

Axel Alvarez

Angel Rodríguez

Sistema Integral de Gestión de Tareas en Tiempo Real

Es una aplicación web que permite a los usuarios gestionar sus tareas de forma eficiente, con soporte para operaciones CRUD (Crear, Leer, Actualizar, Eliminar) y notificaciones en tiempo real gracias a WebSocket.

El sistema fue diseñado con una arquitectura cliente-servidor moderna, donde el frontend se comunica con un API RESTful desarrollado en Node.js, mientras que las actualizaciones de tareas se sincronizan en tiempo real usando Socket.IO.

Cada tarea queda registrada en una base de datos persistente (SQL Server o MongoDB), permitiendo la consulta y modificación desde cualquier dispositivo con conexión. Además, se incluye autenticación básica mediante JSON Web Tokens (JWT) para proteger las rutas críticas del sistema.

El sistema fue desplegado en un servidor remoto usando la plataforma **Render** y está completamente versionado mediante **GitHub** para garantizar trazabilidad y colaboración.

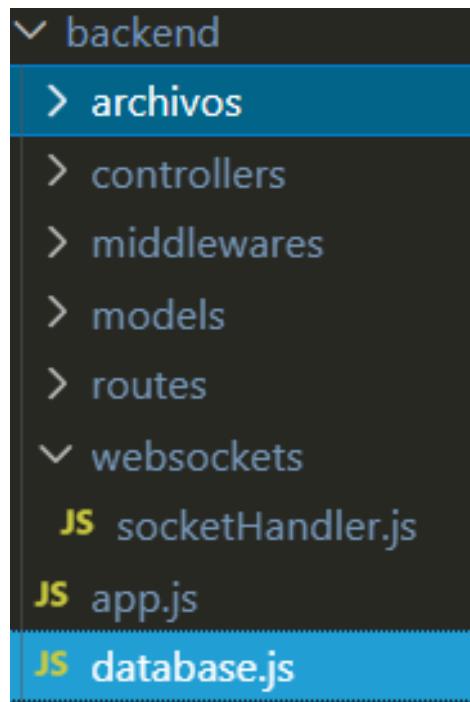
Tecnologías Utilizadas - Backend

El backend del proyecto fue construido utilizando las siguientes tecnologías:

- **Node.js:** Plataforma de desarrollo basada en JavaScript para construir aplicaciones del lado del servidor. Es eficiente, escalable y perfecto para aplicaciones en tiempo real.
- **Express.js:** Framework minimalista para Node.js que facilita la creación de servidores web y APIs RESTful de manera sencilla y estructurada.
- **Socket.IO:** Librería que permite la comunicación bidireccional y en tiempo real entre el servidor y los clientes. Se utilizó para emitir eventos cada vez que se crea, edita o elimina una tarea.
- **Base de Datos (MSSQL o MongoDB):**
 - **MSSQL:** Sistema de gestión de bases de datos relacional utilizado para persistir las tareas, usuarios y demás entidades. Se conectó al backend mediante la librería mssql.
 - (Alternativamente, el sistema puede usar MongoDB si se desea un enfoque NoSQL.)

- **JWT (JSON Web Tokens)**: Sistema de autenticación basado en tokens que protege las rutas del API. Solo los usuarios autenticados pueden realizar operaciones sobre las tareas.
- **CORS y Middlewares**: Para permitir el acceso seguro desde diferentes dominios y procesar peticiones JSON.
- **Render**: Plataforma de alojamiento donde se desplegó la aplicación backend. Permite ejecutar servicios Node.js con conexión a bases de datos remotas.
- **GitHub**: Herramienta de control de versiones donde se alojó todo el código fuente del proyecto. Permite llevar un seguimiento del desarrollo y facilita el trabajo colaborativo.

Estructura del proyecto



/project-root/backend

Es el directorio principal del backend. Dentro de él están organizados los componentes clave de la aplicación del lado del servidor.

/controllers

Responsabilidad: Contiene la lógica del negocio. Aquí se procesan las peticiones entrantes, se aplican las reglas del sistema y se comunican con los modelos para acceder a los datos.

/models

Responsabilidad: Contiene los modelos de datos. Se encargan de las consultas a la base de datos y representan la estructura de las entidades (como Tarea, Usuario, etc.).

