



Módulo 8 Actividad Final

Reto

Mini Sistema de Notas Personales con Categorías + API + Autenticación

Descripción general

El estudiante desarrollará una aplicación web donde un usuario puede:

- Registrarse e iniciar sesión (JWT).
- Crear notas personales sencillas.
- Asignar una **categoría** a cada nota:
 - Personal
 - Trabajo
 - Ideas
 - Recordatorios
- Editar y eliminar notas.
- Filtrar notas por categoría.
- Visualizar notas con diseño responsivo usando Tailwind.

Objetivos de aprendizaje evaluados

- Crear UI moderna con React + Vite + Tailwind.
- Manejar estado con useState y formularios controlados.
- Validaciones sencillas en formularios.
- CRUD conectado a API con Axios.
- Autenticación y rutas protegidas con token.
- Diseño modular de componentes.
- Uso asistido de IA para refactorización y generación de componentes.

Funcionalidades

1. Autenticación (login + registro)

Pantallas:

- /register
- /login

Debe permitir:



- ✓ Registrar usuario
- ✓ Iniciar sesión
- ✓ Manejo de errores (credenciales incorrectas)
- ✓ Guardar token en localStorage
- ✓ Botón Logout

2. CRUD de Notas

Pantalla /notes (ruta protegida)

Cada nota contiene:

- titulo
- contenido
- categoría (select con 4 opciones)
- fechaCreacion (lo entrega la API o el frontend)

Operaciones:

- ✓ Crear nota
- ✓ Listar notas
- ✓ Editar nota
- ✓ Eliminar nota

3. Filtrado por categoría

En la parte superior:

[Todas | Personal | Trabajo | Ideas | Recordatorios]

Al hacer clic, se filtran las notas por categoría.

4. UI básica con Tailwind

- Tarjetas para mostrar notas.
- Formulario con validaciones sencillas.
- Colores por categoría (borde o badge), por ejemplo:
 - Personal → azul
 - Trabajo → verde
 - Ideas → amarillo
 - Recordatorios → rojo



5. Rutas mínimas

/login
/register
/notes(protegida)

📁 Estructura recomendada del proyecto

```
src/
  └── api/
      └── axiosClient.js
  └── components/
      ├── NoteCard.jsx
      ├── NoteForm.jsx
      ├── FilterBar.jsx
      ├── Header.jsx
      └── Footer.jsx
  └── pages/
      ├── Login.jsx
      ├── Register.jsx
      └── Notes.jsx
  └── routes/
      └── ProtectedRoute.jsx
  └── App.jsx
  └── main.jsx
```

🤖 Uso de IA

El estudiante debe documentar:

- Qué componente creó o refactorizó con IA.
- Qué validación o lógica mejoró con ayuda de IA.
- Qué sugerencias aceptó o rechazó.

📦 Entregables del estudiante

1. Repositorio GitHub

Con código limpio y estructurado.

2. README profesional

Incluye:

- Cómo ejecutar el proyecto



- Rutas principales
- Capturas de la UI
- Explicación del uso de IA

Rúbrica

Criterio	Peso
Autenticación + rutas protegidas	25%
CRUD de notas funcional	30%
UI con Tailwind + responsividad	20%
Calidad de componentes + estado	15%
Uso de IA documentado	10%