INSTITUTO DE EDUCACIÓN SUPERIOR CIBERTEC
DIRECCIÓN ACADÉMICA
CARRERA PROFESIONALES

CURSO	**:**	**Desarrollo de Aplicaciones Web I (0265)**
PROFESOR	**:**	**César Enrique Santos Torres**
CICLO	**:**	**Quinto**
SECCIÓN	**:**	**26750**
GRUPO	**:**	**2024336362**
FECHA	**:**	**19/11/2024**
DURACIÓN	**:**	**3 horas**

NOTA

ALUMNO (A) : BLAS GALICIA RAMIRO

CASO DE LABORATORIO 1 (CL1)

Consideraciones generales:

- El laboratorio consta de 4 partes, cada parte tiene una secuencia de pasos las cuales deberá ir acompañada (De forma obligatoria) de capturas de pantalla de lo implementado.
- Sólo debe subir este documento, con sus evidencias y respuestas en él. El código fuente de ambos proyectos debe ser subido a Github (Adjuntar links del repositorio). No se aceptará código zipeado.
- El nombre del presente archivo deberá tener la siguiente estructura: "DAWI-APELLIDOS-NOMBRES.pdf".

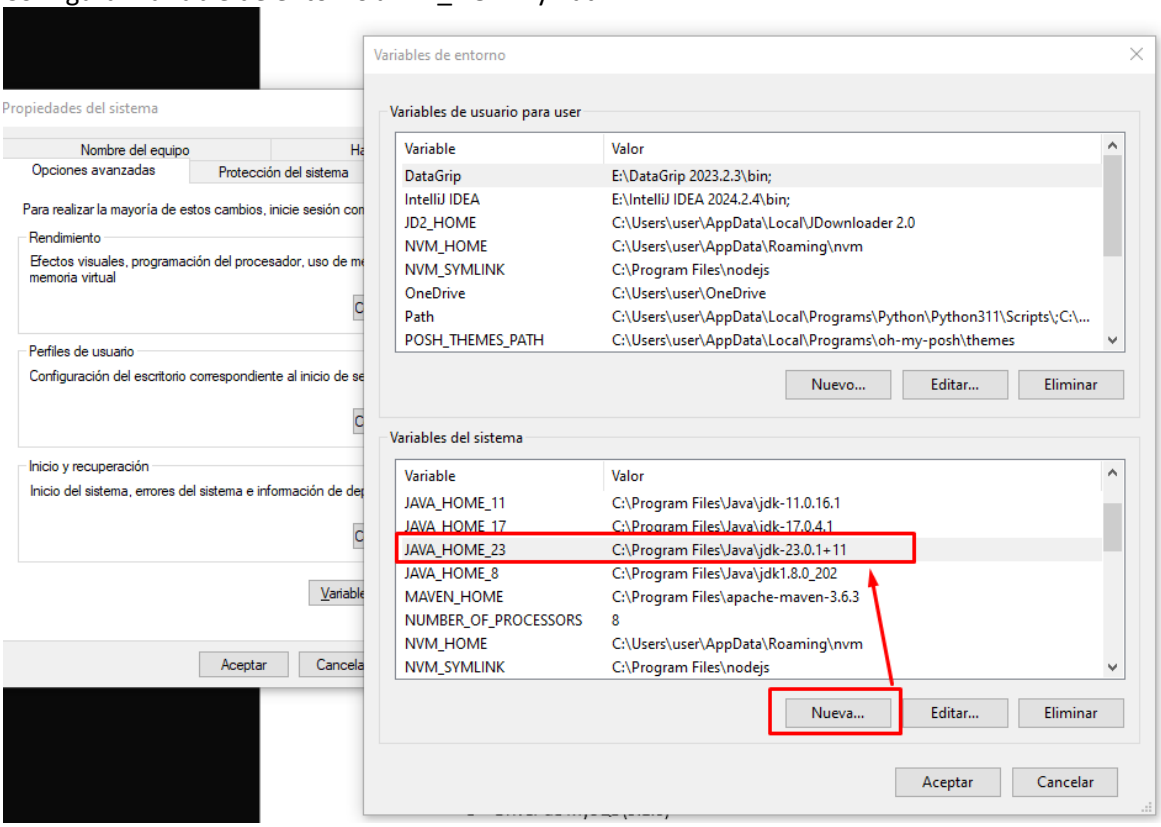
LOGRO DE LA EVALUACIÓN:
Al término de la evaluación, el alumno implementa las operaciones de mantenimiento sobre una entidad utilizando Java Persistence API.

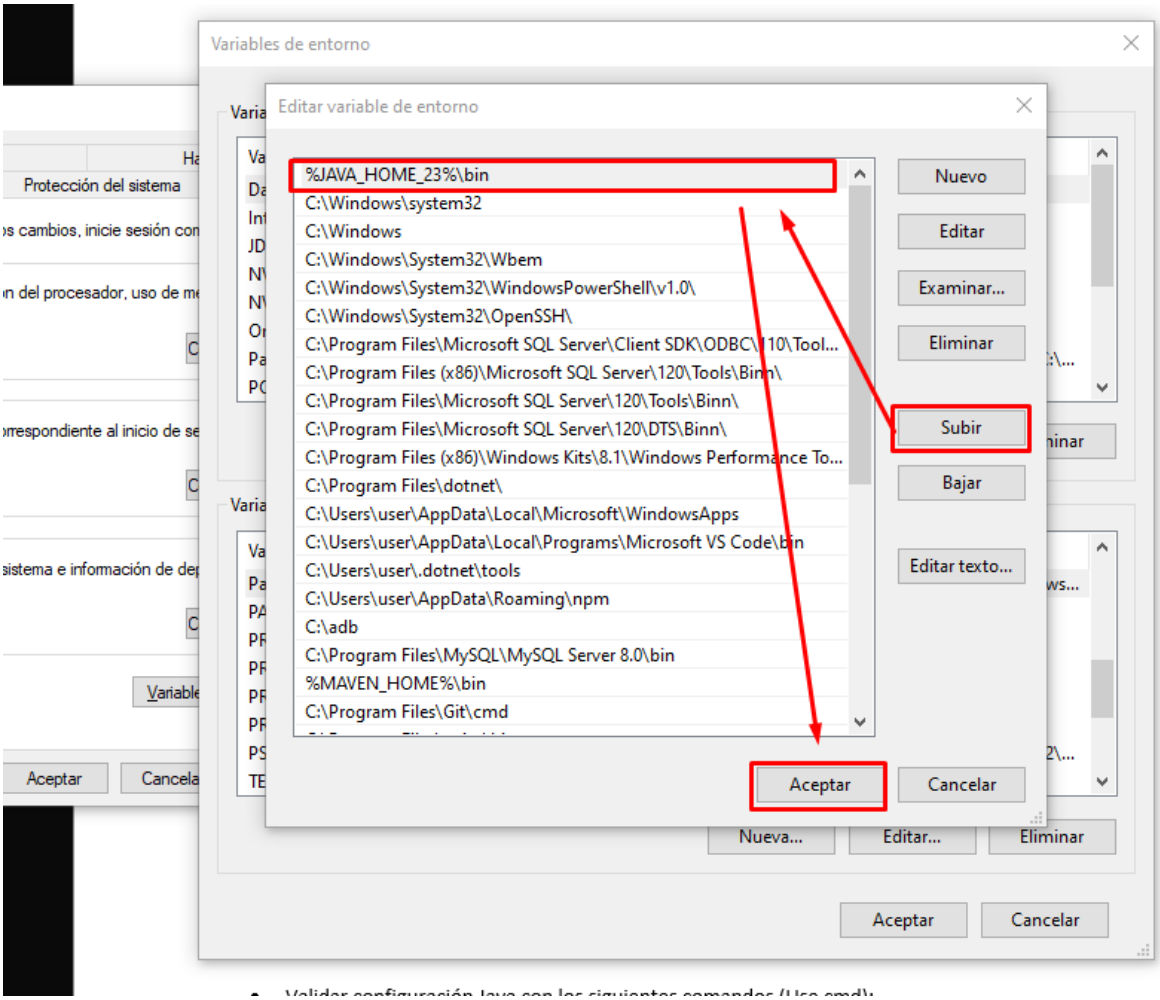
CONSOLIDADO

Pregunta	Puntaje		Llenar solo en caso de Recalificación justificada	
	Máximo	Obtenido	Sustento	Puntaje
1	5			
2	5			
3	5			
4	5			
Total	20			
			Nota Recalificada	

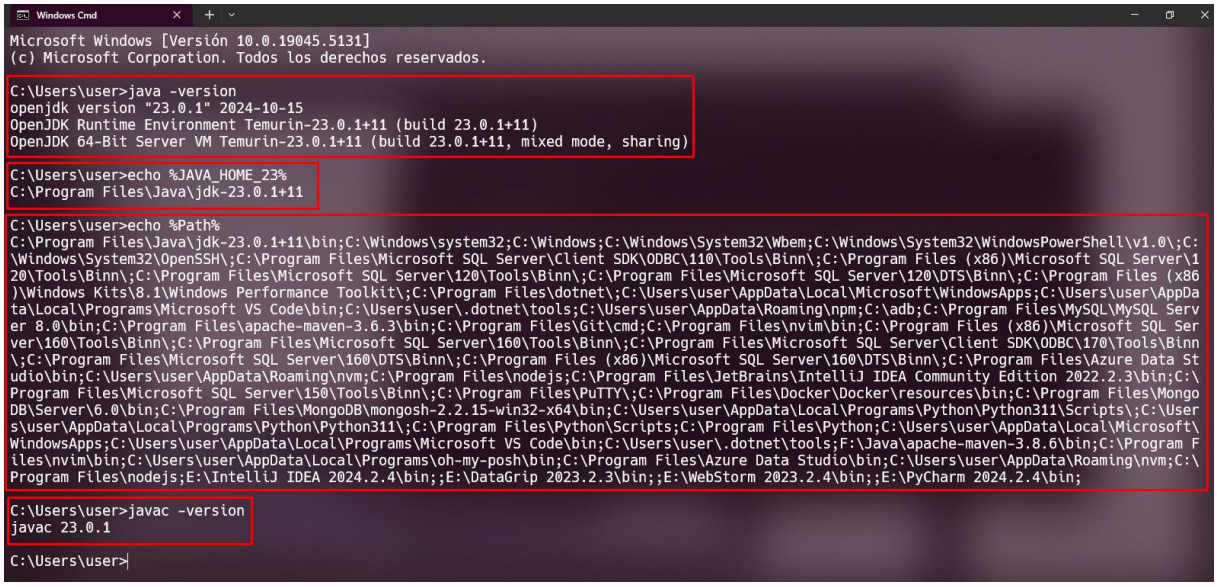
Parte 01 Configuración básica (25%)

