**Resumen del Proyecto**

**1. Descripción del Proyecto**

El proyecto **Examen1Back22025** es una aplicación backend desarrollada con **Spring Boot** para la gestión de entidades como Curso, Docente y Usuario. El sistema incluye capas para manejar la lógica de negocio y el acceso a datos mediante **Spring Data JPA**, con soporte para bases de datos como MySQL y H2.  
Su estructura original estaba configurada para usar **Gradle** como gestor de dependencias y **Java 17** como versión mínima del JDK.

**2. Listado de Errores Corregidos con Explicación**

* **Error de versión de Java**
  + *Problema:* El proyecto requería **Java 17**, pero el IDE estaba usando **Java 11**.
  + *Solución:* Se cambió la configuración en IntelliJ para usar el JDK 17 en **Project SDK** y en la configuración de Gradle/Maven.
* **Falta de getters y setters en la clase Curso**
  + *Problema:* No se podían acceder ni modificar atributos desde otras clases.
  + *Solución:* Se implementaron los métodos get y set para todos los atributos.
* **Clase Docente sin constructores y con getters/setters incompletos**
  + *Problema:* No se podía inicializar correctamente ni manipular todos los atributos.
  + *Solución:* Se agregaron constructores (por defecto y con parámetros) y se completaron los getters/setters.
* **Clase Usuario incompleta**
  + *Problema:* El constructor no incluía el atributo docente y faltaban sus getters/setters.
  + *Solución:* Se agregó el atributo docente al constructor y sus métodos de acceso.
* **Gestor de dependencias incompatible (Gradle → Maven)**
  + *Problema:* El proyecto estaba configurado con Gradle, pero el entorno de trabajo usaba Maven.
  + *Solución:* Se eliminaron los archivos y carpetas de Gradle (build.gradle, settings.gradle, .gradle, gradlew, gradlew.bat) y se creó un archivo pom.xml equivalente con todas las dependencias originales.

**3. Guía Paso a Paso para Conectar a la Base de Datos**

**Conexión a MySQL en Spring Boot:**

1. **Instalar MySQL y crear la base de datos**
2. CREATE DATABASE examen1db;
3. **Agregar dependencia en pom.xml**
4. <dependency>
5. <groupId>com.mysql</groupId>
6. <artifactId>mysql-connector-j</artifactId>
7. <scope>runtime</scope>
8. </dependency>
9. **Configurar application.properties**
10. spring.datasource.url=jdbc:mysql://localhost:3306/examen1db
11. spring.datasource.username=root
12. spring.datasource.password=tu\_password
13. spring.jpa.hibernate.ddl-auto=update
14. spring.jpa.show-sql=true
15. spring.jpa.properties.hibernate.dialect=org.hibernate.dialect.MySQL8Dialect
16. **Crear entidades con anotaciones JPA**
17. @Entity
18. public class Curso {
19. @Id
20. @GeneratedValue(strategy = GenerationType.IDENTITY)
21. private Long id;
22. private String nombre;
23. // getters y setters
24. }
25. **Ejecutar el proyecto**
26. mvn spring-boot:run

**4. Recomendaciones para Evitar Errores Similares**

* **Verificar la versión de Java antes de comenzar**  
  Configurar el IDE y el proyecto con la misma versión de JDK que requiere Spring Boot.
* **Unificar el gestor de dependencias**  
  Si la organización usa Maven, crear el proyecto desde el inicio con Maven para evitar migraciones.
* **Mantener buenas prácticas en las clases de modelo**  
  Siempre incluir constructores, getters y setters completos.
* **Probar conexión a la base de datos en una clase independiente**  
  Antes de integrar en el proyecto, validar que la conexión funcione con credenciales y configuración correctas.
* **Evitar cambios drásticos en medio de un examen o entrega**  
  La migración de Gradle a Maven es viable, pero debe planificarse para evitar pérdida de tiempo.