

Documentación de la Aplicación

1. posts.py (Manejo de publicaciones)

Este archivo contiene las rutas para gestionar las publicaciones del blog.

- **list_posts(tag: Optional[str])** → Obtiene todas las publicaciones o filtra por una etiqueta.
- **create_post(post: schemas.PostCreate)** → Crea un nuevo post si el usuario está autenticado.
- **update_post(post_id: int, post_data: schemas.PostCreate)** → Permite al autor actualizar su post.
- **delete_post(post_id: int)** → Permite al autor eliminar su post.
- **rate_post(post_id: int, rating: schemas.RatingCreate)** → Permite a los usuarios calificar posts de otros.
- **get_post(post_id: int)** → Obtiene un post específico con sus detalles.

2. users.py (Manejo de usuarios)

- **register_user(user: schemas.UserCreate)** → Registra un nuevo usuario.
- **login_user(form_data: OAuth2PasswordRequestForm)** → Maneja el inicio de sesión y genera un token JWT.
- **get_current_user(token: str)** → Obtiene los datos del usuario autenticado.

3. models.py (Definición de base de datos)

Contiene los modelos de la base de datos con SQLAlchemy.

- **User** → Representa a los usuarios registrados.
- **Post** → Modelo de una publicación con autor, contenido y etiquetas.
- **Tag** → Modelo de etiquetas asociadas a publicaciones.
- **Rating** → Modelo de calificaciones de publicaciones.

Frontend (React + Vite)

1. Dashboard.jsx (Vista principal de posts)

- **Obtiene y muestra las publicaciones** con un preview del contenido.
- **Filtra posts por etiquetas** mediante un dropdown.
- **Muestra el nombre del usuario autenticado y un botón para cerrar sesión.**
- **Incluye botón para crear un nuevo post.**

2. PostDetail.jsx (Detalle de una publicación)

- **Muestra el contenido completo del post, el autor, las etiquetas y la fecha de publicación.**
- **Incluye opciones de calificar el post** (si el usuario no es el autor).
- **Incluye botones para editar o eliminar el post** (si el usuario es el autor).

3. Login.jsx (Inicio de sesión)

- **Permite a los usuarios autenticarse** y almacena el token en localStorage.
- **Redirige al Dashboard tras un inicio de sesión exitoso.**

4. EditPost.jsx (Edición de publicaciones)

- **Carga los datos de una publicación existente** y permite modificarlos.
- **Guarda los cambios y redirige al Dashboard tras actualizar el post.**