




# Proyecto Final

**DEV.F.**  
DESARROLLAMOS(PERSONAS);



Elaborado por: Jesua Luján  
 [Jesua Hadai Luján](#)

dev












| 🧠 Proyecto Final: **Gestor de Tareas Colaborativo.**

## 🎯 Objetivo del proyecto

Construir una **aplicación web tipo trello** que permita crear, editar, eliminar y organizar tareas por estado y responsable. El punto principal estará en el uso de **JavaScript moderno, manejo del DOM, consumo de APIs simuladas**, arquitectura del código y lógica de negocio.

**Planning** ...






  5  3  





  2  

  2/3  4  

+ Add a card

**Delivery** ...






  2/3  4  

  4  July 15 

  2    4  4  

+ Add a card

**Evaluation and learning** ...

  1  2/2  Jun 6 

  4  Jun 12     2  4/4  

+ Add a card



### To Do

Create offer notification email

3

Compile video animation alongside blog articles

1

Structure editorial calendar

1

### Research

Check all internal URLs work in blog archive

Feb 26 2

Create e-mail newsletter for blog subscribers

1

Launch podcast

2

Prepare onboarding materials for new UI designers

2

### In Progress

Offer notification email

1

Email to last week's conference attendees

Feb 20 2

Update social media content

2 2/3

Brainstorm & suggest 10x blog post ideas

3

Research weekly blog post and suggest images

3 6

### Review

Create 5x FB ads

2

Sharpen copy across homepage

7 2

Create infographics for weekly blog post

Feb 12 4

Snapchat Story on upcoming post

1

Prompt comments on blog posts

1

### Completed

Redesign About Us page

Feb 10 5

Redesign logo

4 2 7/7

Update iOS app

1

Source images for blog post

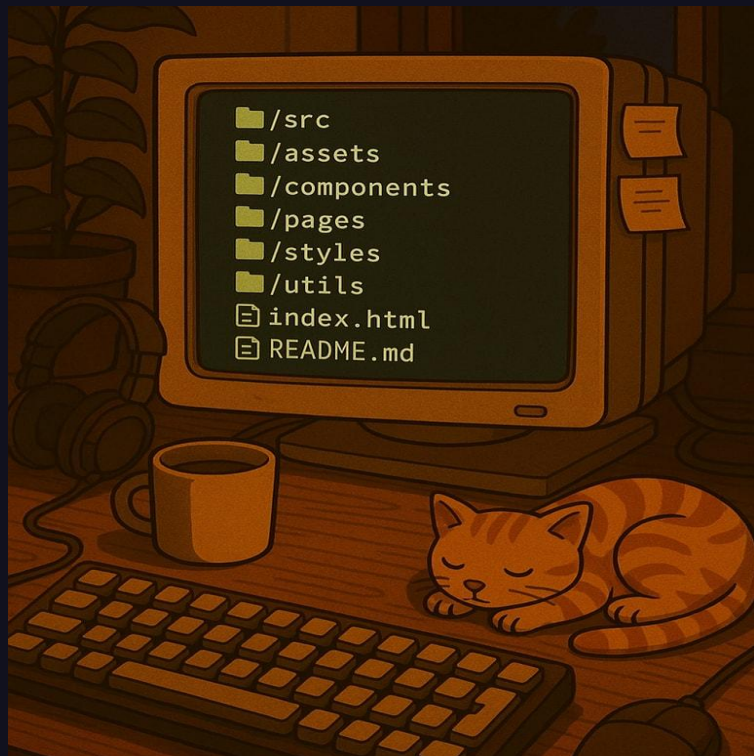
1

Create logo for Medium

7 2



# | CRUD



## Estructura de archivos sugerida:

1. gestor-tareas/
2. |— index.html
3. |— style.css
4. |— app.js
5. |— db.json
6. |— README.md




## I Descripción rápida

Archivo	Propósito
index.html	Estructura HTML básica: contenedor de tareas y formulario.
style.css	Estilos simples para mostrar tareas en columnas.
app.js	Toda la lógica: eventos, fetch, renderizado y manipulación del DOM.
db.json	Base de datos simulada para usar con json-server.
README. md	Instrucciones para ejecutar el proyecto.



