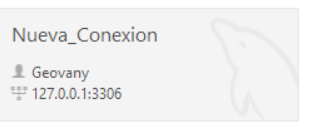
¿Cómo realizo mi conexión a la base de datos?

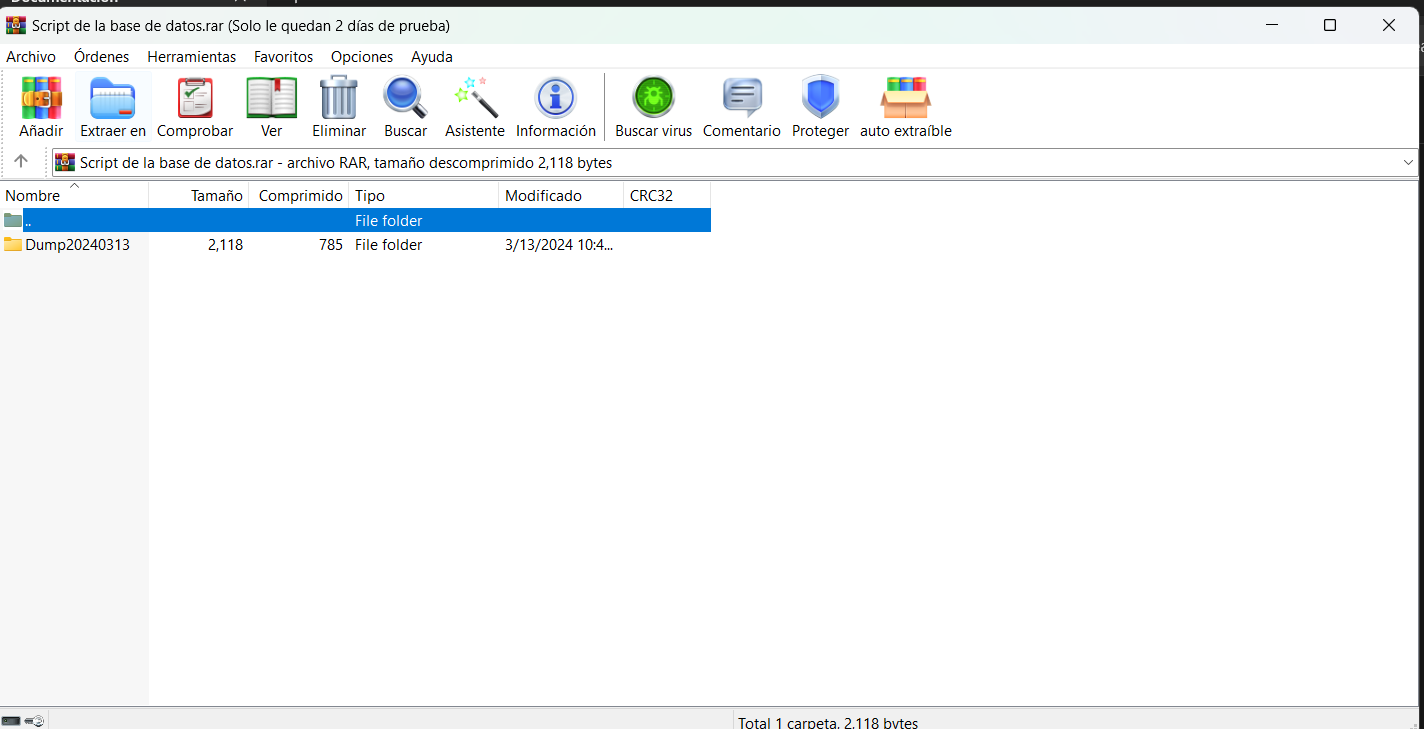
1. Creamos un nuevo usuario para no trabajar en root



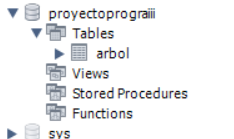
Link de video guía:

<https://www.youtube.com/watch?v=SB36NEQMzVk>

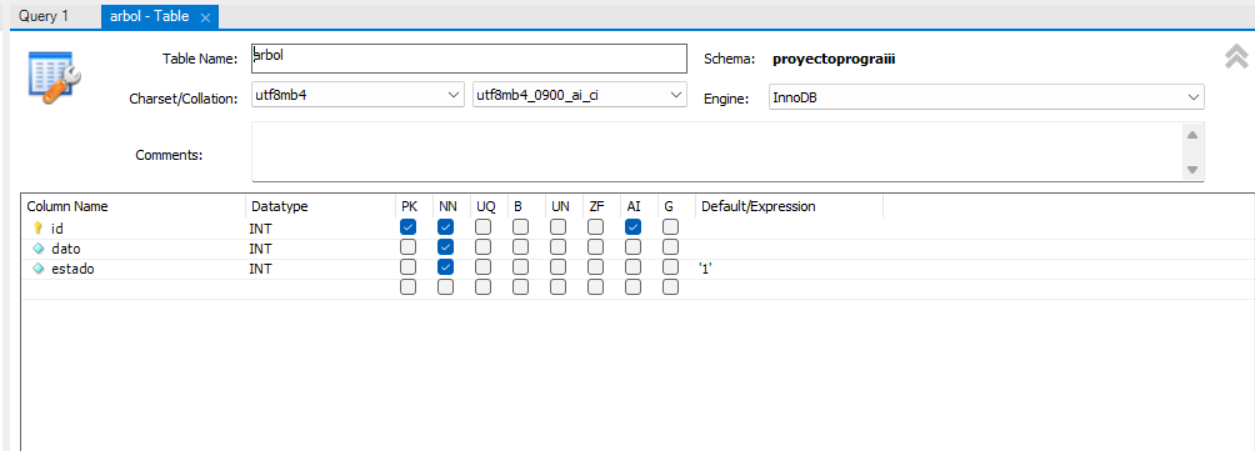
1. Extraemos el script y lo importamos a workbench.



