Documentación del Proyecto — Examen1Back2

# Thomas Serna Triviño

# 1. Descripción del proyecto

El sistema Examen1Back2 es una aplicación desarrollada en Java 17 utilizando el framework Spring Boot 3, el gestor de dependencias Gradle y una base de datos relacional MySQL. Su propósito es gestionar usuarios, docentes y cursos, permitiendo administrar información clave de cada entidad y su relación dentro de un entorno académico.

# 2. Arquitectura del sistema

El proyecto está diseñado bajo la arquitectura de capas MVC (Modelo–Vista–Controlador).

- Modelos (Entidad/JPA): Representan las tablas de la base de datos (Usuario, Docente, Curso, TipoUsuario).  
- Repositorios: Interfaces que extienden JpaRepository y permiten acceder a la base de datos sin necesidad de SQL.  
- Servicios: Contienen la lógica de negocio.  
- Controladores REST: Exponen endpoints accesibles vía HTTP (API REST).  
  
Flujo de información: Cliente → Controlador → Servicio → Repositorio → Base de datos.

# 3. Modelado de datos

Entidades principales:

Usuario: id, nombre, correoElectronico, contrasena, teléfono, tipoUsuario (enum).  
Docente: id, nombre, especialidad, cursos.  
Curso: id, nombre, descripción, docente.  
  
Relaciones:  
- Un Docente puede tener varios Cursos (OneToMany).  
- Un Curso está asociado a un único Docente.  
- Un Usuario tiene un TipoUsuario específico.

# 4. Endpoints REST disponibles

Usuarios:  
- GET /usuarios → Lista todos los usuarios.  
- GET /usuarios/{id} → Obtiene un usuario por su ID.  
- POST /usuarios → Crea un nuevo usuario.  
- PUT /usuarios/{id} → Actualiza un usuario existente.  
- DELETE /usuarios/{id} → Elimina un usuario.  
  
Docentes:  
- GET /docentes → Lista todos los docentes.  
- POST /docentes → Registra un docente.  
  
Cursos:  
- GET /cursos → Lista todos los cursos.  
- POST /cursos → Crea un nuevo curso.  
- GET /cursos/{id} → Obtiene un curso por ID.

# 5. Casos de uso

1. Registrar un usuario:  
 - El administrador envía un POST /usuarios con los datos.  
 - El sistema valida y guarda la información.  
  
2. Asignar un curso a un docente:  
 - El administrador envía un POST /cursos indicando el docente asignado.  
 - El curso se almacena con referencia al docente.  
  
3. Consultar cursos de un docente:  
 - Petición GET /docentes/{id}/cursos.  
 - El sistema devuelve la lista de cursos asociados.

# 6. Instrucciones de despliegue

1. Clonar el repositorio.  
2. Crear la base de datos en MySQL:  
 CREATE DATABASE develop\_db;  
3. Configurar application. properties:  
 spring.datasource.url=jdbc: mysql://localhost:3306/develop\_db  
 spring.datasource.username=tu\_usuario  
 spring.datasource.password=tu\_contraseña  
 spring.jpa.hibernate.ddl-auto=update  
4. Ejecutar: gradlew.bat bootRun  
5. Acceder: http://localhost:8080

# 7. Listado de errores corregidos con explicación

Clase 1:Curso.java

**Errores detectados**

1. @I → debería ser @Id.
2. @Ge(strategy = IDENTITY) → debería ser @GeneratedValue(strategy = GenerationType.IDENTITY).
3. Falta punto y coma (;) en private String nombre.
4. En la anotación @JoinColumn **sobran el ; al final**.
5. Docente docente no tiene modificador de acceso (private).
6. No hay **getters y setters** para los atributos.

Clase 2: Docente.java

**Errores detectados**

1. @Entit → debería ser @Entity.
2. Falta la anotación @Id en el campo id.
3. Faltan constructores (solo hay getters/setters parciales).
4. Le faltan getters/setters para cursos y usuario.

Clase 3: Usuario.java

**Errores detectados**

1. @GeneratedValue(strategy = GenerationType.) → incompleto, debe ser GenerationType.IDENTITY.
2. Faltan getters/setters.
3. Falta constructor vacío y opcional con parámetros.
4. No está importado TipoUsuario (debería ser un enum en el proyecto).

Clase 4: TipoUsuario.java

Creación de Enum que liste los tipos de usuarios

# 8. Guía paso a paso de cómo realizar la conexión a la base de datos

1. Instalar MySQL Server y asegurarse de que esté en ejecución.  
2. Crear la base de datos con el comando:  
 CREATE DATABASE develop\_db;  
3. Editar el archivo src/main/resources/application.properties con las credenciales:  
 spring.datasource.url=jdbc:mysql://localhost:3306/develop\_db  
 spring.datasource.username=tu\_usuario  
 spring.datasource.password=tu\_contraseña  
 spring.jpa.hibernate.ddl-auto=update  
4. Verificar que el puerto 3306 de MySQL esté habilitado.  
5. Ejecutar el proyecto con: gradlew bootRun.  
6. Confirmar la conexión observando en consola la creación de tablas.

# 9. Recomendaciones para evitar errores similares en el futuro

- Utilizar siempre nombres de variables sin caracteres especiales para evitar errores de compilación.  
- Mantener las clases enum correctamente definidas cuando se usen con @Enumerated.  
- Antes de ejecutar 'clean', asegurarse de cerrar procesos Java que puedan estar usando archivos en build.  
- Documentar cada cambio en las entidades y sus relaciones para evitar confusión futura.  
- Probar los cambios con 'gradlew build' antes de ejecutar 'bootRun' para identificar errores de compilación tempranos.

# 10. Conclusiones

El sistema Examen1Back2 representa un backend robusto y escalable para la gestión de usuarios, docentes y cursos. Su arquitectura en capas, uso de Spring Boot y soporte de JPA permiten una fácil extensión y mantenimiento. Gracias al uso de IA generativa, fue posible elaborar la documentación técnica y funcional de manera ágil, lo que facilita la comprensión del proyecto, su despliegue y futuras mejoras.