## I Características mínimas

-  Tareas:
- CRUD completo de tareas.
- Cada tarea debe tener:
  - Título
  - Descripción
  - Estado (pendiente, en progreso, terminada)
  - Responsable (puede ser cualquier nombre escrito a mano)
- Listado de tareas agrupadas por estado.
- Posibilidad de filtrar tareas por responsable o estado.





## | 🧠 Lógica de negocio 🔥

- Validaciones al crear/editar tareas (**campos obligatorios, longitud mínima, etc.**).
- Confirmación al eliminar.
- Guardado y recuperación de datos desde una API simulada (**json-server**).
- Feedback visual al usuario (**mensajes de éxito/error, etc.**).





# I 🏗️ Organización del código 🔥

Todo el código JavaScript va en un solo archivo: **app.js**.

Separa la lógica en secciones con comentarios claros:

- 🛰️ **Comunicación con la API (fetch)**
- 🧠 **Lógica de tareas (crear, leer, renderizar)**
- 🎯 **Eventos del formulario**

Nada de **onclick** ni **onsubmit** en el **HTML**. Todo se maneja con event listeners desde el Javascript.

**Usá nombres de funciones y variables que expliquen bien su propósito.**

Comenta tu código para que sea fácil de entender por otros (buenas prácticas!).

**DEV.E.:**





## Bonus (opcional pero recomendado)



DEV.F.



- Drag & drop entre columnas de estado.
- Contador de tareas por estado.
- Modo oscuro / claro.
- Animaciones sutiles al agregar o mover tareas.
- Etiquetas por prioridad (alta, media, baja).
- Deploy en GitHub Pages.



Repositorio en GitHub con:

- Código fuente bien organizado.
- README.md con:
  - Descripción del proyecto
  - Instrucciones para levantar el proyecto con json-server

Demo funcional (en local o deployado).



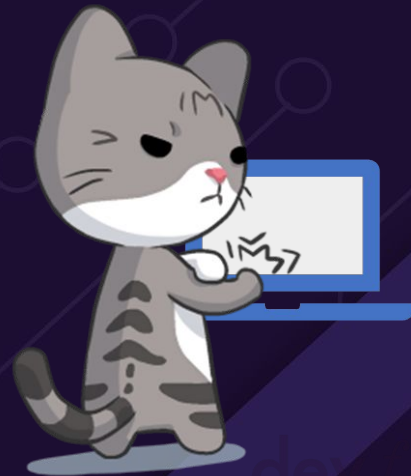
La fecha para entregar el proyecto y presentarlo en clases es la siguiente.

Fecha de entrega: **Jueves 29 de mayo del 2025.** 🔥



# Implementación

**DEV.F**  
DESARROLLAMOS(PERSONAS);



## I 🚀 Sugerencia de flujo de trabajo

### scripts/app.js

- Funciones de fetch:
  - getTareas()
  - crearTarea(tarea)
  - actualizarTarea(id, datos)
  - eliminarTarea(id)

### Renderizado del DOM:

- renderTareasPorEstado(tareas)
- crearElementoTarea(tarea)
- mostrarFormulario(),  
limpiarFormulario(), etc.



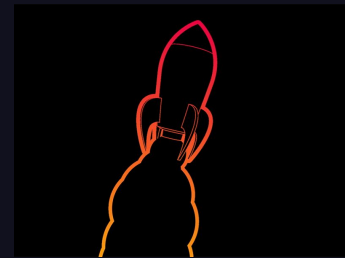
| db.json



DEV.F.

```
{  
  "tareas": [  
    {  
      "id": 1,  
      "titulo": "Planificar estructura",  
      "estado": "pendiente",  
      "responsable": "Luján"  
    }  
  ]  
}
```

Se te proveerá un json  
de 15 tareas en el canal  
de discord para que lo  
copies y lo pegues en  
**db.json**





## I Paso a paso para comenzar

1. Crear la estructura de carpetas y archivos.
2. Escribir un index.html base con los contenedores de tareas.
3. Correr json-server:
  - `npx json-server --watch db.json --port 3000`
  - si lo deseas puedes elegir otro puerto.