/routes

Responsabilidad: Contiene la definición de las rutas del API RESTful. Aquí se definen las rutas HTTP (GET, POST, PUT, DELETE) y se enlazan con los controladores correspondientes.

/websockets

Responsabilidad: Manejo de la comunicación en tiempo real con Socket.IO. Contiene la lógica para emitir y recibir eventos de WebSocket, como notificar cuando una tarea se crea, edita o elimina en tiempo real.

socketHandler.js

Este archivo exporta una función que recibe el objeto io (de Socket.IO) y define todos los eventos de WebSocket que el servidor manejará.

app.js

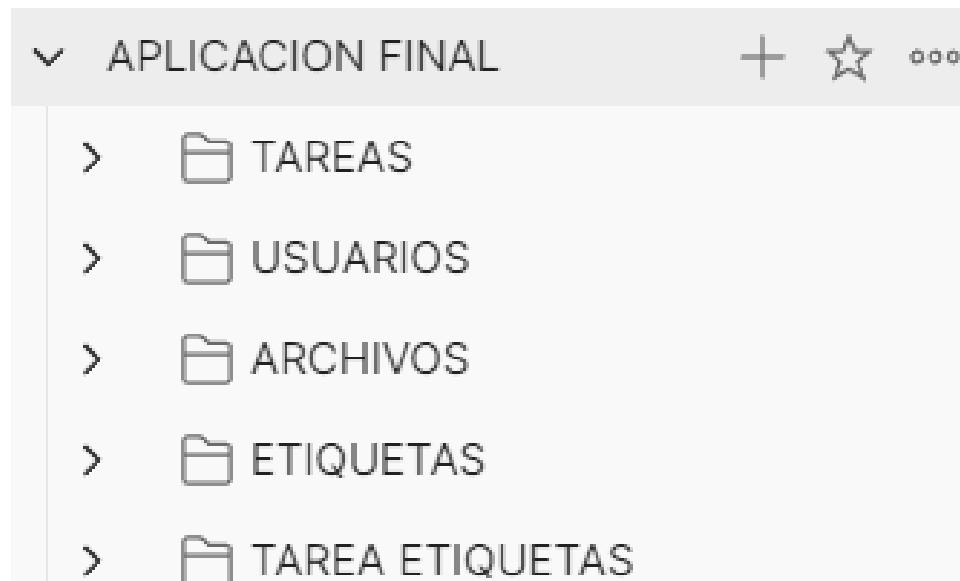
Responsabilidad: Es el archivo principal de configuración de la aplicación Express.

Aquí se inicializa el servidor, se importan las rutas, se aplica middleware, se conecta WebSocket y se inicia el servidor.

database.js

Responsabilidad: Configura y establece la conexión con la base de datos (por ejemplo SQL Server o MongoDB).

Consumo de API en postman



Login

The screenshot shows a POST request to "http://localhost:3000/api/usuarios/login". The "Body" tab is selected, showing the raw JSON payload:

```
1 {  
2   "correo": "admin@gmail.com",  
3   "contrasena": "123456",  
4   "obtenerToken": true  
5 }
```

The response status is 200 OK, with a timestamp of 1.27 s and a size of 45. The "Body" tab shows the JSON response:

```
1 {  
2   "mensaje": "Login exitoso",  
3   "token": "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.  
eyJpZCI6MTAsIm5vbWJyZSI6ImFkbwluZGQ1LCyb2xfawQi0jEsImlhdcI6MTc1MTY0MTI3MSwiZXhwIjoxNjQ0ODcxMjQ.  
E11qj6vebh-5xbzMNDuB0l3XbgTu79WnUXmbK0FqgAg",  
4   "rol_id": 1  
5 }
```

Registro

The screenshot shows a POST request to `http://localhost:3000/api/usuarios/register`. The request body is a JSON object:

```
1  {
2    "nombre": "Axedal",
3    "correo": "axelddaa@example.com",
4    "contrasena": "123456",
5    "rol_id": 2
6 }
```

The response body is also a JSON object:

```
{ } JSON ▾
```

```
1  {
2    "mensaje": "Usuario registrado correctamente"
3 }
```

Solicitudes

Se necesita un token para poder realizar las solicitudes get, post, put y delete como se muestra en esta solicitud para ver usuarios

