

Resumen del Proyecto – Backend 2 (Examen 1)

Descripción del Proyecto

Este proyecto consiste en la construcción de un modelo básico en Java usando **JPA/Hibernate** para representar entidades como Curso, Docente y Usuario, las cuales están relacionadas entre sí. El objetivo es mapear correctamente las clases del modelo a tablas en una base de datos relacional utilizando anotaciones de JPA, y establecer relaciones @OneToOne, @ManyToOne y @OneToMany.

Listado de Errores Corregidos con Explicación

◆ Clase Curso

- **Línea 5:** Se completó la palabra @Entity para indicar que es una entidad JPA.
- **Línea 6:** Se agregó la anotación @Table(name = "curso") para nombrar explícitamente la tabla.
- **Línea 9:** Se agregó @Id para declarar la clave primaria.
- **Línea 10:** Se completó la anotación @GeneratedValue(strategy = GenerationType.IDENTITY) para autogenerar el ID.
- **Línea 12:** Se corrigió agregando un punto y coma al final de la línea para evitar error de sintaxis.
- **Línea 15:** Se eliminó un punto y coma incorrecto dentro de la anotación @JoinColumn.
- **Final:** Se generaron los getters y setters para los atributos id, nombre y docente.

◆ Clase Docente

- **Línea 7:** Se completó la palabra @Entity para definir la clase como entidad.
- **Línea 8:** Se añadió @Table(name = "docente") para nombrar la tabla.
- **Línea 27:** Se agregó el constructor con todos los campos.
- **Línea 34:** Se agregó el constructor vacío.
- **Final:** Se implementaron los getters y setters para los atributos cursos y usuario, necesarios para la relación bidireccional.

◆ Clase Usuario

- **Línea 5:** Se completó la palabra @Entity.
- **Línea 10:** Se completó @GeneratedValue con strategy = GenerationType.IDENTITY.
- **Línea 14:** Se corrigió la anotación @Column(nullable = false, length = 100).

- **Línea 17:** Se corrigió @Column(name = "correo_electronico", unique = true).
 - **Línea 19:** Se cambió contraseña por contrasena para evitar conflictos con codificaciones y compatibilidad.
 - **Enum TipoUsuario:** Se creó un enum para representar los tipos de usuario (ADMIN, DOCENTE, ESTUDIANTE) y se usó con @Enumerated(EnumType.STRING).
 - **Final:** Se generaron los getters y setters correspondientes.
-

Guía paso a paso: Conexión a la Base de Datos con JPA (Hibernate)

1. Agregar dependencias en pom.xml (si usas Maven):

```
<dependencies>
<dependency>
    <groupId>jakarta.persistence</groupId>
    <artifactId>jakarta.persistence-api</artifactId>
    <version>3.1.0</version>
</dependency>
<dependency>
    <groupId>org.hibernate.orm</groupId>
    <artifactId>hibernate-core</artifactId>
    <version>6.2.5.Final</version>
</dependency>
<dependency>
    <groupId>com.mysql</groupId>
    <artifactId>mysql-connector-j</artifactId>
    <version>8.0.33</version>
</dependency>
</dependencies>
```

2. Configurar el archivo persistence.xml (en src/main/resources/META-INF/):

```
<persistence xmlns="https://jakarta.ee/xml/ns/persistence"
    version="3.0">
```

```

<persistence-unit name="examenPU">

    <class>com.example.Examen1Back2.modelos.Usuario</class>

    <class>com.example.Examen1Back2.modelos.Docente</class>

    <class>com.example.Examen1Back2.modelos.Curso</class>

    <properties>

        <property name="jakarta.persistence.jdbc.url"
value="jdbc:mysql://localhost:3306/examen_db"/>

        <property name="jakarta.persistence.jdbc.user" value="root"/>

        <property name="jakarta.persistence.jdbc.password" value="tu_contraseña"/>

        <property name="jakarta.persistence.jdbc.driver" value="com.mysql.cj.jdbc.Driver"/>


        <property name="hibernate.dialect" value="org.hibernate.dialect.MySQL8Dialect"/>

        <property name="hibernate.hbm2ddl.auto" value="update"/>

        <property name="hibernate.show_sql" value="true"/>

    </properties>

</persistence-unit>

</persistence>

```

3. **Verifica que el puerto, base de datos y credenciales sean correctas.**

4. **Crea la base de datos examen_db en MySQL:**

```
CREATE DATABASE examen_db;
```

5. **Ejecuta el proyecto.** Hibernate generará automáticamente las tablas.

✅ Recomendaciones para evitar errores similares en el futuro

1. **Verifica que todas las anotaciones JPA estén completas** (@Entity, @Id, @GeneratedValue, etc.).
2. **Usa nombres de variables sin caracteres especiales** (contraseña → contrasena) para evitar problemas de codificación.
3. **Asegúrate de que los enums estén bien definidos y en archivos separados.**
4. **Revisa los imports:** importa desde jakarta.persistence o javax.persistence de forma consistente.

5. **Usa el IDE para generar los getters, setters y constructores automáticamente** para evitar omisiones.
6. **Siempre prueba la conexión a la base de datos antes de ejecutar la aplicación.**
7. **Usa EnumType.STRING en lugar de ORDINAL** para que los datos del enum sean legibles y más seguros en cambios futuros.