1. Ya importado nos quedará así



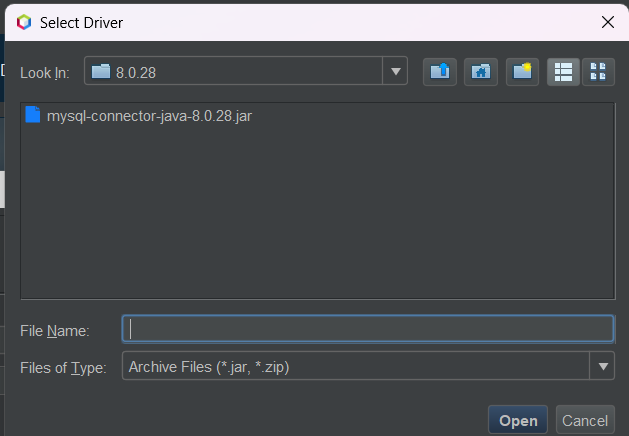
1. Nuestra única tabla es árbol

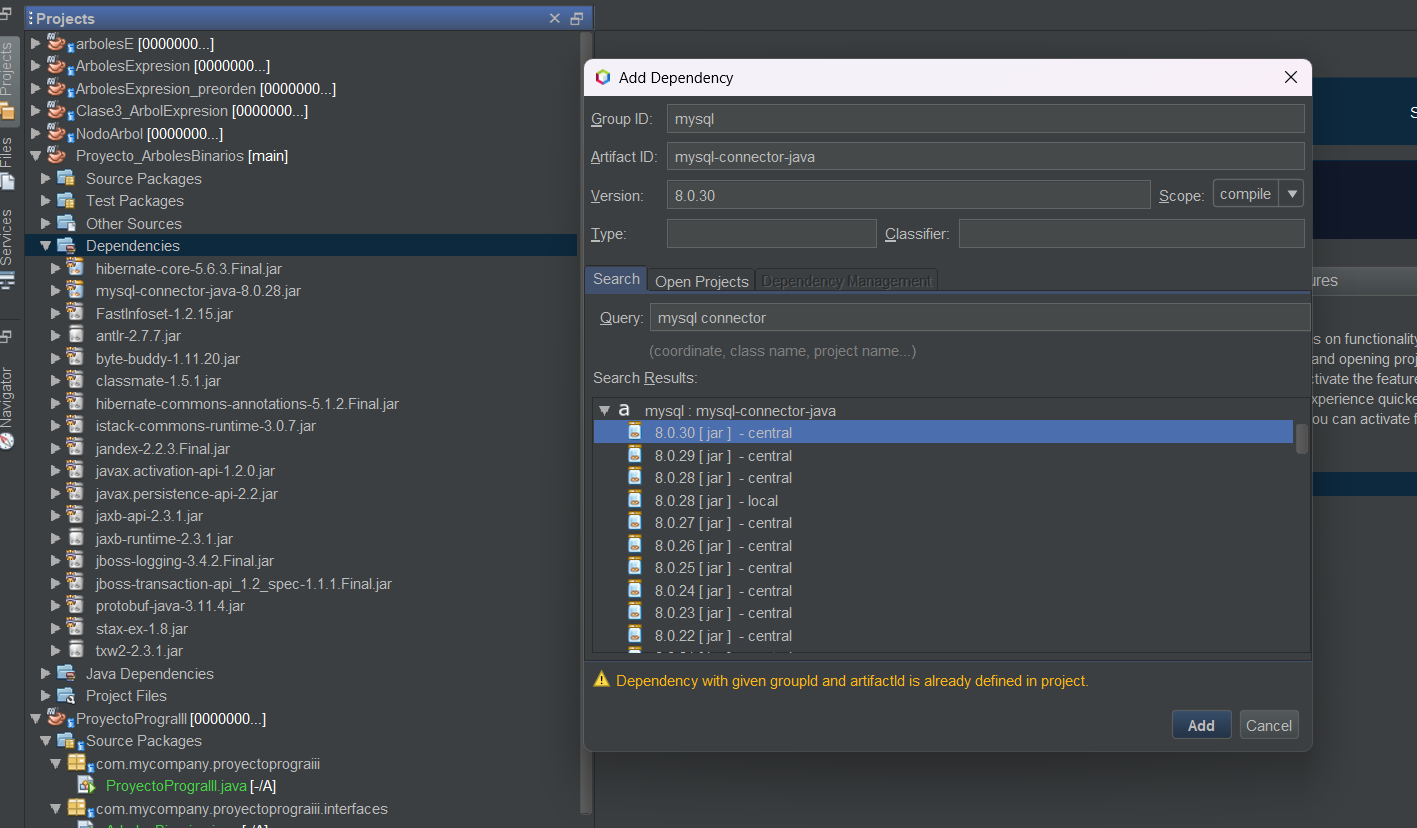


id: Llave primaria que guardará nuestros registros de forma única, es autoincrementadle.

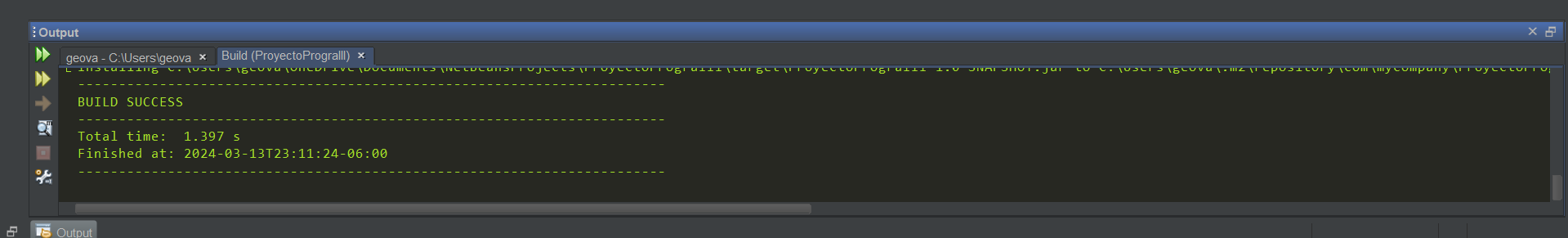
dato: Es un entero, el cual se insertará por cada uno que se lea del archivo de texto.

estado: Este representa el estado del registro, 1=activo, 0=inactivo, lo cual es buena práctica para no perder registros y auditorias de sistemas.

