

# Documentación Código JavaScript

## Funcionalidad:

- **Gestión de Tareas:** Permite a los usuarios crear, administrar, editar, eliminar y reorganizar tareas dentro de columnas.
- **Arrastrar y Soltar:** Facilita la organización intuitiva de tareas al permitir arrastrar y soltar entre columnas y tareas.
- **Edición de Tareas:** Los usuarios pueden modificar directamente las tareas existentes haciendo clic en el botón de edición e ingresando texto.
- **Eliminación de Tareas:** Las tareas se pueden eliminar haciendo clic en el botón de eliminar y confirmando la acción mediante un modal.
- **Reordenación de Tareas:** Las tareas se pueden reorganizar dentro de una columna o entre columnas arrastrándolas y soltándolas.
- **Visualización del Conteo de Tareas:** El número de tareas en cada columna se actualiza dinámicamente para reflejar los cambios.
- **Tareas Predeterminadas:** Se cargan tareas predefinidas al inicio para ofrecer un punto de partida a los usuarios.

## Estructura del Código:

### Declaraciones de Variables:

`modal`: Hace referencia al elemento modal de confirmación.

`columnsContainer`: Apunta al contenedor que contiene todas las columnas de tareas.

`columns`: Un NodeList que contiene elementos individuales de columna de tareas.

`currentTask`: Almacena una referencia a la tarea seleccionada actualmente (utilizada para la confirmación de eliminación).

### Definiciones de Funciones:

`handleDragover` (event): Maneja eventos de arrastre sobre los contenedores de tareas, evitando el comportamiento predeterminado y posicionando correctamente las tareas arrastradas.

**handleDrop** (event): Controlador de eventos de soltar, actualmente vacío ya que no realiza ninguna acción específica.

**handleDragend** (event): Elimina la clase "dragging" de la tarea arrastrada cuando finaliza la operación de arrastre.

**handleDragstart** (event): Configura los efectos de transferencia de datos y agrega la clase "dragging" a la tarea arrastrada.

**handleDelete** (event): Captura la tarea seleccionada, establece su contenido como texto de vista previa en el modal de confirmación de eliminación y muestra el modal.

**handleEdit** (event): Reemplaza la tarea seleccionada con un campo de entrada editable, permitiendo la edición en línea.

**handleBlur** (event): Se activa cuando el campo de entrada pierde el foco. Recorta su contenido, crea un nuevo elemento de tarea y reemplaza el campo de entrada con la nueva tarea.

**handleAdd** (event): Identifica el contenedor de tareas de la columna seleccionada y le agrega un nuevo campo de entrada editable.

**updateTaskCount** (column): Actualiza el conteo de tareas mostrado en el título de una columna según el número actual de tareas hijas.

**observeTaskChanges**: Itera a través de las columnas, creando MutationObservers para monitorear los cambios en la lista de tareas y llamar a **updateTaskCount** cada vez que se detecta un cambio.

**createTask** (content): Crea y devuelve un nuevo elemento de tarea con el contenido especificado, comportamiento de arrastrable y botones de editar/eliminar.

**createTaskInput** (text): Crea y devuelve un campo de entrada editable con un marcador de posición y texto inicial opcional.

## 1. Escuchadores de Eventos:

### Arrastrar y Soltar:

Adjunta escuchadores de eventos **dragover** y **drop** a todos los elementos de contenedor de tareas.

### Acciones de Tarea:

Adjunta un escuchador de eventos de clic al `columnsContainer`.

Dentro del escuchador:

Identifica el elemento seleccionado para manejar las acciones de agregar, editar o eliminar según los atributos de datos.

### Confirmación Modal:

Adjunta un escuchador de eventos `submit` al formulario modal para eliminar la tarea seleccionada si se confirma.

Adjunta un escuchador de eventos de clic al botón de cancelar del modal para cerrarlo sin eliminar.

Adjunta un escuchador de eventos `close` al modal para borrar la referencia `currentTask` cuando se cierra el modal.

### Tareas Predeterminadas:

Define una matriz de matrices (`tasks`), representando tareas pertenecientes a diferentes columnas (Por Hacer, En Progreso, Pendiente, Hecho).

Itera a través de `tasks`, creando elementos de tarea individuales usando `createTask` y adjuntándolos al contenedor de tareas de la columna correspondiente.