- Descargar JDK versión 23 de https://adoptium.net/es/temurin/releases/
- Configurar variable de entorno JAVA_HOME y Path

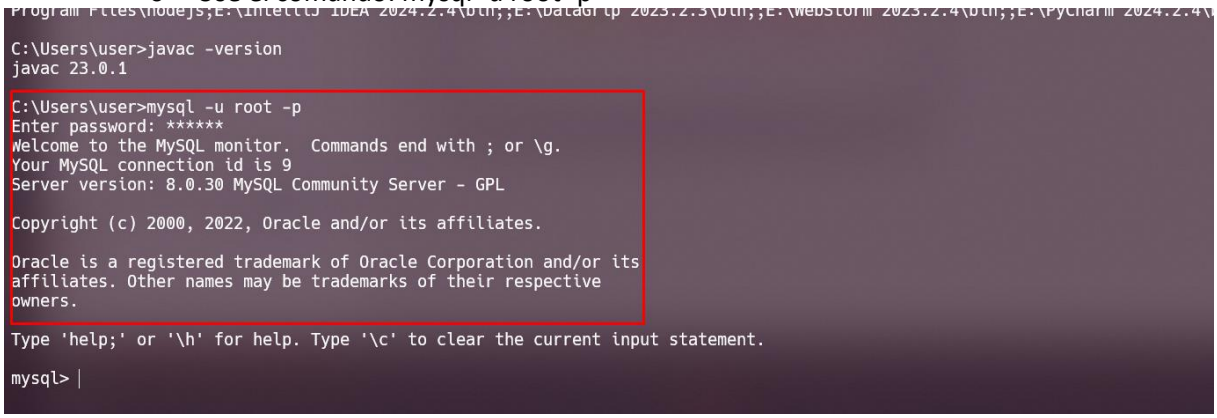




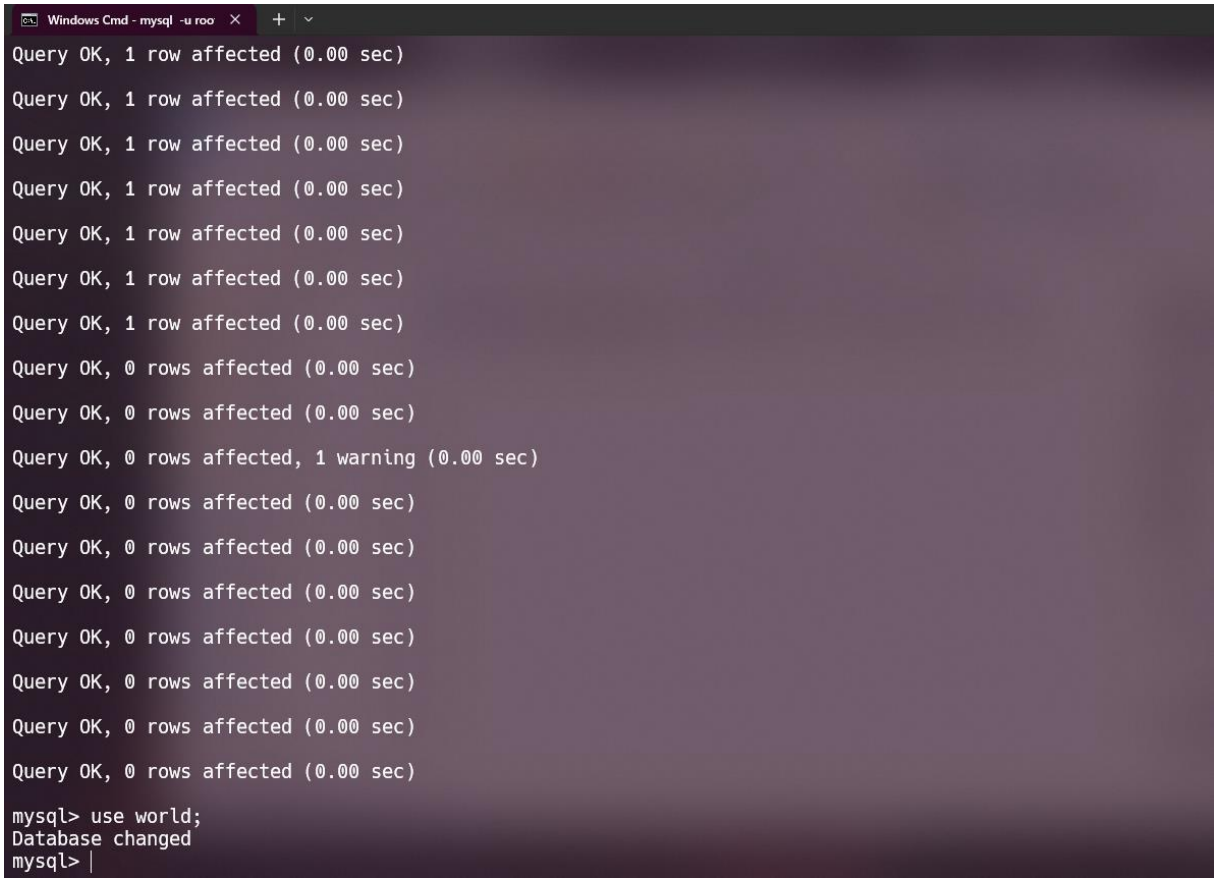
- Validar configuración Java con los siguientes comandos (Use cmd):
  - java -version
  - echo %JAVA_HOME%
  - echo %Path%

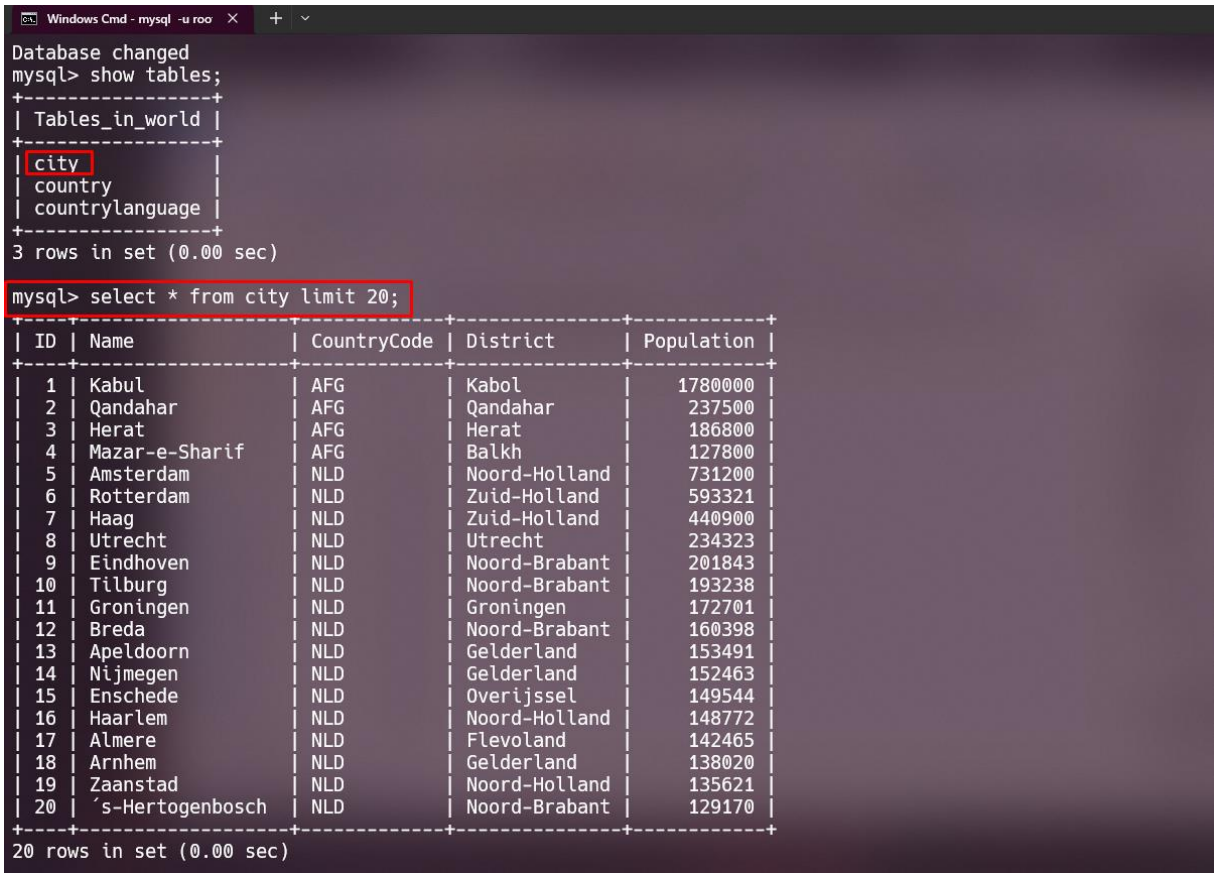


- Conectar al servidor MySQL usando la terminal (Use cmd):
    - Use el comando: mysql -u root -p



