|  |  |
| --- | --- |
| Una imagen de una carretera de curvas con árboles  LISTA DE TAREAS  Lista de Tareas para ir creándola | Descripción breve  Creación de lista de tareas que se acepten rechacen etc.  PabloSG  2ºDAW-DWEC |

**Contenido**

[1. Descripción del proyecto 2](#_Toc210898276)

[2. Aplicación de elementos de JavaScript 2](#_Toc210898277)

[3. Justificación del diseño visual 2](#_Toc210898278)

[4. Organización del trabajo en equipo 2](#_Toc210898279)

[5. Repositorio en GitHub 3](#_Toc210898280)

[6. Uso de Jira 3](#_Toc210898281)

[7. Preparación de la presentación 3](#_Toc210898282)

# 1. Descripción del proyecto

Este proyecto consiste en una aplicación web desarrollada en JavaScript que permite al usuario gestionar una lista de tareas pendientes. Se pueden añadir, marcar como completadas, eliminar, y visualizar el porcentaje de tareas finalizadas. Además, se han aplicado criterios de diseño web para diferenciar visualmente las tareas completadas de las pendientes.

# 2. Aplicación de elementos de JavaScript

A continuación, se detallan los elementos del lenguaje utilizados y cómo se han implementado en el proyecto:

* **Variables (let, const):** Se han utilizado para declarar tanto los datos dinámicos (como la lista de tareas), como referencias a elementos del DOM.
* **Operadores:** Se han empleado operadores lógicos (!, &&) y aritméticos para el control de flujo y el cálculo del porcentaje completado.
* **Condicionales (if):** Usados para validar la entrada del usuario y aplicar estilos según el estado de la tarea.
* **Bucles (forEach):** Utilizados para recorrer la lista de tareas y renderizarlas en el DOM.
* **Funciones:** Toda la lógica está estructurada en funciones como addTask(), deleteTask(), toggleTask() y updateStats(), lo que facilita su comprensión y mantenimiento.
* **Conversión de tipos:** Se aplicó Math.round() para redondear el porcentaje de tareas completadas y evitar decimales innecesarios.
* **Comentarios en español:** El código contiene comentarios explicativos en español para facilitar la comprensión por parte de todo el grupo.

# 3. Justificación del diseño visual

Se ha optado por un diseño minimalista y funcional. Los colores y tipografías fueron seleccionados con base en criterios de accesibilidad y legibilidad:

* **Colores:**
  + Verde para tareas completadas, transmitiendo éxito y finalización.
  + Rojo para tareas pendientes, llamando la atención sobre acciones por hacer.
  + Fondo claro para facilitar la lectura general.
* **Tipografía:** Se ha utilizado Arial, una fuente sans-serif ampliamente legible y comúnmente aceptada en diseño web.
* **Diseño responsivo:** La interfaz se adapta correctamente a distintos tamaños de pantalla, permitiendo su uso tanto en ordenadores como en dispositivos móviles.

# 4. Organización del trabajo en equipo

1. Análisis de requisitos y diseño de estructura HTML.
2. Implementación de funcionalidades básicas en JavaScript.
3. Aplicación de estilos visuales con CSS.
4. Pruebas y depuración.
5. Documentación del proyecto y preparación de la entrega.

**Dificultades encontradas:**

* Control de estados de las tareas (hecha/pendiente).
* Actualización dinámica del porcentaje de tareas completadas.

**Soluciones:**

* Separación de lógica en funciones reutilizables.
* Uso correcto del DOM para modificar la interfaz de forma dinámica.

# 5. Repositorio en GitHub

El proyecto ha sido gestionado con control de versiones mediante Git y alojado en GitHub:

Enlace al repositorio:

En el repositorio se puede observar:

* Estructura organizada del proyecto.
* Historial de commits con descripciones claras.
* Versiones progresivas de la aplicación.

# 6. Uso de Jira

Cada funcionalidad se organizó como una tarea dentro de un tablero Kanban.

Capturas adjuntas:

* Planificación del flujo de trabajo.
* Tareas completadas y pendientes.
* Seguimiento de progreso.

# 7. Preparación de la presentación

Se ha preparado una breve presentación para exponer en clase, con el siguiente contenido:

* Demostración de la funcionalidad de la aplicación.
* Explicación del código JavaScript y los aspectos más relevantes.
* Principios de diseño aplicados.
* Herramientas utilizadas durante el desarrollo (editor, navegador, GitHub).