
Área académica o categoría	Programación
Semestre y año de actualización	2024
Semestre y año en que se imparte	5 Semestre
Tipo de asignatura	[x] Obligatoria [] Electiva
Número de créditos ECTS	6
Director o contacto del programa	Guillermo Roberto Solarte Martínez
Coordinador o contacto de la asignatura	

Descripción y contenidos

<p>1. Breve descripción</p> <p>Se conoce como desarrollo web al proceso de crear y mantener un sitio web que sea funcional en internet, a través de diferentes lenguajes de programación, según el modelo y la parte de la página que corresponda.</p>
<p>2. Objetivos</p> <p>Correspondencia con los objetivos del programa:</p> <ul style="list-style-type: none"> - Al finalizar el curso, los estudiantes deberán tener un dominio sólido de las tecnologías front-end, incluyendo HTML, CSS, JavaScript y un framework moderno (por ejemplo, React, Angular, o Vue.js). - Los estudiantes deberán adquirir habilidades en el desarrollo back-end, comprendiendo cómo crear y gestionar servidores, así como interactuar con bases de datos. - Los estudiantes deberán ser capaces de diseñar, desarrollar y desplegar aplicaciones web completas, abarcando tanto el front-end como el back-end.
<p>3. Resultados de aprendizaje</p> <p>Competencias específicas:</p> <ul style="list-style-type: none"> - Creación de páginas web estáticas y básicas con interactividad. - Creación de aplicaciones web simples con funcionalidad del lado del servidor. - Desarrollo y consumo de API básicas. - Desarrollo de proyectos que abarquen tanto la interfaz de usuario como la lógica del servidor. - Presentación de un proyecto completo y un portafolio que destaque las habilidades adquiridas. - Aplicación de mejores prácticas de codificación y conciencia de la seguridad en el desarrollo web <p>Otras competencias por formar:</p> <ul style="list-style-type: none"> - Capacidad para enfrentar desafíos y crear soluciones eficientes. - Participación activa en discusiones, colaboración en proyectos y presentación efectiva de soluciones. - Elección y uso efectivo de herramientas según los requisitos del proyecto.

<p>4. Contenido</p> <p>Módulo 1: Fundamentos de la Web (5 semanas)</p> <ul style="list-style-type: none"> - Introducción a HTML y creación de documentos web. - Estilización con CSS para diseñar páginas web básicas. - Introducción a JavaScript y manipulación del DOM. <p>Módulo 2: Frameworks y herramientas (2 semanas)</p> <ul style="list-style-type: none"> - Control de versiones con Git y GitHub. - Introducción a React, Angular o Vue.js (elección del estudiante). - Creación de componentes y manejo de eventos. <p>Módulo 3: Desarrollo Back-End (6 semanas)</p> <ul style="list-style-type: none"> - Configuración de un servidor local con un framework (por ejemplo, Node.js). - Introducción a bases de datos SQL y NoSQL. - Desarrollo de una API simple usando Express (Node.js). <p>Módulo 4: Integración Front-End y Back-End (3 semanas)</p> <ul style="list-style-type: none"> - Conexión de la interfaz de usuario con la API. - Manejo de datos en la aplicación completa.
<p>5. Requisitos</p> <ul style="list-style-type: none"> - Familiaridad con conceptos básicos de bases de datos. - Comprender los conceptos fundamentales de HTML y CSS. - Lógica de Programación Básica.
<p>6. Recursos</p> <ul style="list-style-type: none"> - Libros: <ul style="list-style-type: none"> → You Don't Know JS (https://github.com/getify/You-Dont-Know-JS/blob/1st-ed/README.md) → Building Front-End Web Apps with Plain JavaScript (https://web-engineering.info/JsFrontendApp-Book/) - Herramientas: <ul style="list-style-type: none"> → Node.js y npm → IDE (visual studio code, vim, etc) → Navegador Web (firefox, safari, etc)
<p>7. Herramientas técnicas de soporte para la enseñanza</p> <ul style="list-style-type: none"> - Herramienta de comunicación (Classroom) - Repositorios de código (GitHub, Bitbucket, etc) - Jueces en línea (CodeForces, AdventureJS, etc)
<p>8. Trabajos en laboratorio y proyectos</p> <ul style="list-style-type: none"> - Talleres específicos para cada tecnología. - Retos para mejorar la lógica de programación. - Proyectos de impacto.
<p>9. Métodos de aprendizaje</p> <ul style="list-style-type: none"> - Aprendizaje basado en proyectos. - Clases magistrales y demostraciones. - Estudio de casos específicos. - Investigaciones. - Videos y presentaciones.
<p>10. Métodos de evaluación</p>

- Tareas:
Se proponen una serie de retos para mejorar las habilidades en los lenguajes y la lógica de programación.
- Miniproyecto 1:
Creación de un portafolio básico, en el que se reflejen características del estudiante.
- Miniproyecto 2:
Creación de una página web usando una fake API.
- Miniproyecto 3:
Creación de una API la cual ejecute las operaciones elementales (crear, leer, actualizar y eliminar)
- Proyecto final:
Creación de una página web completa, que se alimente de un back end comunicado por medio de una API.