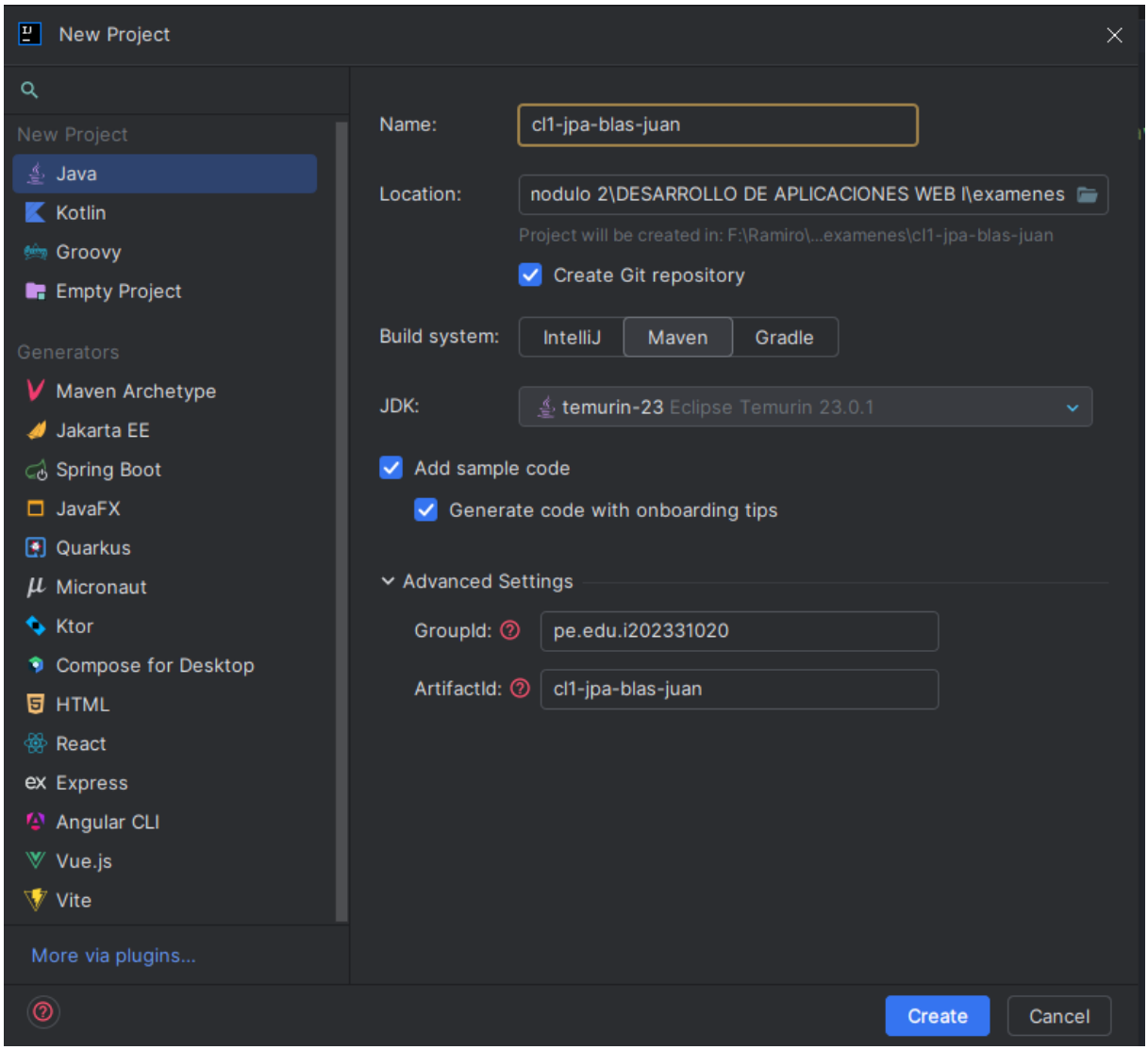
- Restarurar bd “world” de https://downloads.mysql.com/docs/world-db.zip (Use cmd):
    - Use el comando: source <ruta-archivo-world.sql>;
- Usar bd “world” y hacer un select de los primeros 20 registros de la tabla “city” (Use cmd).

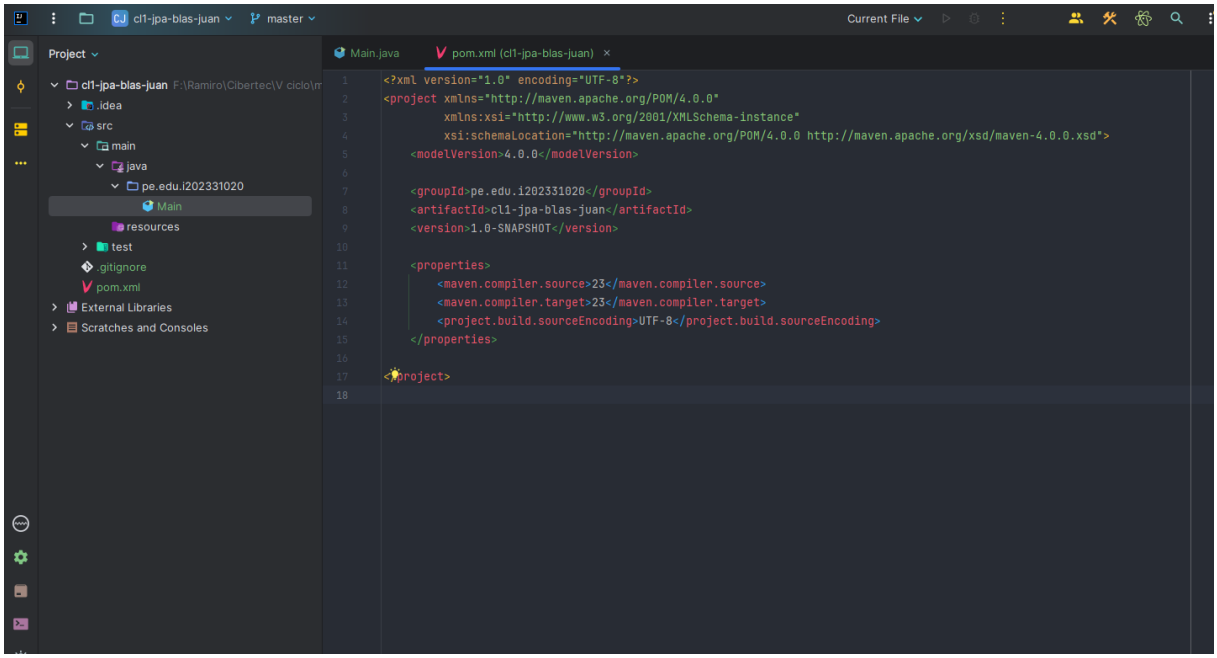




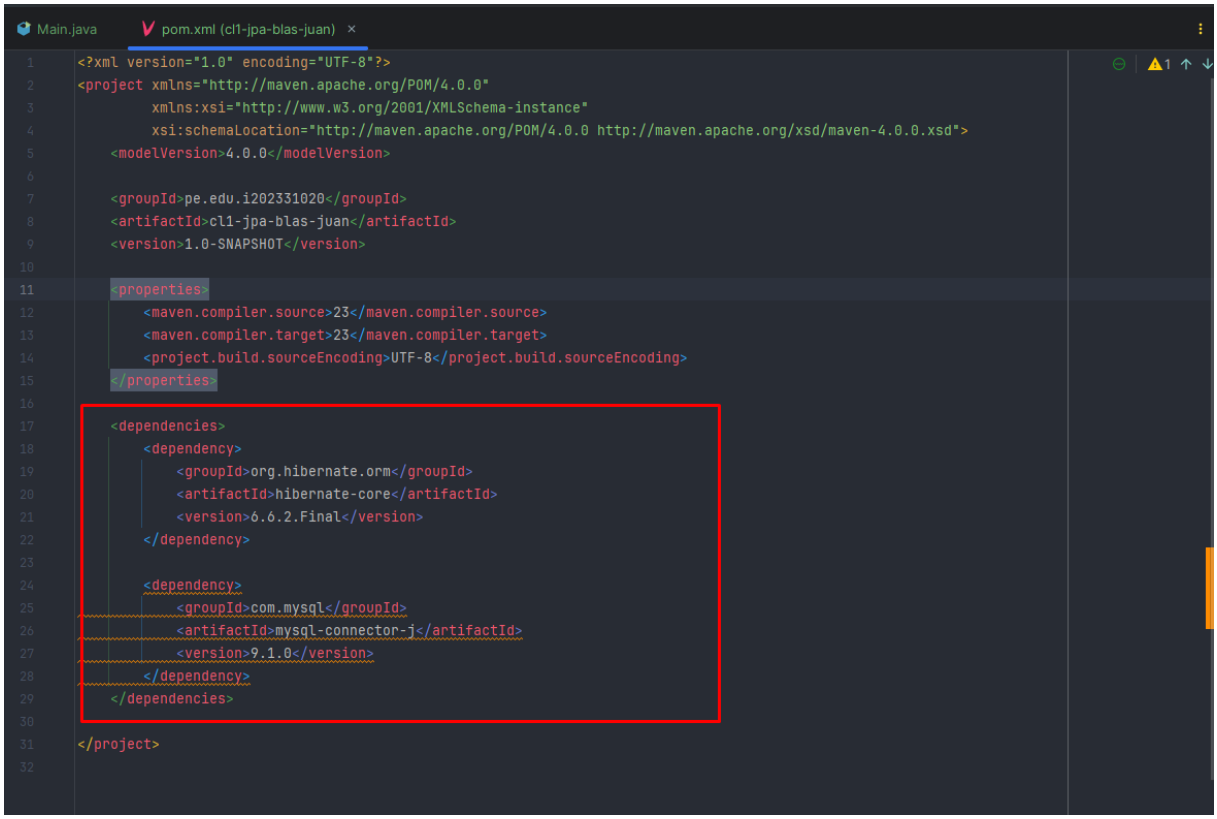
Parte 02 Proyecto JPA-Hibernate (25%)