The screenshot shows a GET request to `http://localhost:3000/api/usuarios`. The Headers tab is selected, showing the following configuration:

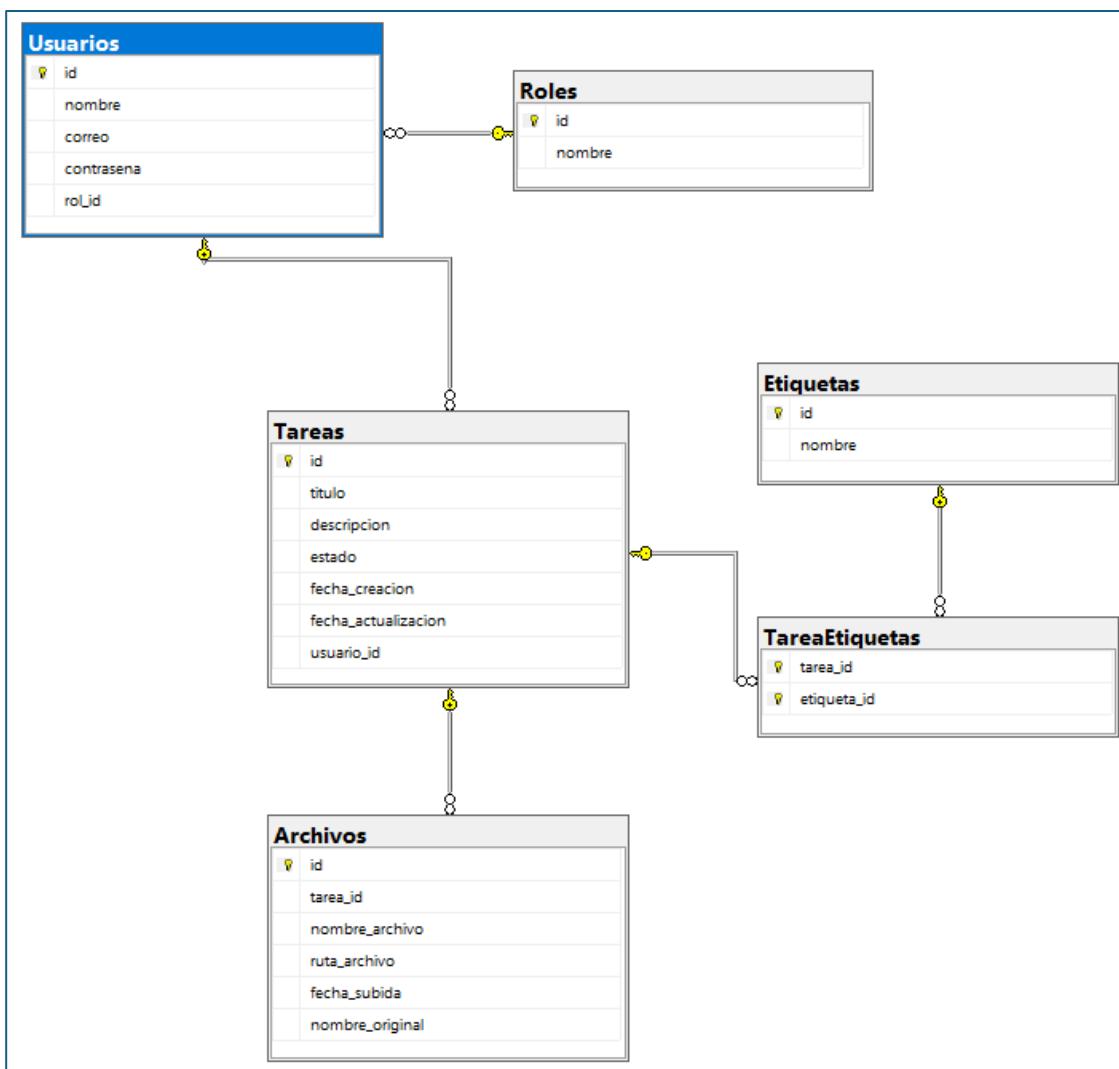
Key	Value
Authorization	Bearer eyJhbGciOiJIUzI1NilsInR5cCI6IkpxVCJ9.eyJpZCI...
Key	Value

```

        "id": 6,
        "nombre": "pedrooo",
        "correo": "pedro@gmail.com",
        "contrasena": "$2b$10$5a79Lwdi6nXKbD5al5pXf0Ijmdrjs86W8zZijAK/.6dXYNsGITaz2",
        "rol_id": 2
    },
    {
        "id": 8,
        "nombre": "Isel",
        "correo": "iselaa@gmail.com",
        "contrasena": "$2b$10$TN7CZ00oKqKS2KAHQE3F4.diFVyi872SVLokUIrGcT91gIt1pBUTu",
        "rol_id": 2
    },
    {
        "id": 11,
        "nombre": "lupillo",
        "correo": "lupillo502@gmail.com",
        "contrasena": "$2b$10$QnG2qTDUt4ANSfG92muC4eRxrxatgXPpCC3hXUMHxzDVQvuKnbhG2",
        "rol_id": 2
    },
    {
        "id": 18,
        "nombre": "Angellll",
        "correo": "angel@gmail.com",
        "contrasena": "123456",
        "rol_id": 2
    }
].

