

Desarrollarás una página web completa, que combine todos los conocimientos adquiridos a lo largo del curso. El proyecto consistirá en la creación de un sitio web de **e-commerce** dinámico e interactivo, que consuma datos de una API REST para mostrar productos, y permita a los usuarios añadir productos a un carrito de compras.

Pasos a seguir:

- **HTML:** Uso de etiquetas semánticas para organizar la página.
- **CSS:** Implementación de un diseño responsivo y atractivo usando Bootstrap y Flexbox.
- **JavaScript:** Integración de una API REST para obtener datos y renderizar productos en el DOM, además de la funcionalidad de un carrito de compras usando `localStorage`.
- **Accesibilidad y SEO:** Implementar prácticas que mejoren la experiencia del usuario y optimicen la página para los motores de búsqueda.

Proyecto Final de E-commerce

Puntos Clave para Revisión y Entrega

- **Subida del Proyecto:** Debe estar disponible en GitHub Pages o Netlify para facilitar su acceso.
- **Control de versiones:** Mantener un historial de commits detallado para documentar cada avance.
- **Presentación:** El archivo [README.md](#) debe incluir una descripción del proyecto, las tecnologías usadas, instrucciones de instalación y cualquier detalle relevante.

Formato de Entrega

Carpeta en Drive

Crear una carpeta en drive (pública) que contenga los archivos y carpetas que conforman tu proyecto.

Link de Entrega

Los links deberán ser entregados en el apartado de “Pre-Entrega de Proyecto” en el Campus Virtual.

Hosting

Compartir el link de Github Pages o subida a servidor.

Requisitos para la entrega

Entrega de Proyecto Final

Obligatorio | Entregable

A partir de la **Clase N° 15** tendrás **7 días** de corrido para realizar la entrega en el campus virtual

1. Estructura Básica de HTML.

- **Estructura semántica:** El HTML debe estar dividido en las etiquetas semánticas principales: **header**, **nav**, **main**, **section**, **footer**.

[README.md](#): Incluir un archivo que explique brevemente el propósito de la página.

2. Formulario de Contacto.

Formulario funcional: Crear un formulario de contacto con campos para **nombre**, **correo electrónico** y **mensaje**, utilizando **Formspree** para manejar el envío de datos.

Entrega de Proyecto Final

Obligatorio | Entregable

3. Estilos Básicos Aplicados con CSS

- **Archivo styles.css:** El proyecto debe contar con un archivo CSS externo que incluya:
 - Estilos básicos aplicados a las secciones de **header**, **footer** y **lista de navegación**.
 - **Fuentes de Google Fonts** correctamente implementadas.

Propiedades de **background** aplicadas en alguna sección de la página (color, imagen, degradado, etc.).

4. Diseño Responsivo con Flexbox y Grid

- **Sección "Productos"**: Organizada en **cards** de forma responsiva utilizando **Flexbox**.
- **Sección "Reseñas"**: Organizada utilizando **Grid**, con una distribución lógica y estética.
- **Sección "Contacto"**: Debe ser responsiva mediante el uso de **Media Queries** para adaptarse a diferentes tamaños de pantalla.

Pre-Entrega de Proyecto

Obligatorio | Entregable

5. Contenido Multimedia y Navegación

- **Multimedia**: deberá incluir archivos multimedia (imagenes, video o iframe) correctamente integrado en la página.

Lista de navegación: Implementar una **lista desordenada** con enlaces que simulen una navegación interna (Inicio, Productos, Contacto, etc.).

6. Subida del Proyecto.

El proyecto debe estar subido a un **hosting gratuito** (Netlify o GitHub Pages), con una **URL funcional** para visualizar el sitio.

Pre-Entrega de Proyecto

Obligatorio | Entregable

7. JavaScript

- **Script.js**: deberá incluir un archivo. Debes crear un archivo script.js para manejar toda la interactividad de la página.

Asegúrate de enlazarlo correctamente en tu archivo HTML.

- **DOM**: Implementa funciones para validar formularios (ej., campos requeridos y formato de correo).

Usa JavaScript para manipular elementos del DOM, por ejemplo, actualizar el carrito y mostrar mensajes al usuario

- **Fetch Api**

Consume datos desde una API REST usando fetch.

Muestra los productos obtenidos de la API en la página en forma de tarjetas (cards).

- **Visualización de Productos:**

Cada producto debe tener su imagen, título y precio, mostrando una lista atractiva para el usuario.

Pre-Entrega de Proyecto

Obligatorio | Entregable

8. Carrito de compras dinámico

- **Agregar Productos al Carrito:** Implementa un carrito de compras donde los usuarios puedan añadir productos desde las tarjetas.
- **Uso de localStorage o sessionStorage:** Guarda el estado del carrito en localStorage o sessionStorage para que no se pierda al actualizar o cerrar la página.

Contador Dinámico: Muestra el número total de productos en el carrito y asegúrate de actualizarlo en tiempo real.

9. Edición y visualización del carrito

- **Visualización de Productos en el Carrito:** Muestra una lista de productos añadidos al carrito, incluyendo cantidad, precio y total.
- **Edición de Cantidades y Eliminación de Productos:** Implementa funciones para que el usuario pueda editar la cantidad de cada producto o eliminarlo del carrito.

Total Dinámico: Actualiza el total de la compra cada vez que se modifiquen los productos en el carrito.

Pre-Entrega de Proyecto

Obligatorio | Entregable

10. SEO & Accesibilidad

Buenas Prácticas de Accesibilidad:

- Usa alt en las imágenes para mejorar la accesibilidad.
- Asegúrate de que se pueda navegar fácilmente con el teclado.

SEO Básico:

- Usa metaetiquetas en el head del HTML para optimizar el SEO.

✨ ✨ ✨ Funcionalidad esperada:

Interactividad Completa:

- La página debe permitir al usuario ver productos, añadirlos al carrito, editar el carrito, y simular la compra.

Formulario de Contacto:

- Implementa un formulario funcional que envíe datos a través de Formspree.

Diseño Responsivo:

- Asegúrate de que el diseño sea adaptable a diferentes tamaños de pantalla.

Persistencia del Carrito:

El carrito debe mantenerse activo incluso si el usuario cierra o actualiza la página, usando localStorage o sessionStorage.