

# Informe Parcial 1 – Gestor de Tareas

## 1. Nombre del proyecto

Gestor de Tareas

## 2. Objetivo

Desarrollar una aplicación web que permita a un usuario crear, visualizar, completar y eliminar tareas personales de forma dinámica y persistente, cumpliendo con los requerimientos mínimos de la materia y agregando algunas mejoras de interacción y usabilidad.

## 3. Descripción de la página

La página se compone de los siguientes elementos:

- **Encabezado:**
  - Título principal: “*Gestor de Tareas*”. Se encuentra centrado y da claridad al usuario sobre la funcionalidad de la página.
- Formulario de ingreso de tareas:
  - Campos: Nombre (obligatorio, máximo 50 caracteres), Descripción (obligatorio, máximo 50 caracteres).
  - Botón Agregar tarea: En negrita y con color principal para destacarlo.
  - Mensaje de error dinámico: Se muestra debajo del formulario si los campos están vacíos o superan los 50 caracteres. Desaparece automáticamente al escribir en cualquiera de los campos.
- **Tabla de tareas:**
  - Columnas: Nombre, Descripción, Acciones.
  - Cada fila representa una tarea.
  - Fila completada: Se diferencia visualmente con un color de fondo más suave y el texto en color gris.
  - Acciones:
    - Icono “tilde” verde para marcar como completada.
    - Icono “flecha amarilla hacia la izquierda” para deshacer la tarea completada.
    - Icono “cruz roja” para eliminar la tarea (con confirmación).
- **Footer:**
  - Muestra la marca de agua “© 2025 Gestor de Tareas” centrada.

## 4. Flujo de uso

1. El usuario ingresa el nombre y la descripción de la tarea en los campos correspondientes.
2. Al presionar el botón Agregar tarea, la aplicación:
  - Valida que los campos no estén vacíos.
  - Valida que los campos no superen los 50 caracteres.
  - Si alguna validación falla, muestra un mensaje de error.
  - Si todo es correcto, agrega la tarea a la lista, la muestra en la tabla y la guarda en localStorage.
3. En la tabla, el usuario puede:
  - Marcar como completada: la fila cambia de color y el icono se reemplaza por el de deshacer.
  - Deshacer completada: vuelve al estado inicial.
  - Eliminar: muestra un cuadro de confirmación antes de borrar la tarea.
4. Todas las acciones se reflejan dinámicamente en la interfaz y se mantienen aunque el usuario recargue la página, gracias a localStorage.

## 5. Tecnologías y herramientas utilizadas

- HTML5: Estructura de la página, formulario y tabla.
- CSS3: Estilos, colores, espaciados, hover, iconos SVG y diseño responsivo.
- JavaScript (ES6):
  - Variables (let, const), condicionales, ciclos, arrays y objetos.
  - Interacción con el DOM para mostrar, actualizar y eliminar tareas.
  - Funciones para agregar, eliminar y marcar tareas como completadas.
  - Uso de localStorage para persistencia de datos.
  - Validación de campos y manejo de errores.
- SVG: Iconos de acciones (completar, deshacer, eliminar).

## 6. Funcionalidades adicionales implementadas

- Confirmación al eliminar una tarea.
- Mensaje de error que desaparece automáticamente al escribir.
- Distinción visual clara entre tareas completadas y pendientes.
- Estética mejorada con paleta de colores, espaciados y hover en filas e iconos.

## 7. Flujo de datos

- Al cargar la página: se leen las tareas guardadas en localStorage y se muestran en la tabla.
- Al agregar, eliminar o completar/deshacer una tarea:
  1. Se modifica el array de tareas en memoria.
  2. Se guarda en localStorage.
  3. Se actualiza la tabla en pantalla.