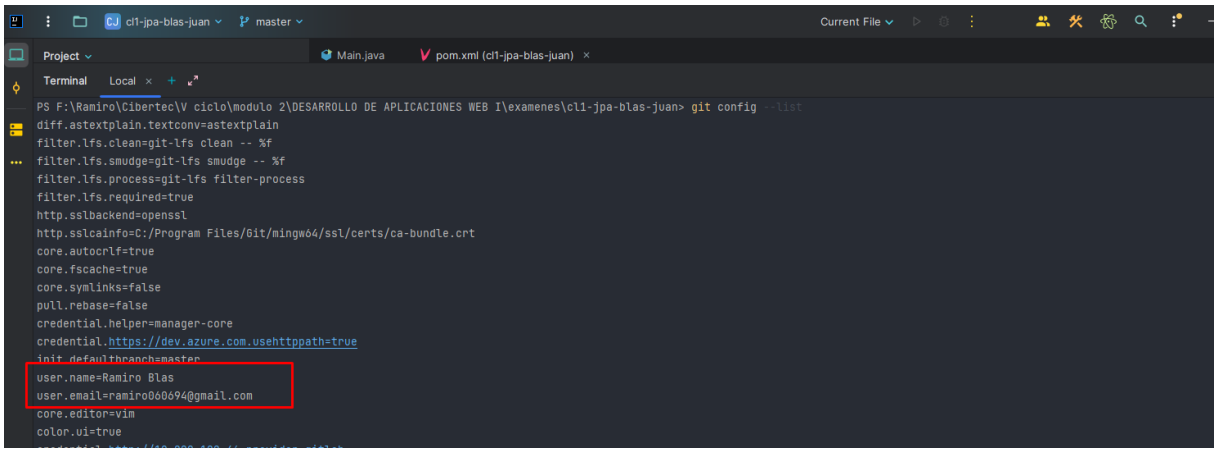
- Crear un proyecto JPA-Hibernate desde IntelliJ Idea con las siguientes características:
  - Name: cl1-jpa-<apellidoPaterno-primerNombre> (Use minúsculas y guión “-”)
  - Location: Seleccione un directorio con permisos de lectura y escritura
  - Create Git repository: Check
  - Build system: Maven
  - JDK: Eclipse Temurin-23
  - Advanced Settings:
    - GroupId: pe.edu.<codigoEstudiante>
    - Artifact: cl1-jpa-<apellidoPaterno-primerNombre>

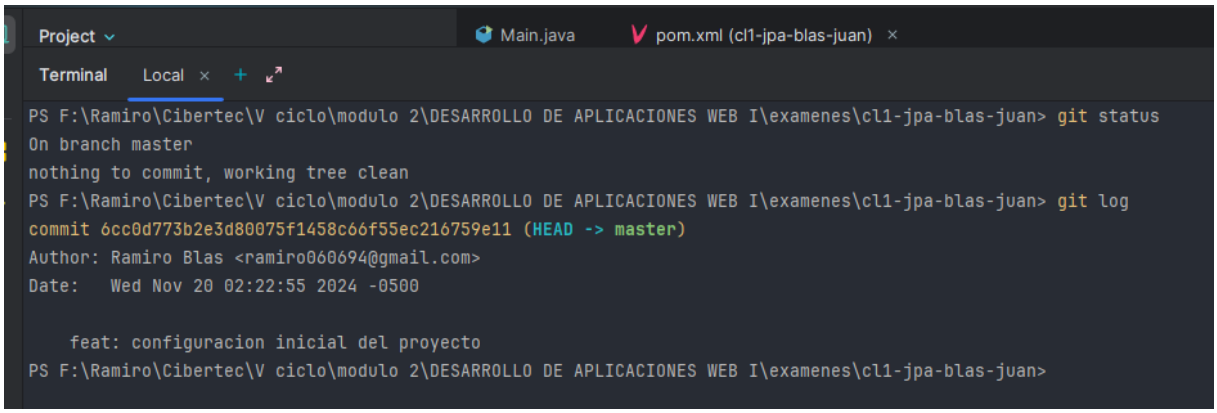

New Project
New Project
Java
Kotlin
Groovy
Empty Project
Generators
Maven Archetype
Jakarta EE
Spring Boot
JavaFX
Quarkus
Micronaut
Ktor
Compose for Desktop
HTML
React
Express
Angular CLI
Vue.js
Vite
More via plugins...
Name:
cl1-jpa-blas-juan
Location:
nodulo 2\DESARROLLO DE APLICACIONES WEB I\examenes
Project will be created in: F:\Ramiro\...examenes\cl1-jpa-blas-juan
Create Git repository
Build system:
IntelliJ
Maven
Gradle
JDK:
temurin-23 Eclipse Temurin 23.0.1
Add sample code
Generate code with onboarding tips
Advanced Settings
GroupId:
pe.edu.i202331020
ArtifactId:
cl1-jpa-blas-juan
Create
Cancel


cl1-jpa-blas-juan
master
Current File
Project
Main.java
pom.xml (cl1-jpa-blas-juan)
cl1-jpa-blas-juan
.idea
src
main
java
pe.edu.i202331020
Main
resources
test
.gitignore
pom.xml
External Libraries
Scratches and Consoles
<?xml version="1.0" encoding="UTF-8"?>
<project xmlns="http://maven.apache.org/POM/4.0.0"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 http://maven.apache.org/xsd/maven-4.0.0.xsd">
<modelVersion>4.0.0</modelVersion>
<groupId>pe.edu.i202331020</groupId>
<artifactId>cl1-jpa-blas-juan</artifactId>
<version>1.0-SNAPSHOT</version>
<properties>
<maven.compiler.source>23</maven.compiler.source>
<maven.compiler.target>23</maven.compiler.target>
<project.build.sourceEncoding>UTF-8</project.build.sourceEncoding>
</properties>
</project>

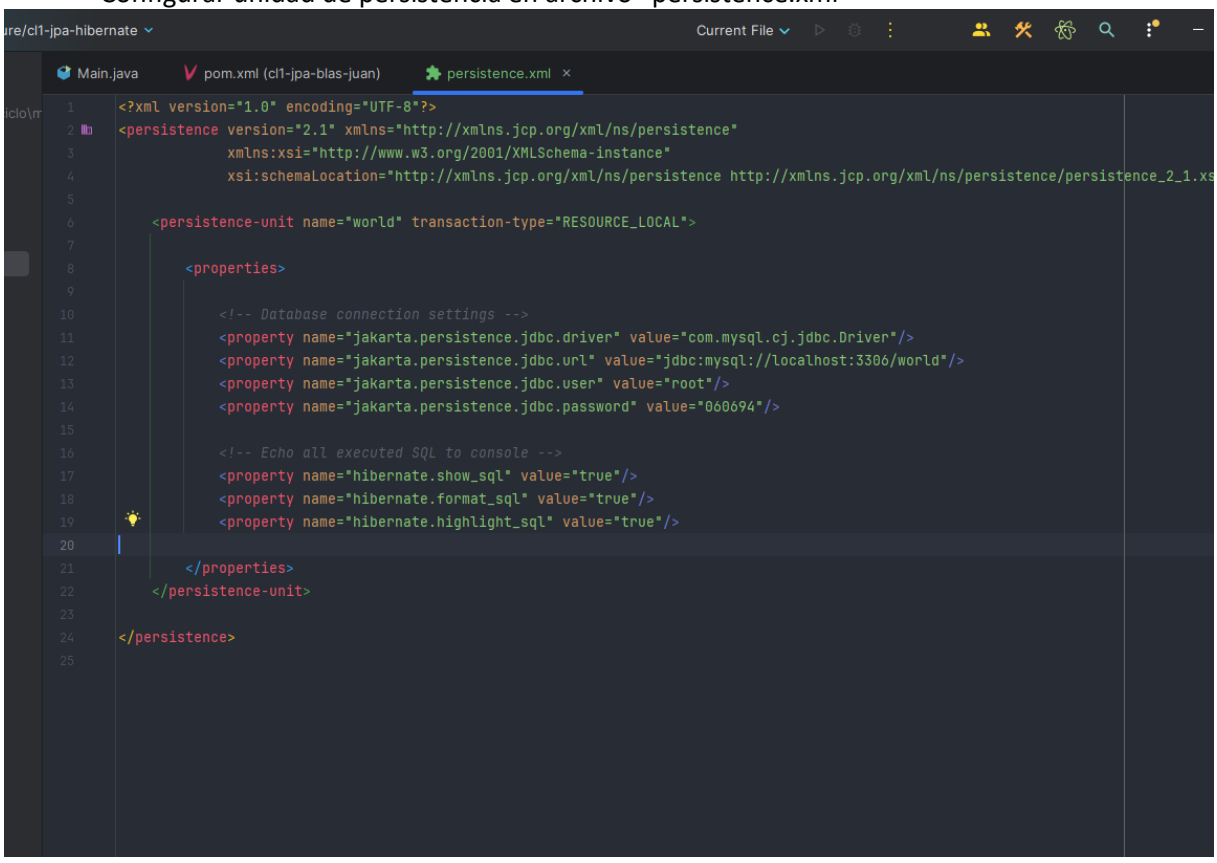
- Configurar dependencias en el pom.xml
    - Hibernate (6.6.2.Final)
    - Driver de MySQL (9.1.0)



