

# Anteproyecto de Aplicación de Gestión de Asignaturas, Temas y Archivos

---

## 1. Introducción

El objetivo de este proyecto es el desarrollo de una aplicación web **Single Page Application (SPA)** que permita a los usuarios gestionar sus **asignaturas, temas y archivos**. Los usuarios podrán organizar sus asignaturas, crear temas dentro de cada asignatura y asociar archivos a dichos temas, los cuales podrán visualizar en el navegador. Además, cada usuario podrá marcar sus asignaturas como favoritas y gestionar sus datos personales.

---

## 2. Objetivos

### Objetivos Generales:

1. Desarrollar una plataforma web interactiva donde los usuarios puedan gestionar su contenido educativo de manera organizada.
2. Implementar un sistema de gestión de archivos para que los usuarios puedan visualizar y organizar sus recursos académicos.
3. Permitir la autenticación de usuarios, con funciones de registro y acceso seguro.
4. Gestionar la relación entre asignaturas, temas y archivos, almacenando los metadatos en una base de datos.

### Objetivos Específicos:

1. **Gestión de Asignaturas:**
    - Los usuarios podrán crear, editar y eliminar asignaturas.
    - Las asignaturas podrán marcarse como favoritas.
  2. **Gestión de Temas:**
    - Los usuarios podrán crear y gestionar temas dentro de cada asignatura.
  3. **Gestión de Archivos:**
    - Los usuarios podrán cargar archivos, asociarlos a un tema y ver sus metadatos (fecha de creación, tipo, etc.).
    - Los archivos serán almacenados en la carpeta local o en un servicio de almacenamiento en la nube.
    - Cuando un usuario haga clic en un archivo, este se abrirá en una nueva pestaña del navegador.
  4. **Autenticación y Seguridad:**
    - Implementar un sistema de **registro e inicio de sesión** de usuarios.
    - Asegurar que cada usuario solo pueda acceder a sus propias asignaturas, temas y archivos.
-

### 3. Requisitos Funcionales

#### Usuarios:

1. **Registro de Usuario:** El usuario podrá registrarse proporcionando su correo electrónico, nombre y contraseña.
2. **Inicio de Sesión:** Los usuarios podrán iniciar sesión con su correo electrónico y contraseña.
3. **Gestión de Perfil:** El usuario podrá editar sus datos personales.

#### Asignaturas:

1. Los usuarios podrán crear **asignaturas** proporcionando un nombre.
2. Podrán editar el nombre de las asignaturas existentes.
3. Los usuarios podrán eliminar asignaturas.
4. Las asignaturas podrán marcarse como **favoritas**, lo que permitirá a los usuarios tener un acceso rápido a sus asignaturas preferidas.

#### Temas:

1. Los usuarios podrán crear **temas** dentro de una asignatura.
2. Cada tema tendrá un nombre y una descripción opcional.
3. Los usuarios podrán eliminar temas.

#### Archivos:

1. Los usuarios podrán subir **archivos** a los temas.
2. Los archivos estarán asociados a un tema específico.
3. Los metadatos de los archivos (nombre, fecha de creación, tipo, tamaño) se almacenarán en la base de datos.
4. Los archivos se almacenarán en una carpeta del servidor o en un sistema de almacenamiento en la nube.
5. Cuando un usuario haga clic en un archivo, este se abrirá en una nueva pestaña del navegador.

#### Autenticación y Seguridad:

1. Los usuarios deberán autenticar su identidad antes de acceder a sus asignaturas, temas y archivos.
2. El sistema será **seguro**, almacenando las contraseñas de forma cifrada.

---

### 4. Arquitectura de la Aplicación

La arquitectura de la aplicación se basará en una **aplicación de una sola página (SPA)**, que comunicará el frontend con el backend mediante **API REST**.

#### Frontend:

- **Framework:** React.js (para la construcción de la SPA).
- **Estado:** React Query (para la gestión de datos asíncronos y caché).
- **Estilos:** Bootstrap y/o CSS personalizado para la maquetación y diseño.

#### Backend:

- **Lenguaje:** Java con Spring Boot.
- **Base de Datos:** MySQL (para el almacenamiento de usuarios, asignaturas, temas y archivos).
- **Almacenamiento de Archivos:** Se utilizará una carpeta local para almacenar los archivos inicialmente. Posteriormente, se podría integrar con un servicio de almacenamiento en la nube como Amazon S3.
- **Autenticación:** JWT (JSON Web Tokens) para la autenticación de usuarios.

#### Diagrama de la Estructura de la Base de Datos:

- **Usuarios:** Tabla users con campos id, email, password, name.
- **Asignaturas:** Tabla subjects con campos id, name, isFav, user\_id.
- **Temas:** Tabla topics con campos id, name, description, subject\_id.
- **Archivos:** Tabla files con campos id, name, type, size, filePath, creationDate, topic\_id.

\*Aproximadamente

#### Relaciones:

1. Un **Usuario** puede tener muchas **Asignaturas**.
2. Una **Asignatura** tiene muchos **Temas**.
3. Un **Tema** tiene muchos **Archivos**.

## 5. Implementación

### 1. Backend (Spring Boot):

1. **Modelos de Datos:** Se implementarán los modelos de User, Subject, Topic, y File con sus respectivas relaciones.
2. **Controladores:** Se implementarán los endpoints para gestionar usuarios, asignaturas, temas y archivos.
3. **Autenticación:** Se implementará un sistema de autenticación usando JWT para proteger las rutas.
4. **Almacenamiento de Archivos:** Los archivos se almacenarán inicialmente en el servidor local

### 2. Frontend (React.js):

1. **Pantalla de Registro e Inicio de Sesión:** Se implementarán formularios para que los usuarios puedan registrarse e iniciar sesión.

2. **Vista de Asignaturas:** Los usuarios podrán ver todas sus asignaturas, crear nuevas, editarlas y marcarlas como favoritas.
3. **Vista de Temas:** Al hacer clic en una asignatura, se mostrarán los temas de esa asignatura. Los usuarios podrán crear, editar y eliminar temas.
4. **Vista de Archivos:** Los usuarios podrán subir archivos a los temas, visualizar sus metadatos y acceder a los archivos en el navegador.