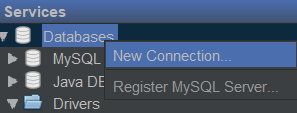
1. Debemos de descargar el driver de mysql
2. Database – new connection
3. Seleccionamos el driver



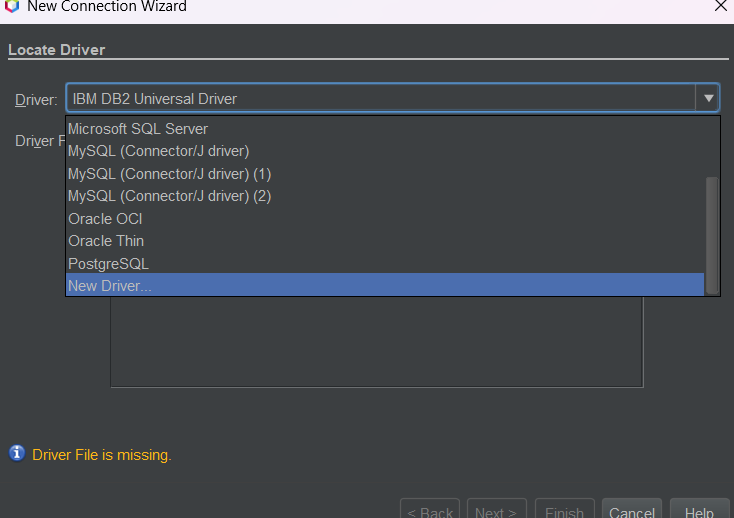
1. Construimos el proyecto

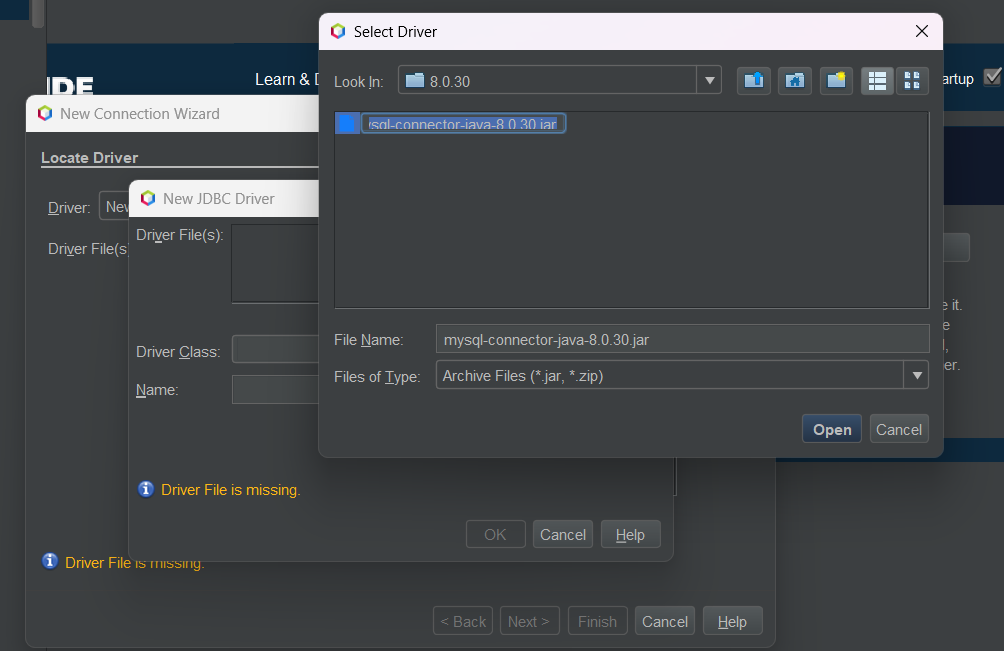


1. Ahora nos vamos a services – Databases – New Connection

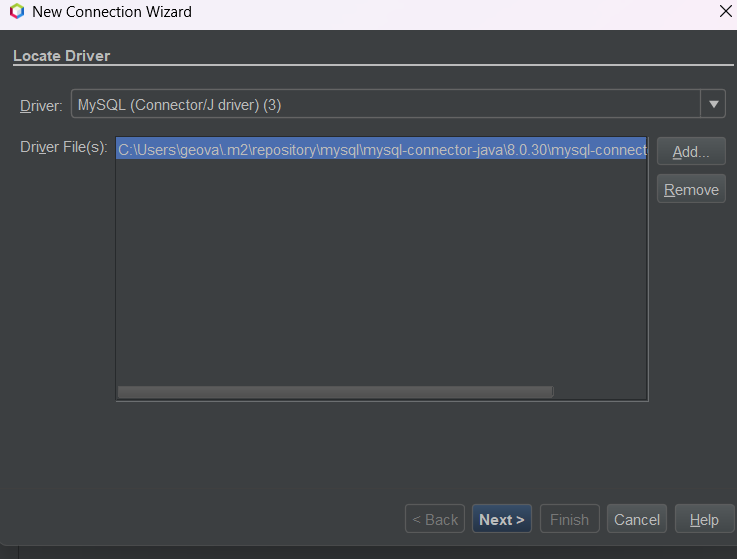