```

Diagrama UML Base de datos



1. Descripción General

El **Sistema de Tareas** es una plataforma web diseñada para gestionar tareas, usuarios, archivos y etiquetas de forma organizada y colaborativa. Está estructurada para ofrecer roles diferenciados (**Administrador** y **Usuario**) y se ha construido utilizando **Tailwind CSS**, lo que permite un diseño minimalista, responsive y coherente.

2. Estructura de Vistas y Funcionalidades

1. Landing Page

- **Descripción:** Pantalla de bienvenida con nombre del sistema (**GestorPRO**), mensaje principal, llamada a la acción y navegación para iniciar sesión o registrarse.
- **Funcionalidad:** Invita al usuario a registrarse o iniciar sesión. Diseño limpio con onda decorativa y footer.
- **Tailwind CSS:** Utiliza flex, text-center, bg-gradient, botones con hover y tipografía legible.



2. Login

- **Descripción:** Formulario para autenticación de usuarios.
- **Funcionalidad:** Campos para correo y contraseña, botón de acceso y enlace para registro si no se tiene cuenta.
- **Tailwind CSS:**
Formulario centrado con sombra (shadow-lg), bordes redondeados (rounded-lg), focus:outline para campos y botón principal estilizado.

Gestión de Tareas

Iniciar sesión Registrarse

Inicia sesión en tu cuenta

Correo Electrónico

Contraseña

Entrar

¿No tienes una cuenta? [Regístrate aquí](#)

3. Registro

- **Descripción:** Formulario para crear una nueva cuenta.
- **Funcionalidad:**
Campos: nombre, correo y contraseña.
Botón de registro y enlace para volver al login.
- **Tailwind CSS:** Mismo estilo del login para coherencia visual. bg-white, shadow, botones hover y rounded.