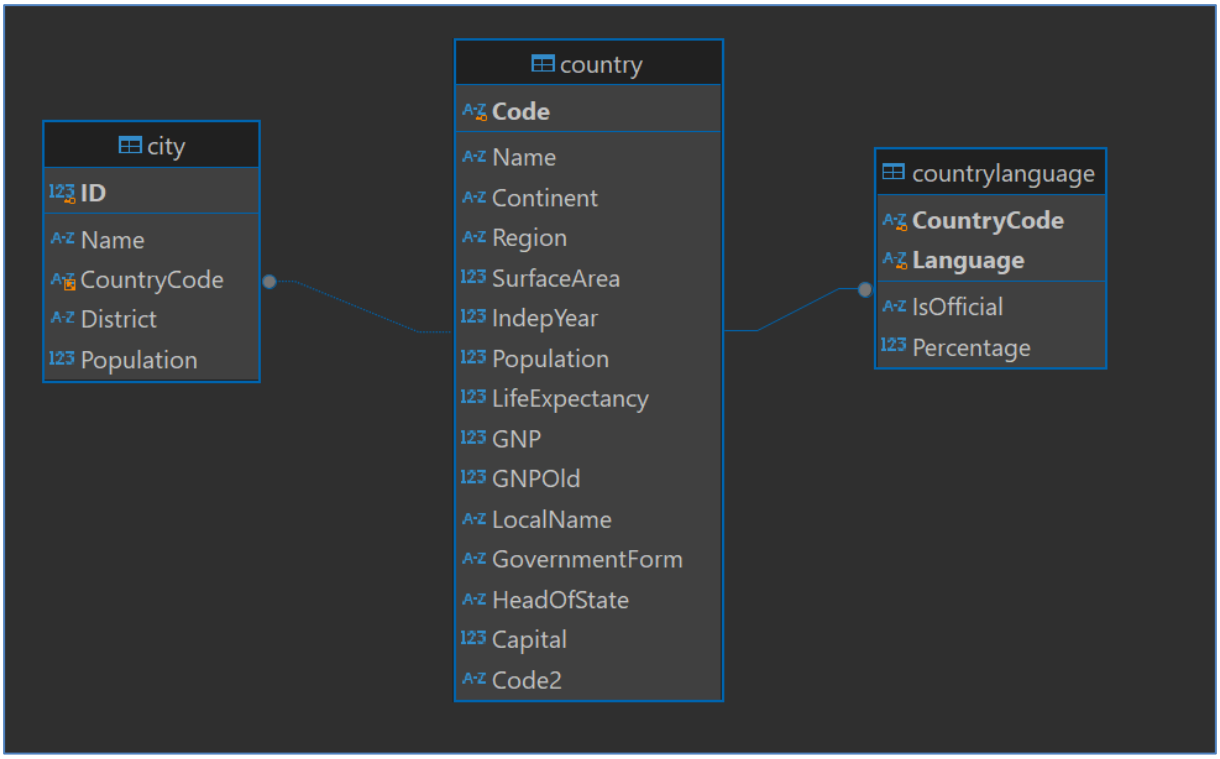




- Configurar unidad de persistencia en archivo “persistence.xml”

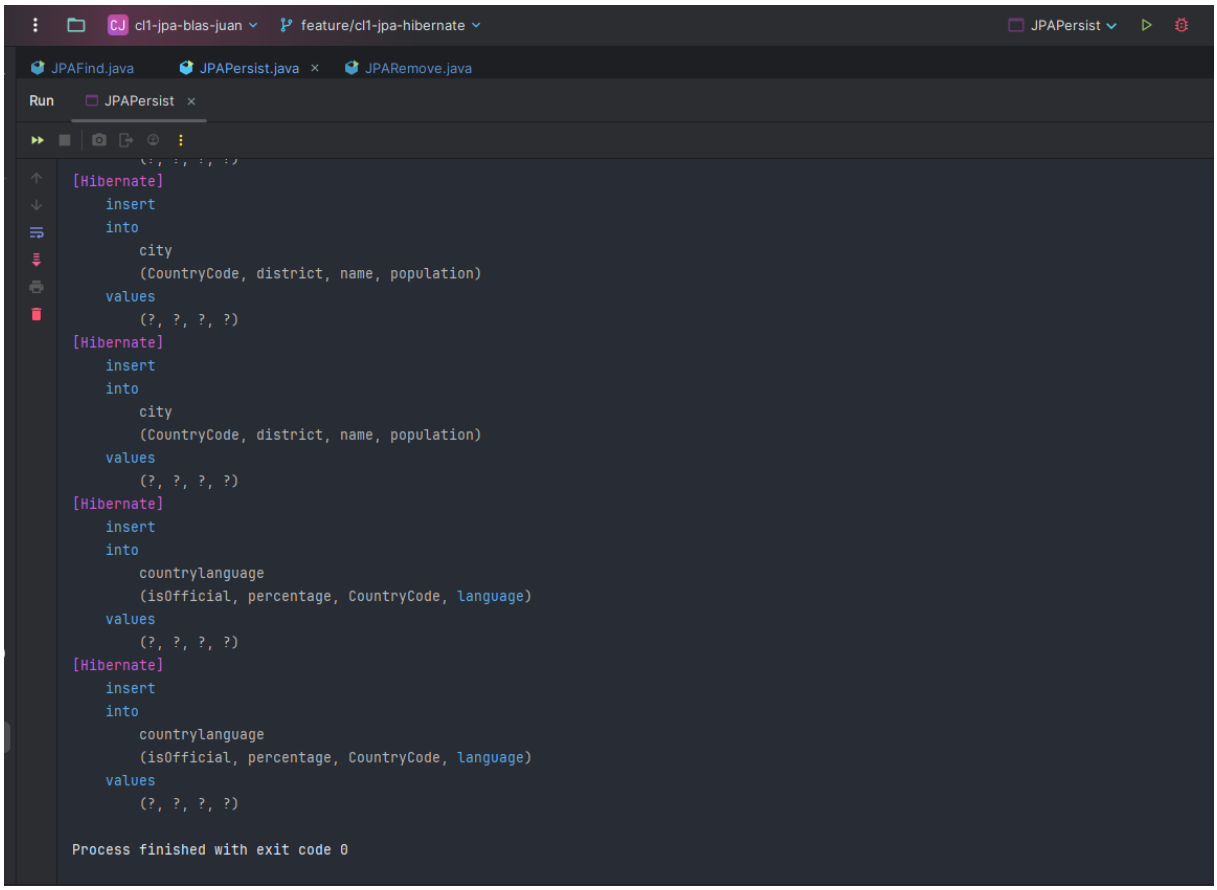


- Crear las entidades correspondientes a las siguientes tablas de la bd “world”:

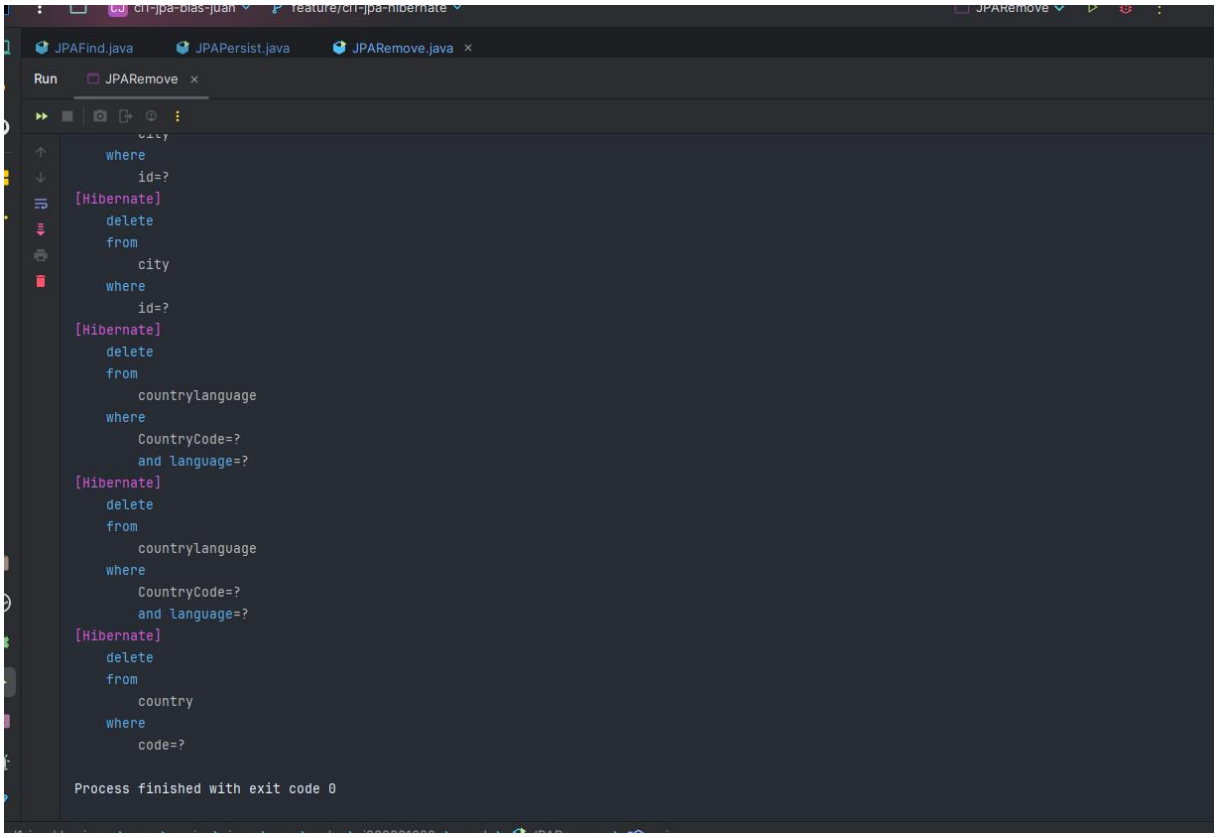


  - Mapear las 3 entidades de forma tradicional (Sin Lombok).
  - Definir la estrategia de generación correcta para los PKs.
  - Considerar el mapeo de las relaciones de forma bidireccional.

