

# Guía para Desarrolladores - API REST de Usuarios y Notas

---

## Introducción

Bienvenido al proyecto **API REST de Usuarios y Notas**. Este sistema está desarrollado en **Java** utilizando **Spring Boot** y permite gestionar usuarios y sus notas de manera persistente. La arquitectura sigue buenas prácticas, como la separación de capas y el uso de **JPA** para el acceso a datos.

---

## Estructura del Proyecto

El proyecto está organizado en las siguientes carpetas y archivos principales:

```
com.miapp.notasapi/  
  
├─ controller/ # Controladores REST  
  
|   ├─ UsuarioController.java # Endpoints de usuario  
  
|   └─ NotaController.java # Endpoints de nota  
  
|  
  
├─ service/ # Lógica de negocio  
  
|   ├─ UsuarioService.java  
  
|   ├─ UsuarioServiceImpl.java  
  
|   └─ NotaService.java  
  
|   └─ AbstractCrudService.java # Implementación genérica de CRUD  
  
|
```

```
├── repository/ # Acceso a datos
|   ├── UsuarioRepository.java
|   └── NotaRepository.java
|
├── model/ # Entidades
|   ├── Usuario.java # Entidad usuario
|   └── Nota.java # Entidad nota
|
├── resources/ # Configuración
|   ├── application.properties # Configuración de la aplicación
|   └── static/ # Archivos estáticos (si aplica)
|
├── test/ # Pruebas del sistema
|   └── NotasapiApplicationTests.java
|
├── README.md # Documentación básica del proyecto
├── LICENSE # Licencia del proyecto
└── Documentación_Desarrolladores_API_Notas.md # Guía para desarrolladores
```



## Cómo Ejecutar el Proyecto

### Requisitos Previos

1. **Java:** Asegúrate de tener instalado **Java 17** o superior.
2. **Maven:** El proyecto utiliza Maven como herramienta de construcción.

3. **MySQL:** Configura una base de datos MySQL local con el nombre `notasapi`.

## Configuración de la Base de Datos

1. Crea una base de datos en MySQL:

```
``sql  
  
CREATE DATABASE notasapi;  
  
...
```

2. Configura las credenciales en el archivo `application.properties`:

```
spring.datasource.username=<tu_usuario>  
  
spring.datasource.password=<tu_contraseña>
```

## Pasos para Ejecutar

1. Clona el repositorio:

```
git clone https://github.com/tu_usuario/API-Notas.git  
  
cd API-Notas
```

2. Compila el proyecto con Maven:

```
./mvnw clean install
```

3. Ejecuta la aplicación:

```
./mvnw spring-boot:run
```

4. Accede a la API en <http://localhost:8080>.

---

## Descripción de las Capas

### Controller

Los controladores definen los endpoints REST para interactuar con la API. Ejemplo:

- `UsuarioController.java` : Gestiona usuarios.
- `NotaController.java` : Gestiona notas.

### Service

La capa de servicio contiene la lógica de negocio. Ejemplo:

- `UsuarioServiceImpl.java` : Implementación de los métodos para gestionar usuarios.
- `AbstractCrudService.java` : Implementación genérica para operaciones CRUD.

### Repository

Los repositorios interactúan directamente con la base de datos utilizando **Spring Data JPA**. Ejemplo:

- `UsuarioRepository.java` : Operaciones sobre la entidad `Usuario` .
- `NotaRepository.java` : Operaciones sobre la entidad `Nota` .

### Model

Las entidades representan las tablas de la base de datos. Ejemplo:

- `Usuario.java` : Define la estructura de la tabla `usuarios` .
- `Nota.java` : Define la estructura de la tabla `notas` .

---

## Pruebas

Las pruebas están ubicadas en la carpeta `test/` . Utilizan **JUnit** para validar la funcionalidad de la aplicación.

---



# Licencia

Este proyecto está bajo la licencia **MIT**. Consulta el archivo `LICENSE` para más detalles.

---

¡Gracias por contribuir al proyecto!