1. Agregamos el driver que acabamos de descargar

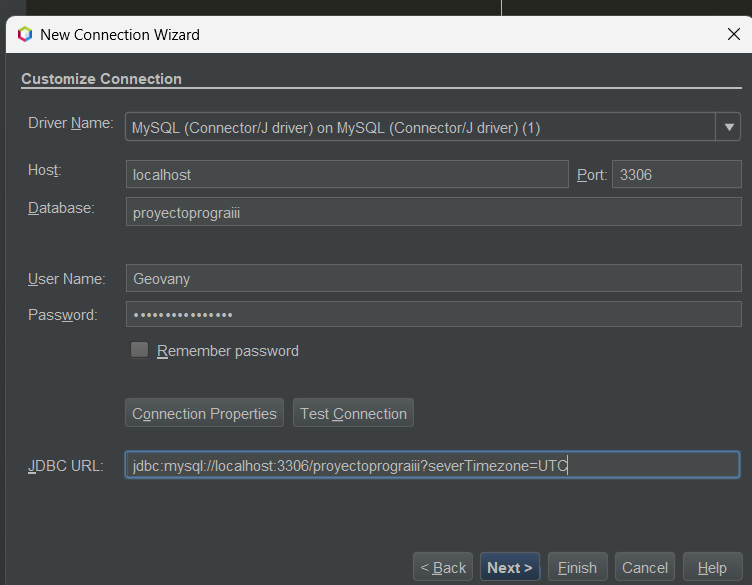




1. Le damos siguiente

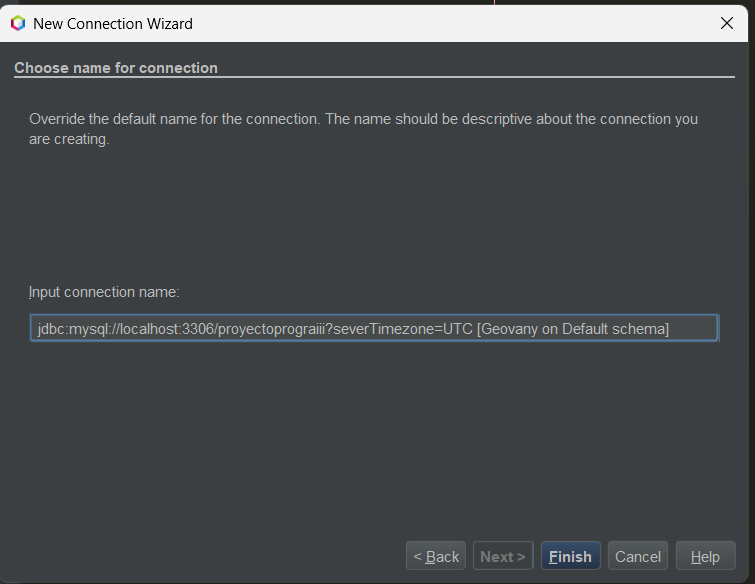


1. Llenamos datos

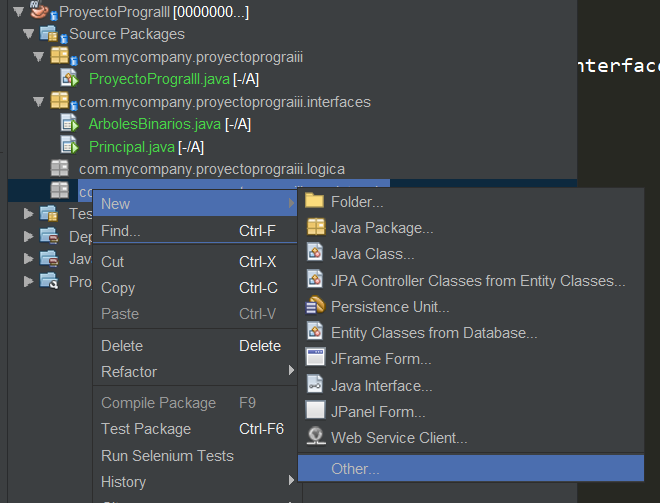


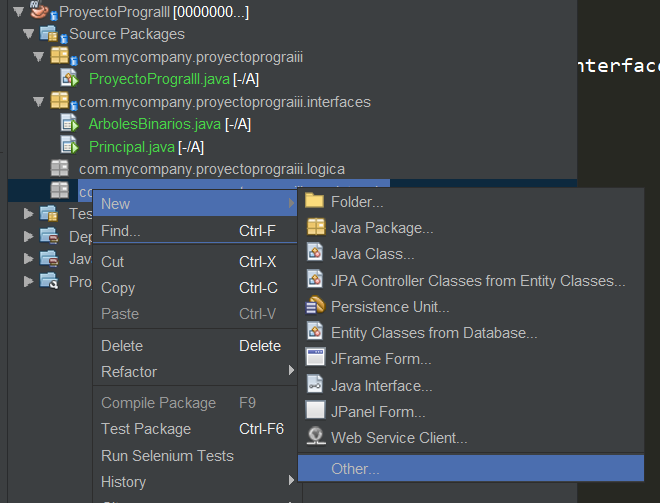
El serverTimeZone nos sirve para que no hayan errores con los horarios.