Gestión de Tareas

Iniciar sesión Registrarse

Regístrate en la plataforma

Nombre

Correo electrónico

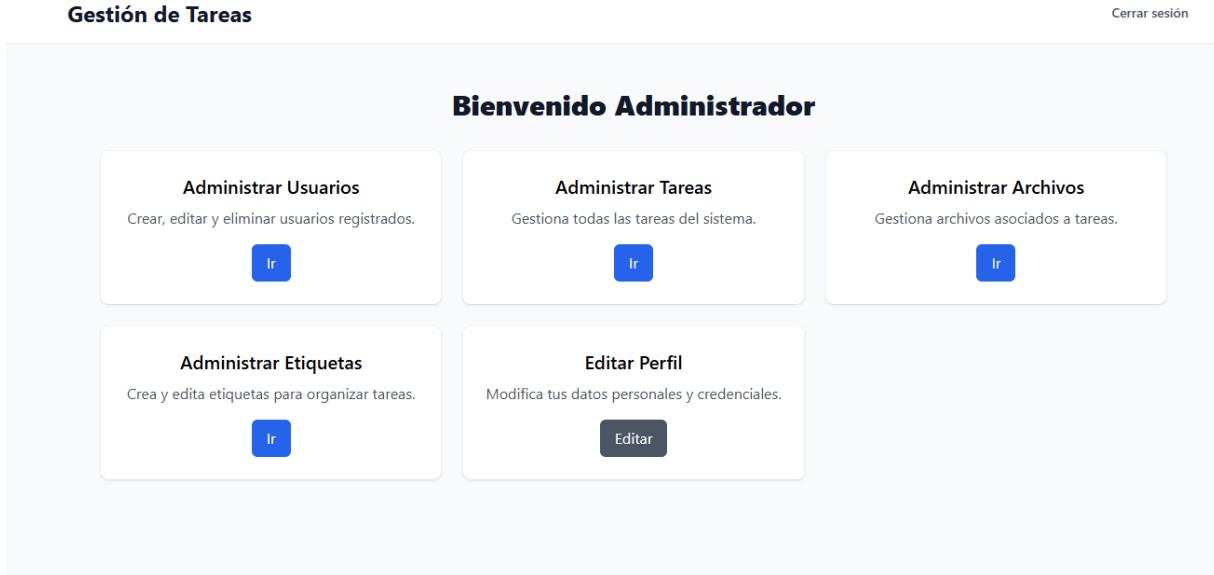
Contraseña

Registrarse

¿Ya tienes una cuenta? [Inicia sesión aquí](#)

4. Panel Principal Administrador

- **Descripción:** Dashboard del **Administrador** con acceso rápido.
- **Funcionalidad:** Tarjetas para:
 - **Administrar Usuarios**
 - **Administrar Tareas**
 - **Administrar Archivos**
 - **Administrar Etiquetas**
 - **Editar Perfil**
- **Tailwind CSS:** grid para disposición de tarjetas, shadow, botones coloridos bg-blue-500, hover:bg-blue-600.



5. Administración de Usuarios

- **Descripción:** Tabla de usuarios con sus datos.
- **Funcionalidad:**
 - **Agregar Usuario**
 - **Editar datos**
 - **Eliminar usuario**
- **Tailwind CSS:** Tabla estilizada table-auto, botones bg-blue y bg-red, bordes y hover para resaltar acciones.

Administrar Usuarios						
ID	NOMBRE	CORREO	ROL ID	EDITAR	ELIMINAR	
2	Angel Rodriguez	angel23@gmail.com	2	<button>Editar</button>	<button>Eliminar</button>	
3	Juan Perez	prueba@gmail.com	2	<button>Editar</button>	<button>Eliminar</button>	
4	Mariana Saez	prueba1@gmail.com	2	<button>Editar</button>	<button>Eliminar</button>	
5	Rosa Alvarez	rosa@gmail.com	2	<button>Editar</button>	<button>Eliminar</button>	
6	Diego Armando	diego@gmail.com	2	<button>Editar</button>	<button>Eliminar</button>	

6. Crear Usuario

- Descripción:** Formulario para registrar un nuevo usuario.
- Funcionalidad:** Campos: nombre, correo y contraseña. Botones para crear o cancelar.
- Tailwind CSS:** Formulario vertical con shadow, botones diferenciados por color (green y gray).

Crear Usuario

Nombre

Correo

Contraseña

El usuario creado tendrá rol **Usuario** por defecto.

Crear Cancelar

7. Administración de Tareas (Admin)

- **Descripción:** Tabla para gestionar todas las tareas del sistema.
- **Funcionalidad:**
 - Ver ID, título, descripción, estado, fecha, usuario asignado y etiquetas.
 - Editar tarea, eliminar tarea, gestionar etiquetas.
- **Tailwind CSS:** Tabla responsive, colores para etiquetas, acciones resaltadas (bg-blue, bg-red, bg-yellow).

Gestión de Tareas								Cerrar sesión
Administrar Tareas								
<input type="text" value="Buscar por ID, Título, Descripción o Estado..."/>								
ID	TÍTULO	DESCRIPCIÓN	ESTADO	FECHA	USUARIO	ETIQUETA(S)	ACCIONES	ACCIONES ETIQUETAS
1	Terminar proyecto final	Terminar el frontd del proyecto.	Pendiente	3/7/2025	Angel Rodriguez (angel23@gmail.com)	Desarrollo	<button>Editar</button> <button>Eliminar</button>	<button>Editar Etiqueta</button>
2	Cambiar la bienvenida	Cambiar la pantalla de bienvenida por el nombre del usuario.	Cancelada	3/7/2025	Angel Rodriguez (angel23@gmail.com)	Entrega	<button>Editar</button> <button>Eliminar</button>	<button>Editar Etiqueta</button>
3	Comprar marcadores	Comprar marcadores para pizzarron color verde y azul	Pendiente	4/7/2025	Rosa Alvarez (rosa@gmail.com)	Trabajo	<button>Editar</button> <button>Eliminar</button>	<button>Editar Etiqueta</button>
4	Cambiar notas	Cambiar notas finales	En progreso	4/7/2025	Rosa Alvarez (rosa@gmail.com)	Corrección	<button>Editar</button> <button>Eliminar</button>	<button>Editar Etiqueta</button>

