

Práctica 1: Desarrollo Multiplataforma - Lista de Tareas

Descripción del Proyecto

Esta práctica consiste en el desarrollo de una aplicación web multiplataforma básica utilizando HTML, CSS y JavaScript. El objetivo es crear una interfaz de usuario simple que permita gestionar una lista de tareas, incluyendo agregar nuevas tareas, marcarlas como completadas, eliminarlas y almacenar los datos localmente en el navegador.

Objetivos

- Aprender los fundamentos del desarrollo web multiplataforma.
- Implementar una interfaz de usuario responsive y moderna.
- Integrar funcionalidades básicas como manipulación del DOM, eventos y almacenamiento local (localStorage).
- Aplicar buenas prácticas de desarrollo frontend, incluyendo separación de responsabilidades (HTML, CSS, JS).

Tecnologías Utilizadas

- *Lenguaje de Programación:* JavaScript (ES6+).
- *Tecnologías Web:* HTML5, CSS3.
- *Herramientas:* Visual Studio Code, Navegador web (Chrome, Firefox, etc.).
- *Dependencias:* Ninguna externa; utiliza APIs nativas del navegador.

Estructura del Proyecto

practica1/

— index.html	# Estructura principal de la aplicación
— index.js	# Lógica de la aplicación (manejo de tareas)
— styles.css	# Estilos CSS para la interfaz

└─ docs/

└─ asd.md

Este documento de entregable

Funcionalidades Implementadas

- *Agregar Tareas*: Permite ingresar una nueva tarea mediante un campo de texto y un botón.
- *Marcar como Completada*: Cada tarea puede ser marcada o desmarcada como completada, con visualización diferenciada (tachado).
- *Eliminar Tareas*: Opción para eliminar tareas individuales.
- *Almacenamiento Persistente*: Las tareas se guardan en localStorage, persistiendo entre sesiones del navegador.
- *Interfaz Responsive*: Diseño adaptable a diferentes tamaños de pantalla, con estilos modernos y transiciones suaves.
- *Validación Básica*: Verifica que la descripción de la tarea no esté vacía antes de agregarla.

Instrucciones de Instalación y Ejecución

1. Clona o descarga el proyecto en tu máquina local.
2. Abre el archivo index.html en un navegador web moderno (recomendado: Chrome o Firefox).
3. La aplicación se carga automáticamente; no requiere servidor ni instalación adicional.
4. Para desarrollo: Abre los archivos en un editor como VS Code y modifica según sea necesario.

Inconvenientes

El principal desafío que presento este proyecto fue guardar las diferentes tareas en localStorage, haciendo que estas sean persistentes, ya que en un principio se utilizaron listas para guardar estas tareas.

Resultados y Conclusiones

La aplicación funciona correctamente en navegadores modernos, proporcionando una experiencia de usuario fluida y responsive. Se logró

implementar todas las funcionalidades básicas de una lista de tareas con un diseño atractivo. Para futuras mejoras, se podría integrar sincronización con una API externa, notificaciones push o soporte para categorías de tareas.