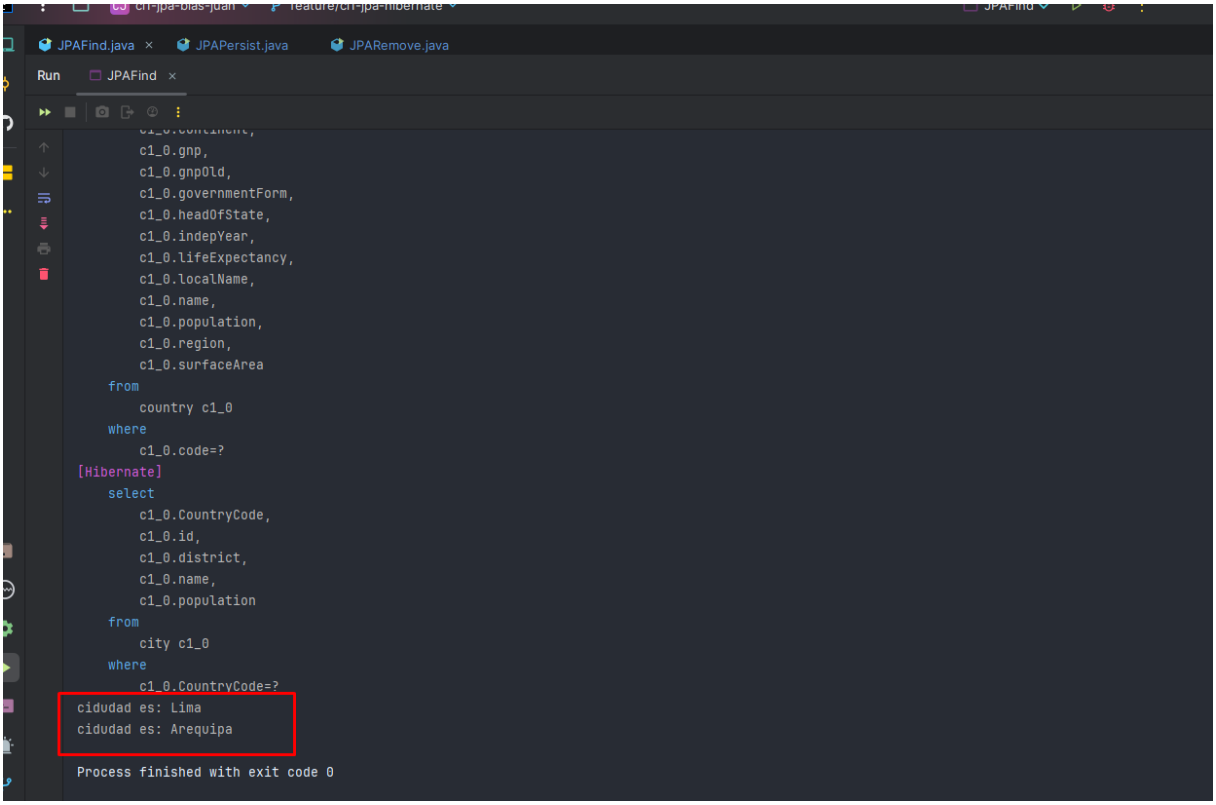
- Crear una clase “JPAPersist” y en ella registre un país imaginario, que tenga 3 ciudades y 2 lenguajes nativos. Sólo debe realizar un llamado al método “persist”.



- Crear una clase “JPARemove” y en ella elimine el país imaginario (Previamente creado). La eliminación debe eliminar el rastro de sus 3 ciudades y 2 lenguajes nativos. Sólo debe realizar un llamado al método “remove”. Considere, no afectar la funcionalidad de la clase “JPAPersist”.

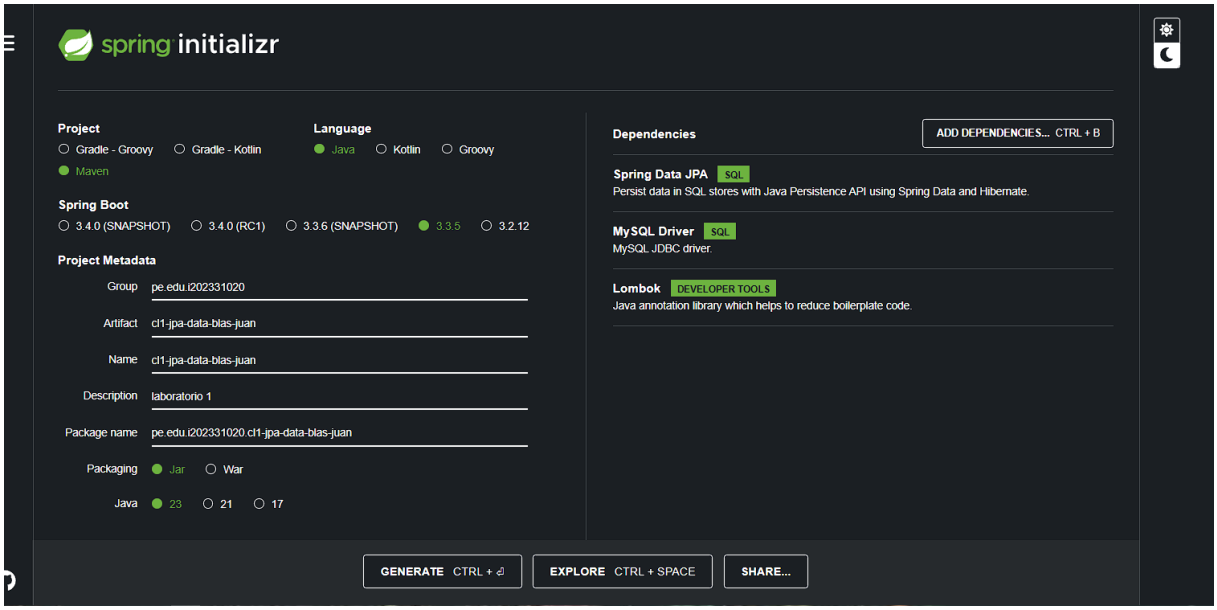


- Crear una clase “JPAFind” y en ella realice una sola consulta a la entidad “Country” (Busque el código “PER” usando find) y en base al resultado imprima el nombre de las ciudades peruanas con población > 700k. Deberá usar una función lambda para discriminar el resultado.

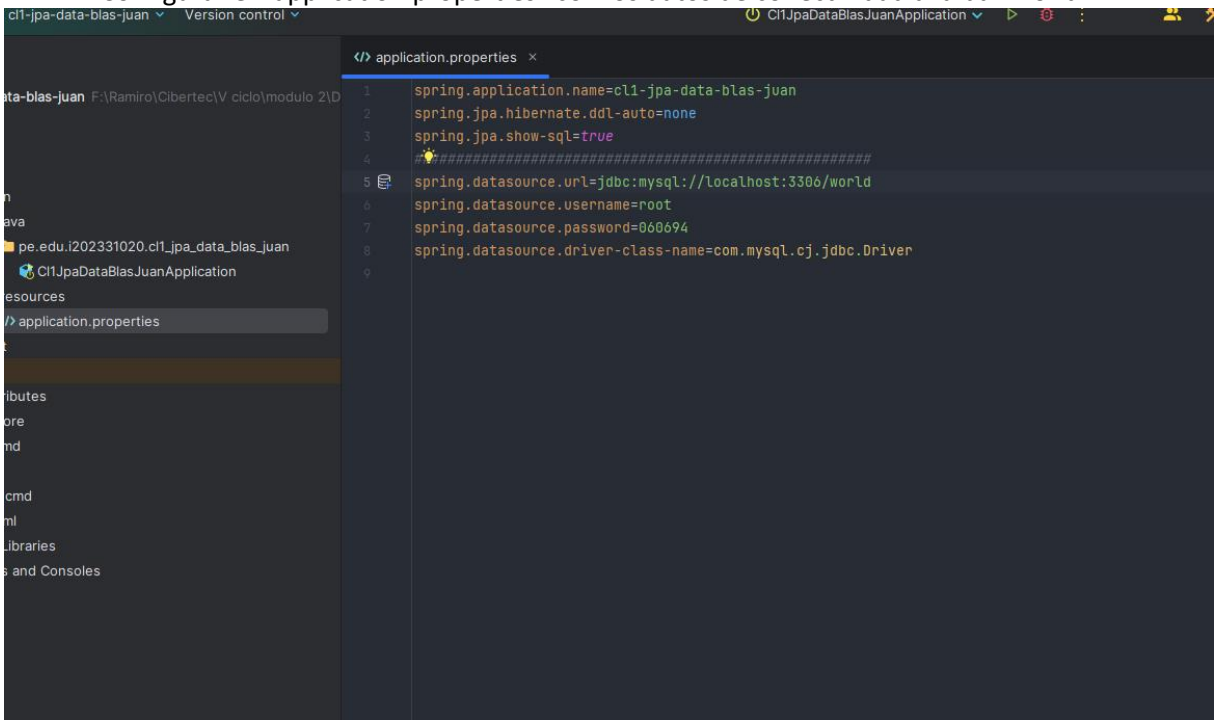


Parte 03 Proyecto Spring Data JPA (25%)