8. Administración de Archivos (Admin)

- **Descripción:** Tabla de archivos subidos al sistema.
- **Funcionalidad:**
 - Ver archivos con nombre, ruta y fecha.

Gestión de Tareas						Cerrar sesión
Lista de Archivos						
ID	ID TAREA	NOMBRE ARCHIVO	RUTA ARCHIVO	FECHA SUBIDA	ACCIONES	
1	1	Proyecto_Final del Curso.pdf	/archivos/Proyecto_Final del Curso-1751569698321-711340579.pdf	3/7/2025, 7:08:18 a. m.	<button>Editar</button> <button>Eliminar</button>	
2	2	Proyecto_Microservicio_WebSocket_Node_MSSQL.docx	/archivos/Proyecto_Microservicio_WebSocket_Node_MSSQL-1751641768201-799409490.docx	4/7/2025, 3:09:28 a. m.	<button>Editar</button> <button>Eliminar</button>	

- Editar o eliminar archivo.
- **Tailwind CSS:** Tabla overflow-x-auto, botones hover y tipografía clara.

9. Administración de Etiquetas

- **Descripción:** Tabla de etiquetas disponibles.
- **Funcionalidad:**
 - Agregar etiqueta
 - Editar o eliminar etiquetas existentes.
- **Tailwind CSS:** Botones verdes para agregar, azules para editar, rojos para eliminar. Estructura flex y grid.

ID	NOMBRE	ACCIONES
8	Corrección	<button>Editar</button> <button>Eliminar</button>
7	Desarrollo	<button>Editar</button> <button>Eliminar</button>
13	En Espera	<button>Editar</button> <button>Eliminar</button>
5	Entrega	<button>Editar</button> <button>Eliminar</button>
14	Finalizado	<button>Editar</button> <button>Eliminar</button>
1	Importante	<button>Editar</button> <button>Eliminar</button>
6	Investigación	<button>Editar</button> <button>Eliminar</button>
2	Personal	<button>Editar</button> <button>Eliminar</button>
9	Prioridad Alta	<button>Editar</button> <button>Eliminar</button>
11	Prioridad Baja	<button>Editar</button> <button>Eliminar</button>
10	Prioridad Media	<button>Editar</button> <button>Eliminar</button>
4	Reunión	<button>Editar</button> <button>Eliminar</button>
12	Revisión	<button>Editar</button> <button>Eliminar</button>
3	Trabajo	<button>Editar</button> <button>Eliminar</button>

10. Editar Perfil (Admin)

- **Descripción:** Formulario para actualizar datos del administrador.
- **Funcionalidad:** Editar nombre y correo.
- **Tailwind CSS:** Formulario simple y alineado, botón azul prominente.

Gestión de Tareas Cerrar sesión

Editar Perfil (Administrador)

Nombre

Correo

Guardar

11. Panel Principal Usuario

- **Descripción:** Dashboard de bienvenida personalizado.
- **Funcionalidad:**
 - Acceso a **Mis Tareas**
 - **Mis Archivos**
 - **Etiquetas Personales**
 - **Editar Perfil**
- **Tailwind CSS:** Estructura grid, botones con colores diferenciados (green para crear, gray para editar).

The screenshot shows a dashboard with a blue header bar containing the text "Sistema de Tareas" on the left and "Cerrar sesión" on the right. Below the header, the text "Bienvenido Angel Rodriguez" is displayed. The main area contains four cards arranged in a 2x2 grid:

- Mis Tareas**: Ver y administrar tus tareas personales. Includes a green "Ir" button.
- Administrar Etiquetas a mis tareas**: Agregar etiquetas a mis tareas. Includes a green "Crear" button.
- Mis Archivos**: Sube y administra tus archivos de tarea. Includes a green "Ir" button.
- Editar Perfil**: Actualiza tu información de cuenta. Includes a gray "Editar" button.

