Introducción

El desarrollo web moderno abarca un conjunto diverso de herramientas, arquitecturas y enfoques que permiten crear aplicaciones dinámicas, escalables y seguras. Ya no basta con dominar HTML o CSS; el profesional de hoy debe comprender cómo se estructuran los sistemas distribuidos, cómo se integran servicios externos, cómo se protege la información y cómo se automatiza la entrega del software.

Este laboratorio invita al estudiante a investigar categorías clave del ecosistema web actual, comprendiendo su función dentro de un flujo de desarrollo profesional. A partir de esa exploración, cada estudiante seleccionará tecnologías o herramientas reales que representen cada categoría, las documentará técnicamente y mostrará cómo podrían aplicarse en un proyecto real.

Objetivo general

Investigar y documentar el funcionamiento de herramientas modernas utilizadas en el desarrollo web profesional. El propósito es fortalecer el criterio técnico del estudiante para seleccionar tecnologías según necesidades reales de proyectos.

Objetivos específicos

- Identificar y describir categorías fundamentales del desarrollo web moderno y su rol dentro de un flujo de desarrollo completo.
- Seleccionar tecnologías o herramientas que representen cada categoría, analizando sus características, requisitos, ventajas y limitaciones.
- Documentar técnicamente cada categoría en formato Markdown, utilizando ejemplos prácticos y diagramas explicativos.
- Comparar tecnologías dentro de una misma categoría (cuando aplique), evaluando su aplicabilidad en distintos contextos.
- Desplegar el resultado de la investigación como un sitio estático funcional, aplicando herramientas modernas de publicación web.

Estructura del laboratorio

1. Investigación y documentación técnica (Markdown)
2. Despliegue automatizado de sitio web con Markdown

Parte 1 – Investigación: Características del desarrollo web moderno

El estudiante investigará y documentará **cinco categorías clave** del desarrollo web actual, con base en fuentes confiables y ejemplos reales. La documentación se realizará en un archivo README.md estructurado con Markdown.

Categorías obligatorias:

1. **Frameworks de desarrollo web**
    a. ¿Qué es un framework y qué problema resuelve?
    b. Arquitectura general y enfoque (MVC, SPA, SSR, etc.).
    c. Ejemplo práctico documentado (estructura de proyecto, fragmento de código comentado).
    d. Comparación breve entre al menos dos frameworks (según lenguaje o enfoque).
2. **Control de versiones y trabajo colaborativo**
    a. ¿Qué es el control de versiones y por qué es esencial?
    b. Conceptos clave: repositorio, commit, branch, merge, pull request.
    c. Flujos de trabajo comunes (Git Flow, trunk-based, feature branches).
    d. Ejemplo de cómo usar Git en un proyecto (inicialización, commits, ramas).
    e. Herramientas recomendadas (GitHub, GitLab, Bitbucket).
3. **Autenticación y seguridad moderna**
    a. Conceptos: autenticación, autorización, tokens, JWT, OAuth.
    b. Diagrama de flujo explicativo del proceso de autenticación con JWT.
    c. Buenas prácticas en seguridad web.
    d. Aplicaciones reales en plataformas modernas.
4. **Gestores de contenido desacoplados (Headless CMS)**
    a. Definición de Headless CMS vs CMS tradicional.
    b. Arquitectura basada en APIs.
    c. Ventajas, limitaciones y casos de uso comunes.
    d. Ejemplo de cómo se conecta el frontend a un CMS headless.
5. **Pasarelas de pago en aplicaciones web**
    a. ¿Qué es una pasarela de pago? ¿Qué rol cumple en una aplicación moderna?
    b. Requisitos comunes: cuenta de comercio, seguridad, integración técnica.
    c. Ventajas y limitaciones de integrar pagos en línea.
    d. Comparación entre al menos dos pasarelas (ej. Stripe, TiloPay, Bancos, etc.)
6. **Automatización del despliegue y hosting moderno**

a. ¿Qué es CI/CD y por qué se usa en desarrollo web?
b. Hosting estático vs dinámico.
c. Flujo de despliegue automatizado.
d. Documentar el proceso seguido para desplegar la parte 2 del laboratorio

Parte 2: Publicación del sitio web

En esta segunda parte del laboratorio, pondrás en práctica una de las características clave del desarrollo web moderno: el despliegue automático de sitios estáticos en línea. A partir del contenido documentado en tu README.md, vas a publicar un sitio web usando Netlify. Ejemplo: https://ic8057-lab1.netlify.app/

¿Qué se debe hacer?

1. **Crear un proyecto local** en una carpeta llamada desarrollo-moderno-nombre.
2. **Utilizar los siguientes archivos**:
   - index.html (provisto por el docente): archivo base que carga y muestra el contenido del README.
   - README.md: el archivo que documentaron durante la Parte 1.
3. **Seguir el paso a paso que ya documentaron** en su sección de tutorial para subir el proyecto a GitHub y publicar el sitio con Netlify.
4. **Verificar visualmente que el sitio funcione correctamente**, es decir:
   - Que la página se cargue sin errores.
   - Que el contenido del README.md se muestre dentro del index.html.
5. **Compartir la URL pública del sitio desplegado** como parte de su entrega.

Rubrica de evaluación

La evaluación del laboratorio se realizará bajo los siguientes criterios:

Criterio	1 - Insuficiente	2 - Deficiente	3 - Aceptable	4 - Bueno	5 - Excelente
1. Investigación técnica por categoría *(Se valoran la profundidad, claridad, precisión técnica y el uso de ejemplos o comparaciones cuando corresponda)*	No se desarrolla o contiene errores graves en varias categorías.	Contenido incompleto, superficial o con explicaciones vagas.	Desarrollo aceptable pero con poca profundidad o sin ejemplos claros.	Categorías bien explicadas, con ejemplos relevantes y algunas comparaciones.	Desarrollo completo, técnico y detallado. Usa ejemplos adecuados y compara tecnologías con criterio.

2. Uso profesional de Markdown *(Títulos, listas, tablas, formato consistente, legibilidad)*	Desorganizado, sin sintaxis Markdown.	Uso pobre, con errores frecuentes.	Markdown funcional pero básico o inconsistente.	Bien estructurado y legible, con formato claro.	Markdown profesional, con estilo técnico claro, limpio y organizado.
3. Despliegue técnico y documentación *(Sitio funcional, hosting usado, documentación del proceso clara)*	No se desplegó o presenta errores críticos.	Despliegue incompleto o sin documentación adecuada.	Sitio publicado con errores menores, documentación poco clara.	Sitio funcional y correctamente documentado.	Sitio 100% funcional, con proceso documentado paso a paso y sin errores.

Entrega

Cada estudiante deberá entregar un archivo comprimido (.zip) con el proyecto completo, siguiendo estas instrucciones:

1. Una carpeta con el repositorio local completo, incluyendo:
    a. README.md: documento en formato Markdown con la investigación y el tutorial.
    b. index.html: archivo base provisto para visualizar el contenido.
2. Un archivo denominado enlaces.txt que contenga
    a. URL del repositorio público en GitHub:
    b. URL del sitio desplegado (GitHub Pages o Netlify)

Fecha límite: Al finalizar la clase del día asignado para el laboratorio.

Forma de entrega: El archivo .zip debe subirse a la plataforma TEC Digital, en el espacio correspondiente al curso.

Portafolio del curso: Este laboratorio forma parte del portafolio individual del curso.