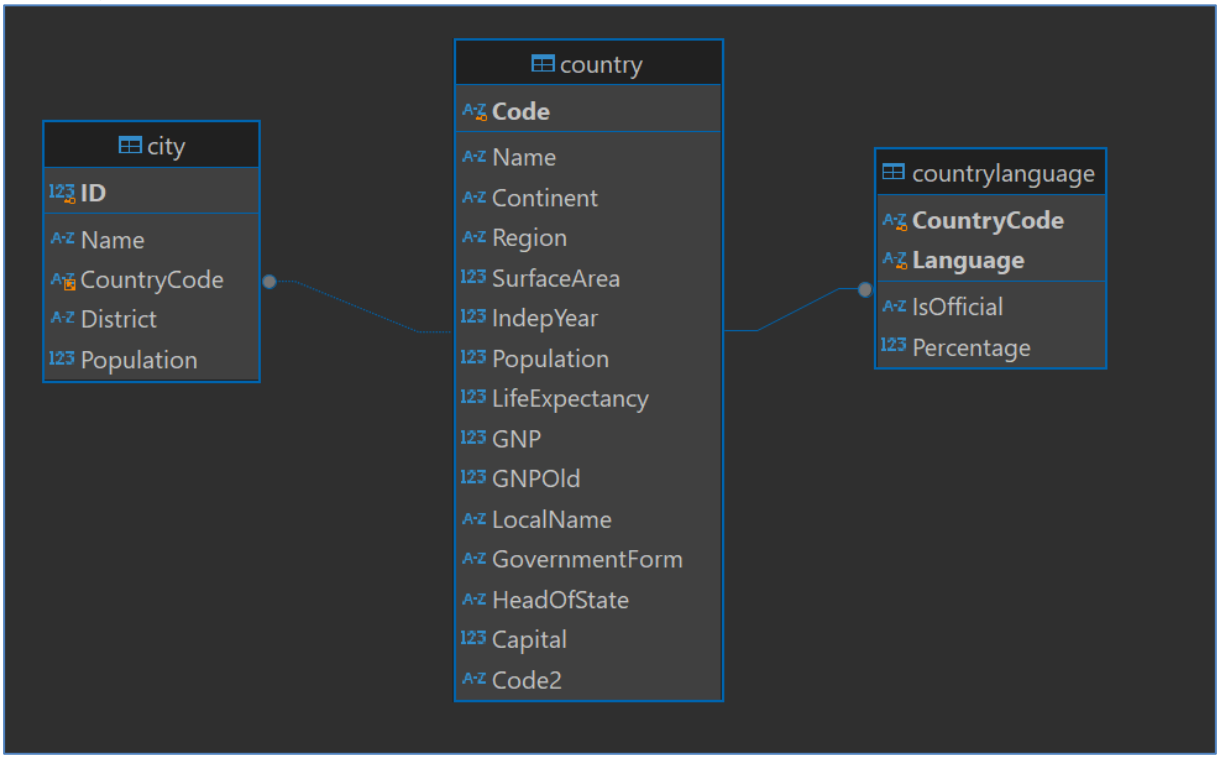
- Generar un proyecto con Spring Data JPA desde https://start.spring.io/ con las siguientes características:
    - Project: Maven
    - Language: Java
    - Spring Boot: 3.3.5
    - Group: pe.edu.<codigoEstudiante>
    - Artifact: cl1-jpa-data-<apellidoPaterno-primerNombre>
    - Packaging: Jar
    - Java: 23
    - Dependencies:
        - Spring Data JPA
        - MySQL Driver
        - Lombok



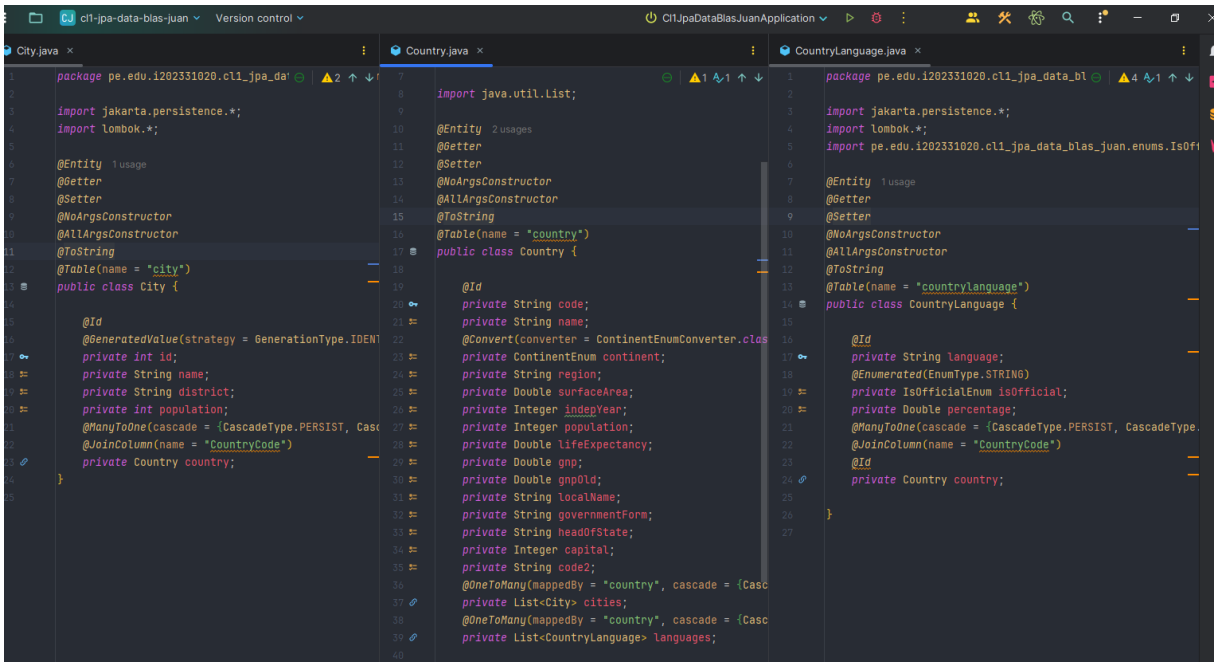
- Configurar el “application.properties” con los datos de conectividad a la bd “world”.



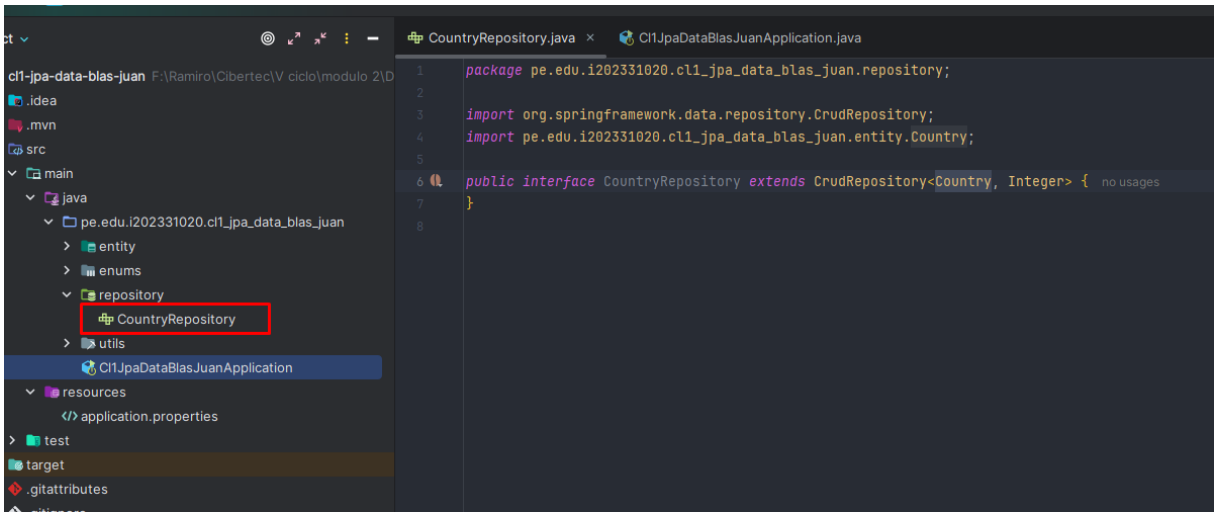
- Crear las entidades correspondientes a las siguientes tablas (Las mismas del proyecto anterior):



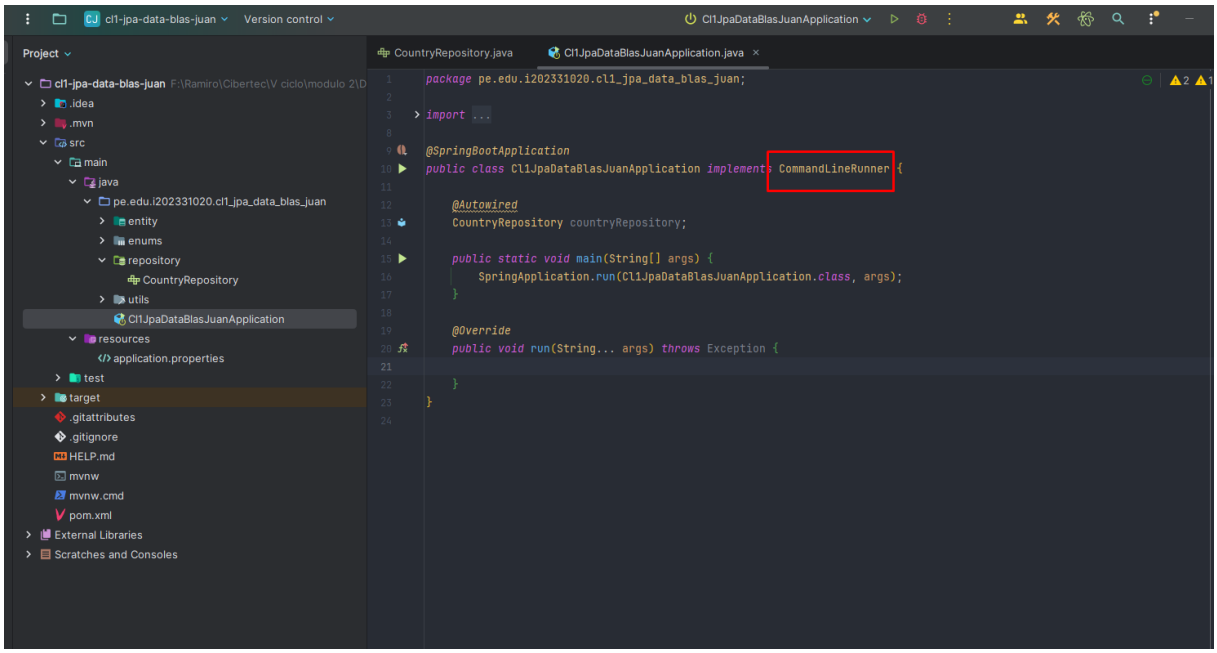
  - Mapear las 3 entidades usando Lombok.
  - Definir la estrategia de generación correcta para los PKs.
  - Considerar el mapeo de las relaciones de forma bidireccional.



- Crear una interfaz “CountryRepository” que extienda de “CrudRepository”.