12. Mis Tareas (Usuario)

- **Descripción:** Tabla para gestión de tareas propias.
- **Funcionalidad:**
 - Crear tarea
 - Editar o eliminar tarea
 - Editar etiquetas de tarea
- **Tailwind CSS:** Tabla clara con hover, shadow sutil, acciones coloridas.

The screenshot shows a table titled "Mis Tareas" with a blue header bar containing the text "Sistema de Tareas" on the left and "Cerrar sesión" on the right. The table has columns: ID, Título, Descripción, Estado, Fecha, and Acciones. It contains two rows of data:

ID	Título	Descripción	Estado	Fecha	Acciones
1	Terminar proyecto final	Terminar el frontd del proyecto.	Pendiente	3/7/2025	<button>Editar</button> <button>Eliminar</button>
2	Cambiar la bienvenida	Cambiar la pantalla de bienvenida por el nombre del usuario.	Cancelada	3/7/2025	<button>Editar</button> <button>Eliminar</button>

13. Mis Archivos (Usuario)

- **Descripción:** Tabla para archivos del usuario.
- **Funcionalidad:**
 - Ver archivos relacionados a tareas propias.
 - Editar o eliminar.
- **Tailwind CSS:** Igual coherencia visual con tablas anteriores.

The screenshot shows a table titled 'Mis Archivos' (My Files). The table has columns: 'ID', 'Tarea ID', 'Nombre Archivo' (File Name), 'Ruta Archivo' (File Path), 'Fecha Subida' (Upload Date), and 'Acciones' (Actions). There are two rows of data. Each row includes a blue 'Editar' (Edit) button and a red 'Eliminar' (Delete) button.

ID	Tarea ID	Nombre Archivo	Ruta Archivo	Fecha Subida	Acciones
1	1	Proyecto_Final del Curso.pdf	/archivos/Proyecto_Final del Curso-1751569698321-711340579.pdf	3/7/2025	<button>Editar</button> <button>Eliminar</button>
2	2	Proyecto_Microservicio_WebSocket_Node_MSSQL.docx	/archivos/Proyecto_Microservicio_WebSocket_Node_MSSQL-1751641768201-799409490.docx	4/7/2025	<button>Editar</button> <button>Eliminar</button>

14. Agregar Archivo

- **Descripción:** Formulario para subir archivo relacionado a una tarea.
- **Funcionalidad:** Seleccionar tarea, subir archivo, guardar o cancelar.
- **Tailwind CSS:** Campos rounded, focus:outline, botones azul y gris.

The screenshot shows a form titled 'Agregar Archivo'. It has a dropdown menu labeled 'Tarea' with the placeholder 'Seleccione una tarea'. Below it is a file input field labeled 'Seleccionar archivo' with the placeholder 'Seleccionar archivo' and 'Sin archivos seleccionados'. At the bottom are two buttons: a blue 'Guardar' (Save) button and a grey 'Cancelar' (Cancel) button.

15. Editar Perfil (Usuario)

- **Descripción:** Formulario para actualizar datos personales.
- **Funcionalidad:** Editar nombre y correo, guardar cambios.
- **Tailwind CSS:** Diseño minimalista, centrado y limpio.

The screenshot shows a form titled 'Editar Perfil Usuario'. It has two input fields: 'Nombre' (Name) containing 'Angel Rodriguez' and 'Correo' (Email) containing 'angel23@gmail.com'. At the bottom is a blue 'Guardar Cambios' (Save Changes) button.

3. Uso de Tailwind CSS

Todo el proyecto utiliza **Tailwind CSS** para:

- Rapidez en prototipado de interfaces.
 - Diseño limpio, moderno y responsive.
 - Colores consistentes (bg-blue, bg-green, bg-red).
 - Tipografías claras (font-bold, text-xl).
 - Sombra y bordes redondeados (shadow-lg, rounded-lg).
 - Botones con hover para mejorar la interacción.
 - Estructura **flex** y **grid** para organizar tarjetas, formularios y tablas de forma modular.
-

4. Conclusión

Este frontend ofrece una experiencia **intuitiva, ordenada** y adaptable, gracias a la potencia de **Tailwind CSS** para mantener la coherencia visual y optimizar tiempos de desarrollo.