1. Avanzamos y le damos finish

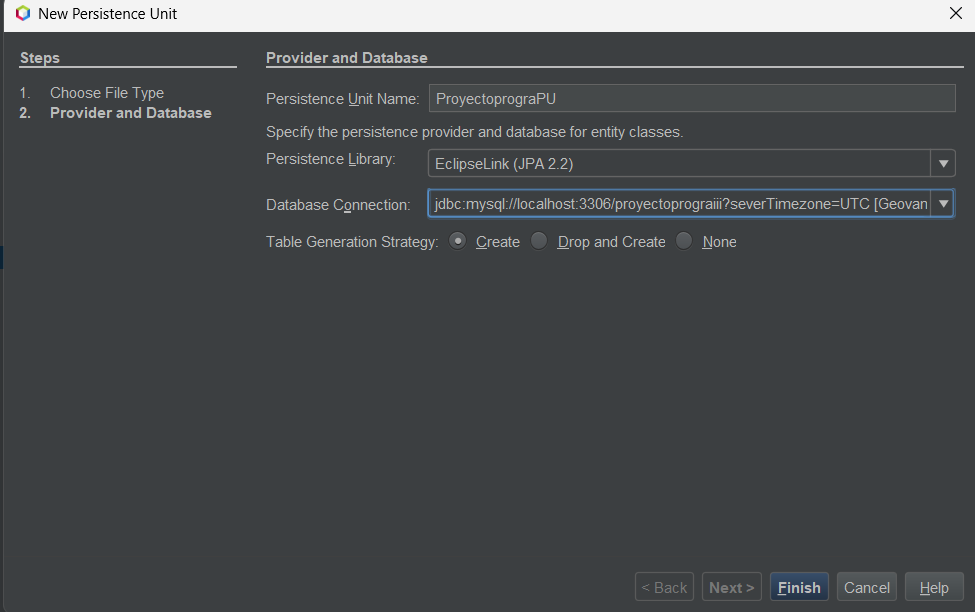


1. Ahora crearemos el archivo de persistence

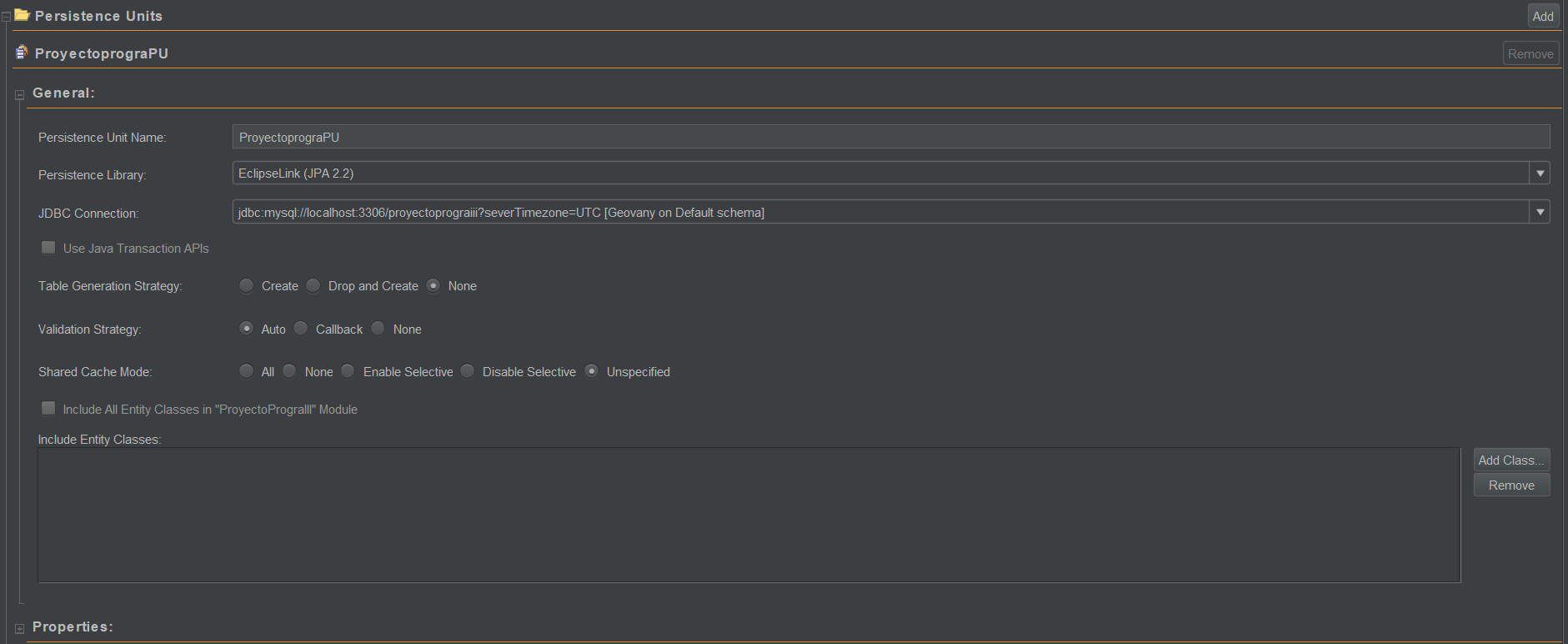




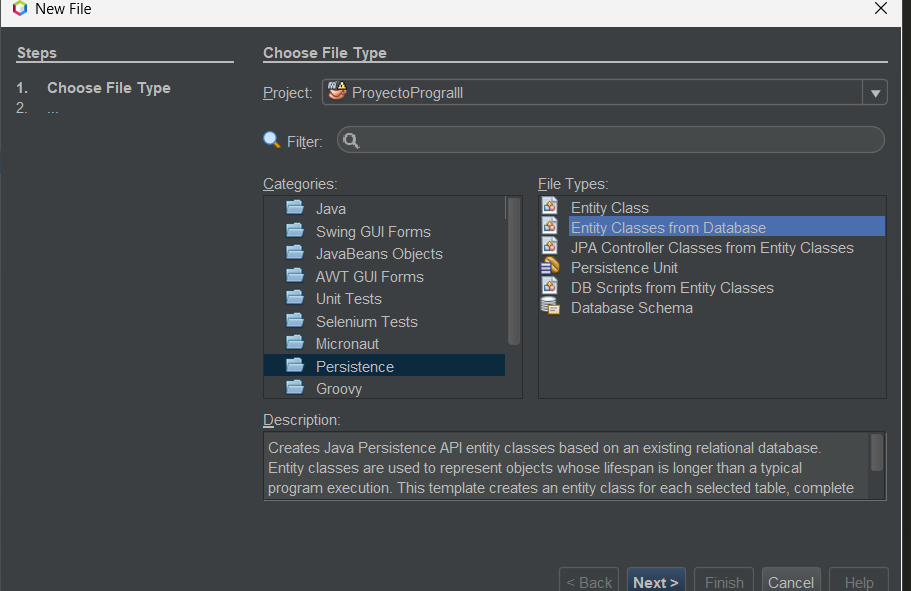
1. Llenamos los datos