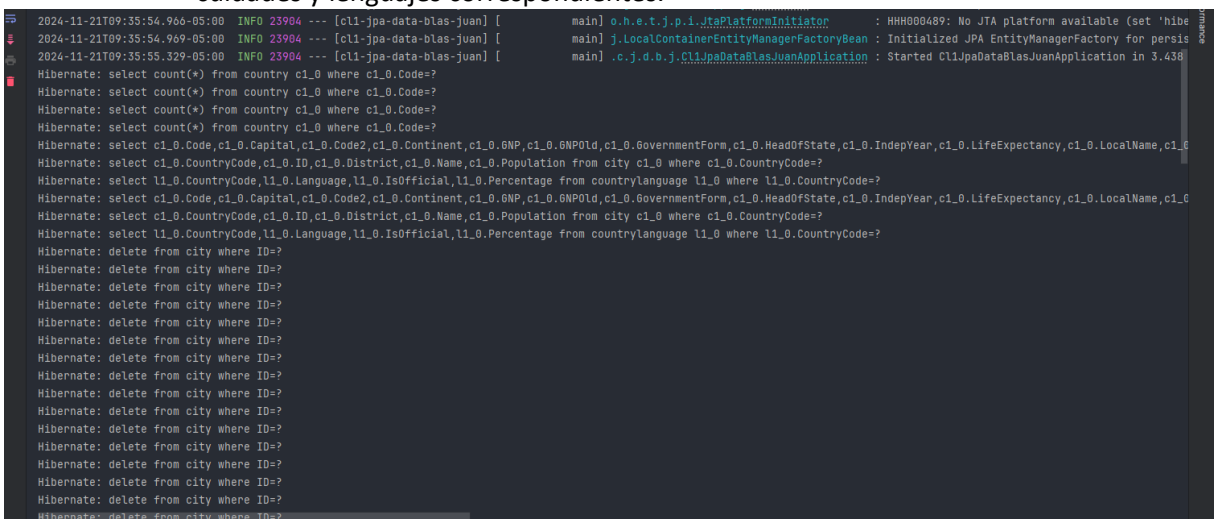
- Implementar el método “run” definido en la interfaz “CommandLineRunner” desde la clase principal del proyecto de Spring Boot.



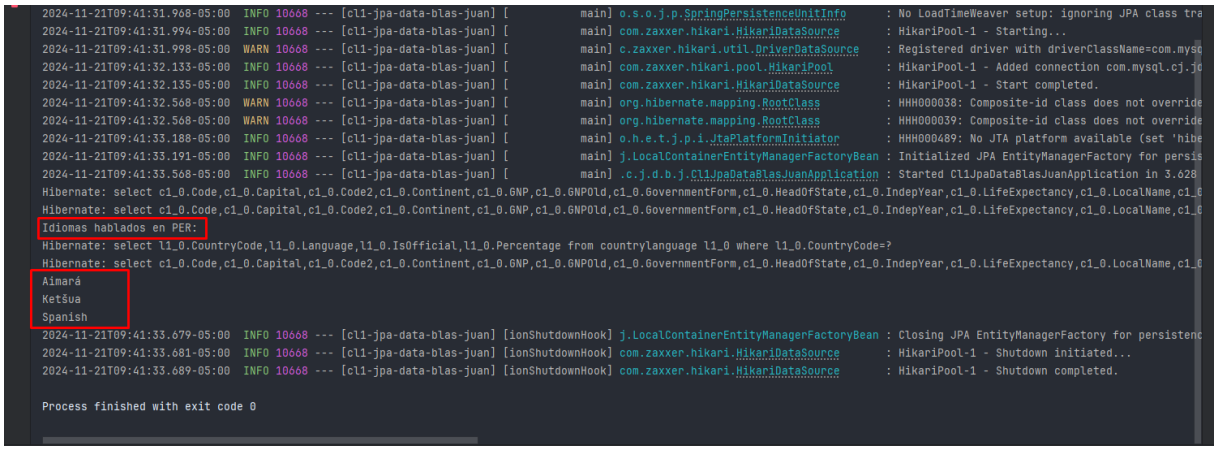
- Deberá implementar las siguientes 3 consultas:

  - **ifPresentOrElse()**
    Imprimir en la terminal los nombres de los lenguajes que se hablan en el país “ARG” (Angentina). En caso de no obtener resultado, deberá imprimir los nombres de los lenguajes del país “PER” (Perú).

- **deleteAllById()**
  Eliminar 2 países: “COL” y “ARG”. La eliminación deberá ser cascada y borrará sus cuidades y lenguajes correspondientes.



- Volver a ejecutar la primera consulta, pues al eliminar “ARG”, deberá ejecutarse el flujo alterno. (Deberá restaurar la BD desde la terminal en cada prueba)



Parte 04 Gestión de conexiones (25%)

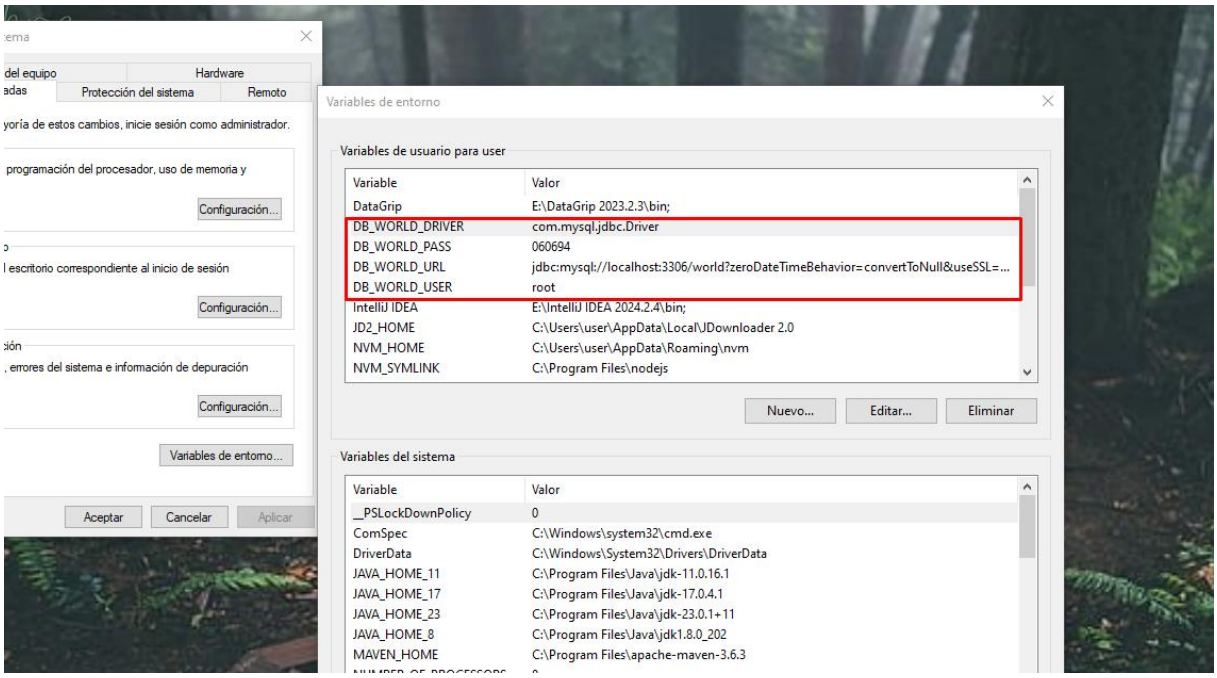
- Crear una clase de configuración denominada “ConexionesConfig.java”, en ella deberá implementar una personalización del DataSource de HikariCP. Considere los siguientes valores para el pool de conexiones:
  - MaximunPoolSize: 30
  - MinimumIdle: 4
  - IdleTimeout: 4 minutos

- ConnectionTimeout: 45 segundos

```
    @Value("${DB_WORLD_USER}")
    private String dbWorldUser;
    @Value("${DB_WORLD_PASS}")
    private String dbWorldPass;
    @Value("${DB_WORLD_DRIVER}")
    private String dbWorldDriver;

    @Bean  new *
    public HikariDataSource hikariDataSource() {
        HikariConfig config = new HikariConfig();

        config.setJdbcUrl(dbWorldUrl);
        config.setUsername(dbWorldUser);
        config.setPassword(dbWorldPass);
        config.setDriverClassName(dbWorldDriver);

        config.setMaximumPoolSize(30);
        config.setMinimumIdle(4);
        config.setIdleTimeout(240000);
        config.setConnectionTimeout(45000);

        System.out.println("###### HikariCP initialized ######");

        return new HikariDataSource(config);
    }
}
```

- Las credenciales del DataSource (Parámetros de conexión a la BD) no deben ser visibles en el código fuente. Para ello deberá crear las siguientes variables de entorno:
  - DB_WORLD_URL
  - DB_ WORLD_USER
  - DB_ WORLD_PASS
  - DB_ WORLD_DRIVER

```
import org.springframework.context.annotation.Bean;
import org.springframework.context.annotation.Configuration;

@Configuration  new *
public class ConexionesConfig {

    @Value("${DB_WORLD_URL}")
    private String dbWorldUrl;
    @Value("${DB_WORLD_USER}")
    private String dbWorldUser;
    @Value("${DB_WORLD_PASS}")
    private String dbWorldPass;
    @Value("${DB_WORLD_DRIVER}")
    private String dbWorldDriver;

    @Bean  new *
    public HikariDataSource hikariDataSource() {
        HikariConfig config = new HikariConfig();

        config.setJdbcUrl(dbWorldUrl);
        config.setUsername(dbWorldUser);
        config.setPassword(dbWorldPass);
        config.setDriverClassName(dbWorldDriver);

        config.setMaximumPoolSize(30);
        config.setMinimumIdle(4);
        config.setIdleTimeout(240000);
        config.setConnectionTimeout(45000);

        System.out.println("###### HikariCP initialized ######");

        return new HikariDataSource(config);
    }
```



- Luego de configurar el DataSource, ¿Es necesario proporcionar las credenciales desde el archivo “application.properties”?. ¿Por qué? Explique con sus propias palabras.

  . ya no es necesario brindar las credenciales en el properties porque esta configurado desde las variables de entorno, para mayor seguridad de datos.

  Repositorio1 : https://github.com/RamiroBlas/cl1-jpa-blas-juan
  Repositorio 2: https://github.com/RamiroBlas/cl1-jpa-data-blas-juan