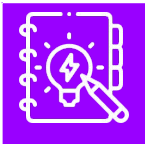


# Proyecto Final

**Clase 15 - “Entrega de proyecto”**



# Proyecto Integrador

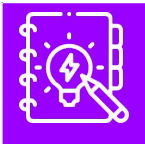


## Obligatorio | Entregable

Desarrollarás una página o sitio web completa, que combine todos los conocimientos adquiridos a lo largo del curso. Este 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. Tu proyecto deberá demostrar una sólida comprensión de la estructuración semántica, el diseño responsivo, la manipulación del DOM, la interacción con APIs, y la implementación de funcionalidades específicas de un e-commerce.

### Requerimientos:

- Estructura HTML avanzada.
- Estilos CSS avanzados.
- JavaScript
- Accesibilidad y SEO.
- Control de Versiones y Documentación.



# Proyecto Integrador



Obligatorio | Entregable

## Estructura básica HTML

1. Generar la estructura básica HTML dividida en etiquetas semánticas estructurales principales (header, nav, main, section, footer).
2. Inclusión de un archivo README.md que explique brevemente el propósito de la página.

## Formulario de contacto

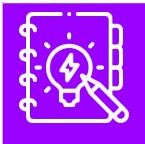
1. Crear de un formulario de contacto con campos de nombre, correo electrónico y mensaje, utilizando Formsfree.

## Contenido Multimedia y Navegación

1. Incluir al menos un archivo multimedia (imagen, vídeo o audio) correctamente integrado en la página.
2. Implementación de una lista desordenada con enlaces que simule una navegación interna en la página (Inicio, Productos, Contacto, etc.).

## Subida del Proyecto

1. El proyecto debe estar subido a un hosting gratuito (Netlify o GitHub Pages), con una URL funcional.



# Proyecto Integrador



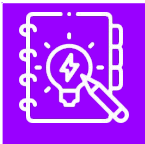
Obligatorio | Entregable

## Estilos Básicos Aplicados con CSS

1. El proyecto debe tener un archivo **styles.css** que contenga los estilos básicos aprendidos aplicados al header, footer, y lista de navegación.
2. Incorporar fuentes de Google Fonts.
3. Aplicar las propiedades de background en alguna sección de la página.

## Diseño Responsivo con Flexbox y Grid

1. Una sección, por ej: "Productos", debe estar organizada en cards de manera responsiva utilizando propiedades de Flexbox.
2. Una sección, por ej: "Reseñas", debe estar organizada utilizando Grid.
3. Una sección, por ej: "Contacto", debe ser responsive utilizando Media Queries



# Proyecto Integrador



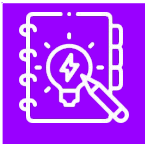
Obligatorio | Entregable

## Modelo de Caja

1. Implementar las propiedades de Box Modeling (márgenes, rellenos, bordes) en los diferentes elementos del proyecto.

## Uso de Bootstrap y Git

1. Reemplazar el nav creado anteriormente por el navbar responsivo creado con Bootstrap.
2. Adaptar la navbar de Bootstrap a la paleta de colores de tu proyecto.
3. El proyecto debe estar en un repositorio GitHub, con commits que documenten los cambios.



# Proyecto Integrador



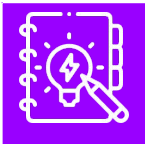
Obligatorio | Entregable

## Funcionalidad JavaScript

1. Crear un archivo script.js y linkearlo en el documento HTML.
2. Debemos poder revisar el código por medio del inspector que nos ofrece nuestro navegador y asegurarnos de que nuestro proyecto no nos está dando ningún error.

## Condicionales y Ciclos

1. Implementar una función que verifique si todos los campos del formulario de contacto están completos, mostrando un mensaje en la consola.
2. Crear un ciclo que genere dinámicamente una lista de productos disponibles y los muestre en la consola



# Proyecto Integrador



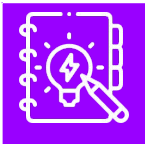
Obligatorio | Entregable

## Manipulación Básica del DOM y Eventos

1. Implementar un evento click que muestra la descripción ampliada del producto que clickeamos.
2. Crear un listado de productos incluidos en nuestro HTML generado por medio de una función en Js.

## Funciones Modulares

1. Crear de una función que cree un array de productos y los muestre en la página utilizando una plantilla HTML dinámica.



# Proyecto Integrador



Obligatorio | Entregable

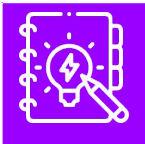
## Asincronía y Consumo de API REST

1. Utilización de `fetch` para obtener datos de una API pública y mostrarlos en la sección main del HTML.
2. Procesar los datos obtenidos de la API para organizarlos en cards, aplicando Flexbox o Grid para mantener la coherencia en el diseño.

## Carrito de compras

1. Debemos tener implementado un carrito de compras que permita a los usuarios añadir productos desde las cards, utilizando `localStorage` y `sessionStorage` para almacenar la información del carrito.
2. Los productos en el carrito se deben poder visualizar, editar (cambiar la cantidad) y eliminar.
3. La información debe mantenerse después de recargar la página.





# Proyecto Integrador



Obligatorio | Entregable

## Proyecto Integrador Final

- Las páginas deben estar estructuradas correctamente utilizando etiquetas semánticas.
- Implementar al menos una animación CSS para mejorar la experiencia del usuario.
- Utilizar fuentes personalizadas de Google Fonts.
- El CSS debe estar embebido, aplicando estilos que incluyan el uso de Flexbox o Grid para un diseño responsivo.
- Incluir un navbar creado con Flexbox.
- Consumir una API REST mediante fetch.
- Mostrar elementos dinámicos en el HTML a partir de datos JSON.
- Utilizar la manipulación del DOM para integrar los datos obtenidos en la interfaz.
- El proyecto debe estar subido a GitHub con un historial de commits que documenten el progreso.