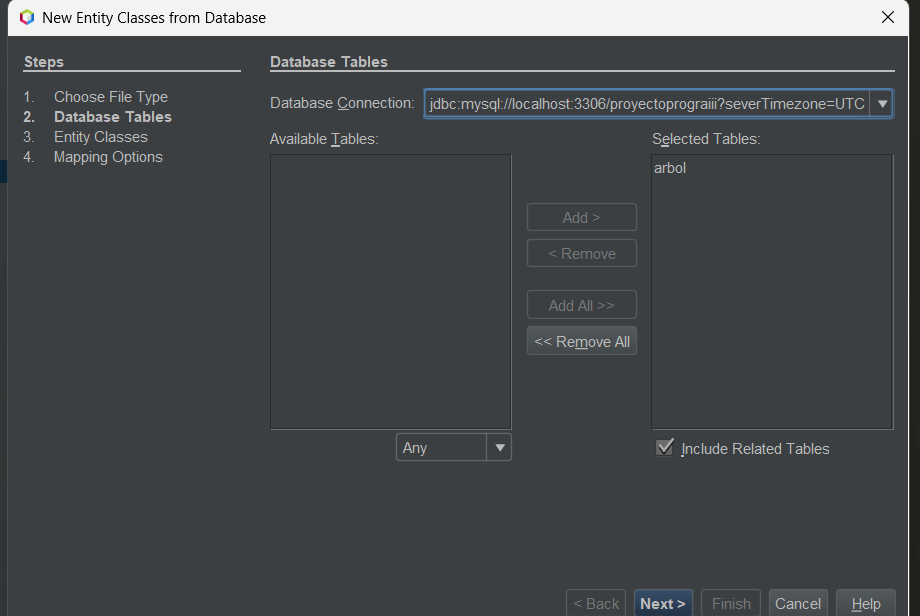
1. Se nos crea el persistence



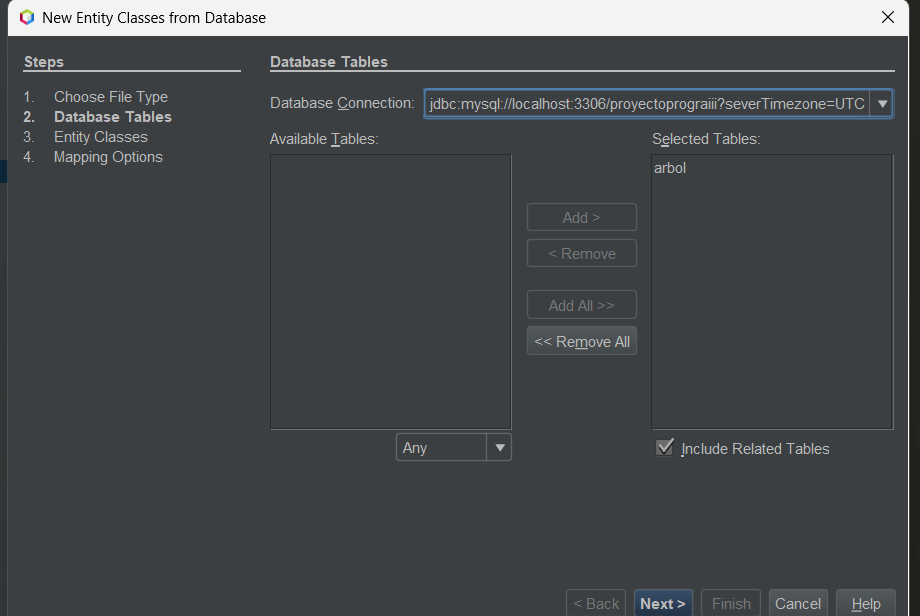
1. Crearemos la entity classes



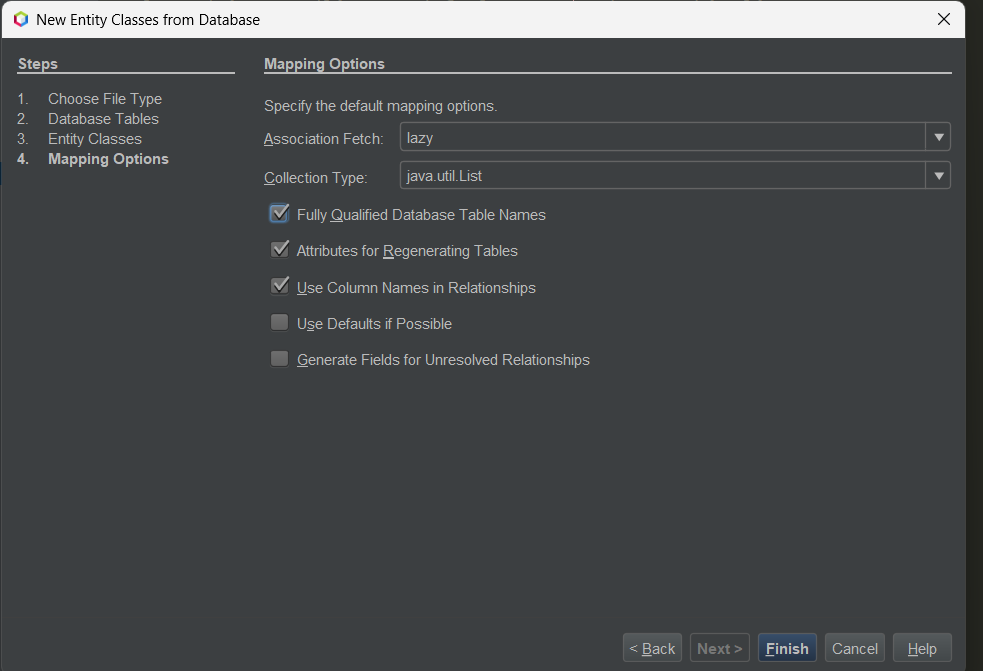
1. Agregamos la tabla.



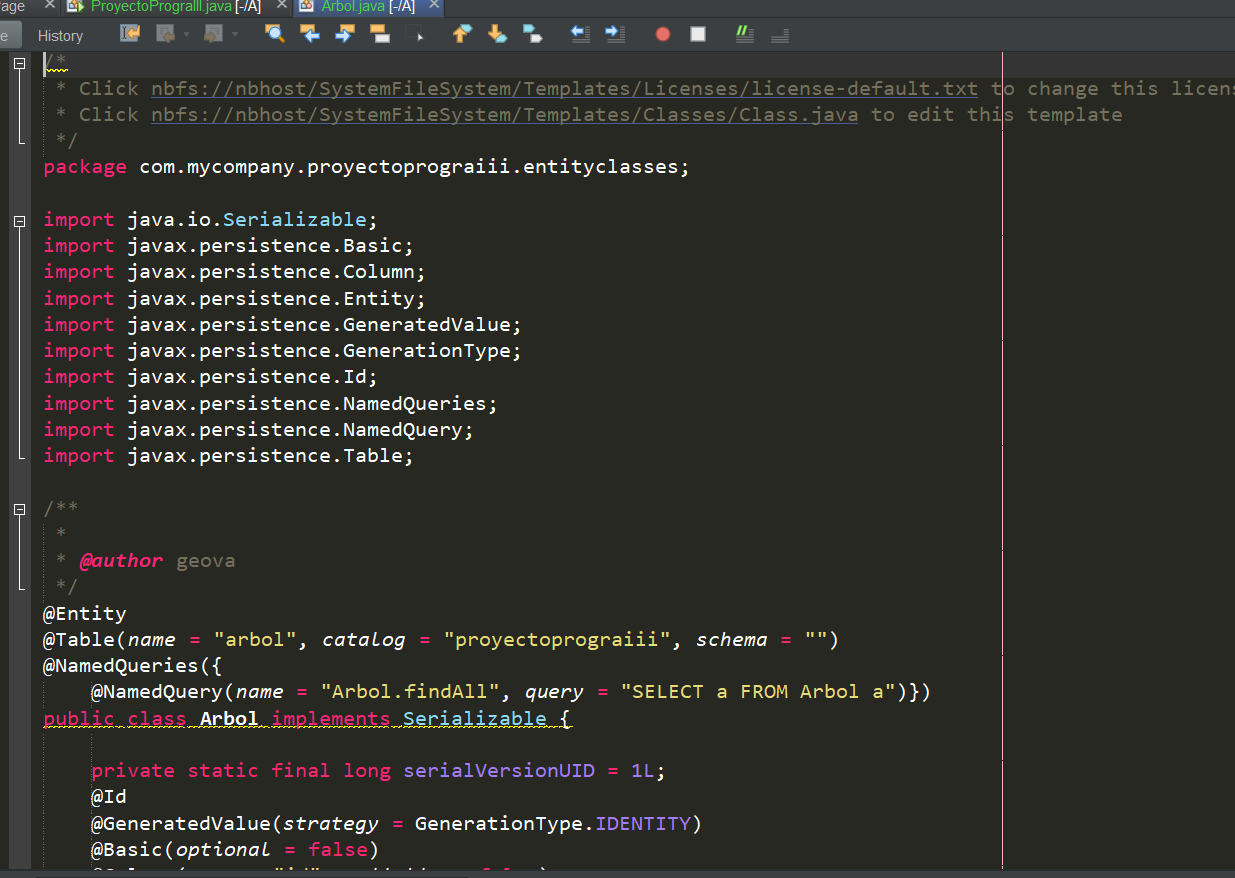
1. Seguimos avanzando



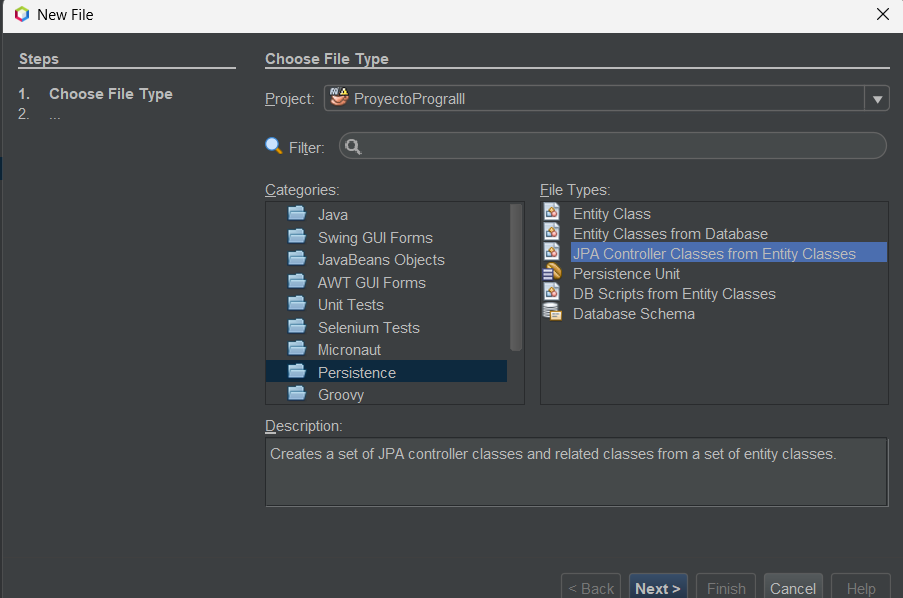
1. Asociación: lazy y Colletion type: list, para usar el método de lista



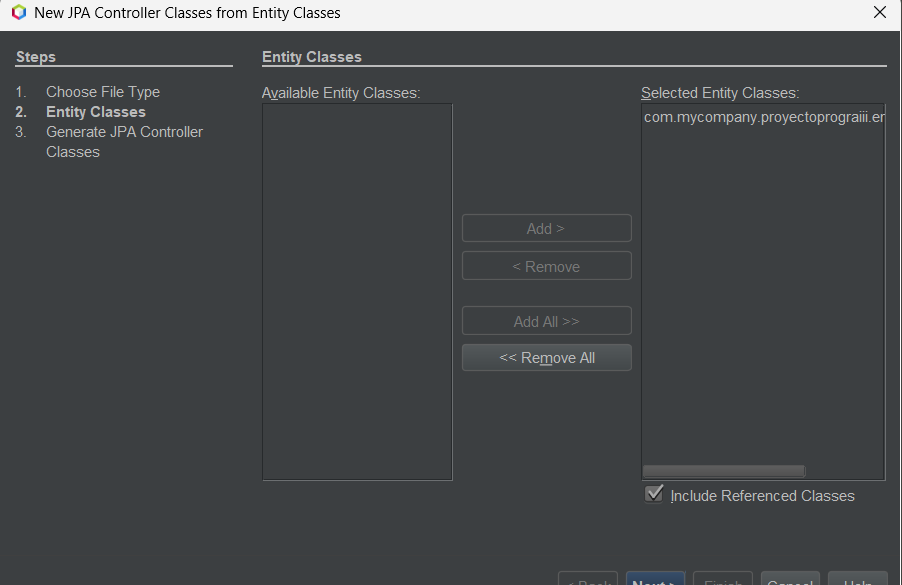
1. Ya la tenemos



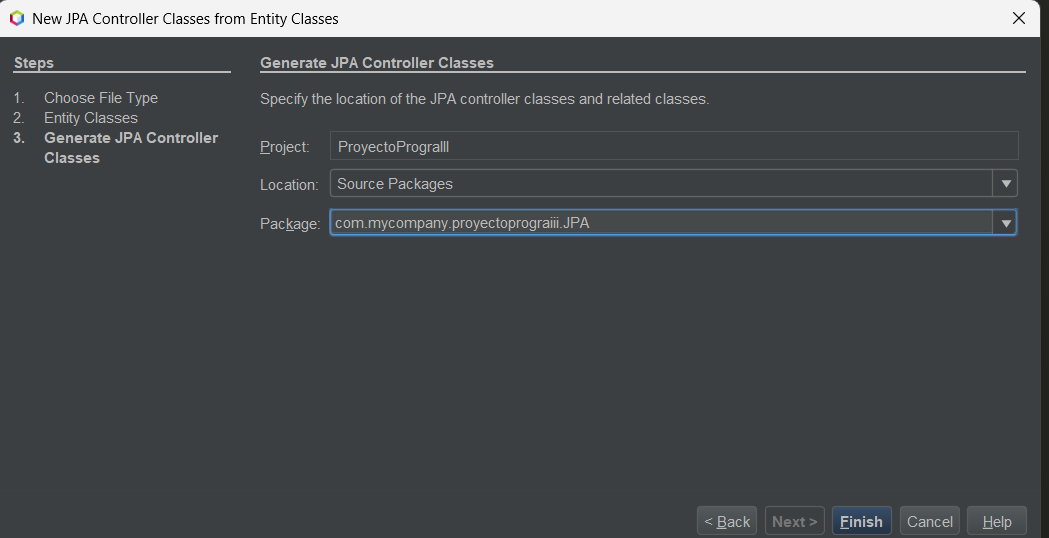
1. Ahora crearemos el jpa



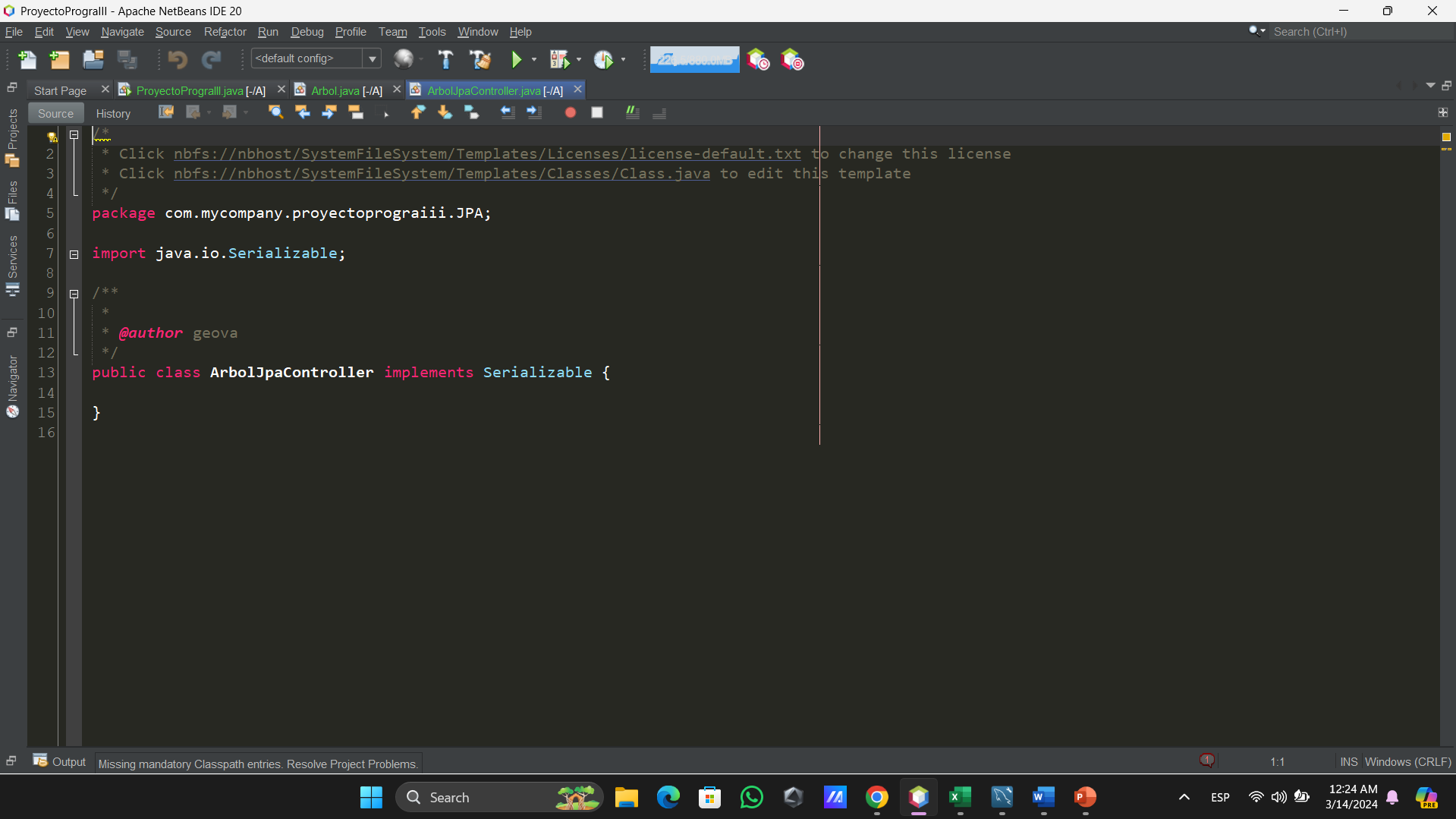
1. Pasamos las clases



1. Configuramos donde se guardará



1. Creado el JPA



1. Pegar este código

package com.mycompany.proyectoprograiii.JPAC;

import com.mycompany.proyectoprograiii.entityclasses.Arbol;

import javax.persistence.EntityManagerFactory;

import javax.persistence.EntityTransaction;

import javax.persistence.Query;

import java.util.List;

import javax.persistence.EntityManager;

public class ArbolJpaController {

private final EntityManagerFactory entityManagerFactory;

public ArbolJpaController(EntityManagerFactory entityManagerFactory) {

this.entityManagerFactory = entityManagerFactory;

}

public void create(Arbol arbol) {

EntityManager entityManager = entityManagerFactory.createEntityManager();

EntityTransaction transaction = entityManager.getTransaction();

try {

transaction.begin();

entityManager.persist(arbol);

transaction.commit();

} catch (Exception ex) {

if (transaction.isActive()) {

transaction.rollback();

}

ex.printStackTrace();

} finally {

entityManager.close();

}

}

public Arbol find(Integer id) {

EntityManager entityManager = entityManagerFactory.createEntityManager();

try {

return entityManager.find(Arbol.class, id);

} finally {

entityManager.close();

}

}

public void update(Arbol arbol) {

EntityManager entityManager = entityManagerFactory.createEntityManager();

EntityTransaction transaction = entityManager.getTransaction();

try {

transaction.begin();

entityManager.merge(arbol);

transaction.commit();

} catch (Exception ex) {

if (transaction.isActive()) {

transaction.rollback();

}

ex.printStackTrace();

} finally {

entityManager.close();

}

}

public void delete(Integer id) {

EntityManager entityManager = entityManagerFactory.createEntityManager();

EntityTransaction transaction = entityManager.getTransaction();

try {

transaction.begin();

Arbol arbol = entityManager.find(Arbol.class, id);

if (arbol != null) {

entityManager.remove(arbol);

}

transaction.commit();

} catch (Exception ex) {

if (transaction.isActive()) {

transaction.rollback();

}

ex.printStackTrace();

} finally {

entityManager.close();

}

}

public List<Arbol> findAll() {

EntityManager entityManager = entityManagerFactory.createEntityManager();

try {

Query query = entityManager.createQuery("SELECT a FROM Arbol a", Arbol.class);

return query.getResultList();

} finally {

entityManager.close();

}

}

}