

**Una empresa dispone de una base de datos contiene las tablas con la información necesaria para su gestión. Las tablas son las siguientes:**

- **Tabla DEPT que contiene información de los diferentes departamentos que tiene la empresa. La clave principal DEPTNO.**
- **Tabla EMP que contiene la información de los diferentes empleados que tiene la empresa. Tiene como clave principal EMPNO y ajenas DEPTNO que relaciona con la tabla departamentos y MGR que establece la relación con la misma tabla mostrando ser jefe de.**

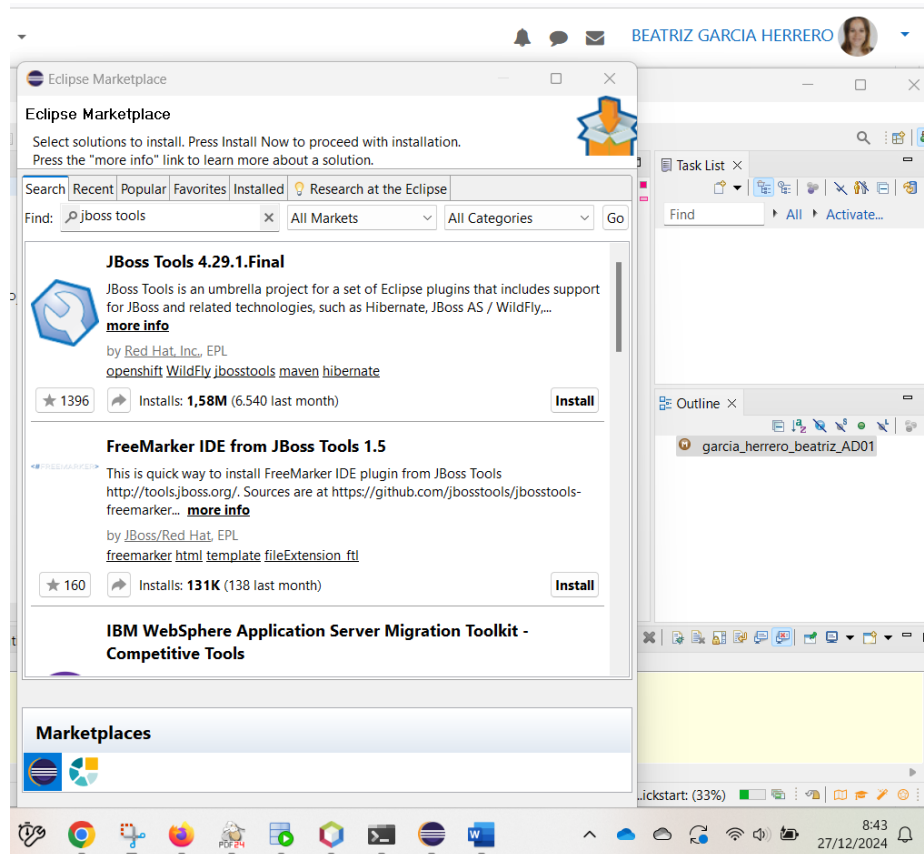
**La creación de las tablas y sus relaciones se dan en el fichero tablas.sql que se adjunta.**

**La base de datos que se utilizará será Oracle.**

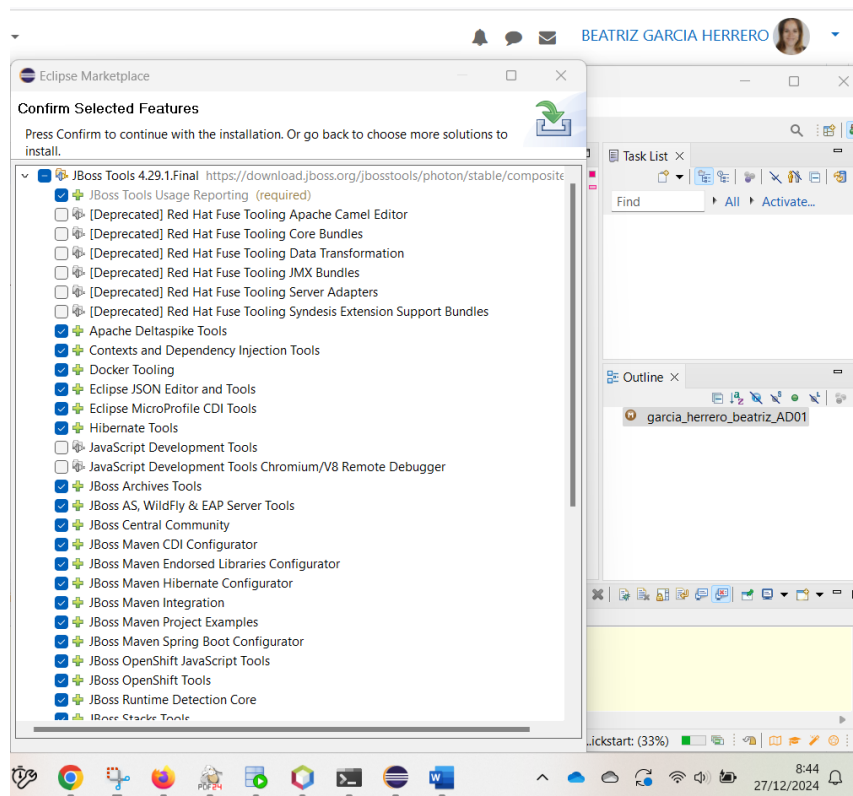
**Mapea las tablas utilizando Hibernate con NetBeans y realiza un proyecto Java llamado HibernateOracle que obtenga lo siguiente:**

- 1. Crea la base de datos.**
- 2. Configura y crea la ORM Hibernate.**
- 3. Realiza una inserción y un borrado sobre la tabla EMP.**
- 4. Obtener un listado sobre las tablas EMP y DEPT que visualice empno, ename, sal, dname y loc.**
- 5. Redactar un documento donde se explique el proceso seguido para la realización de la práctica.**

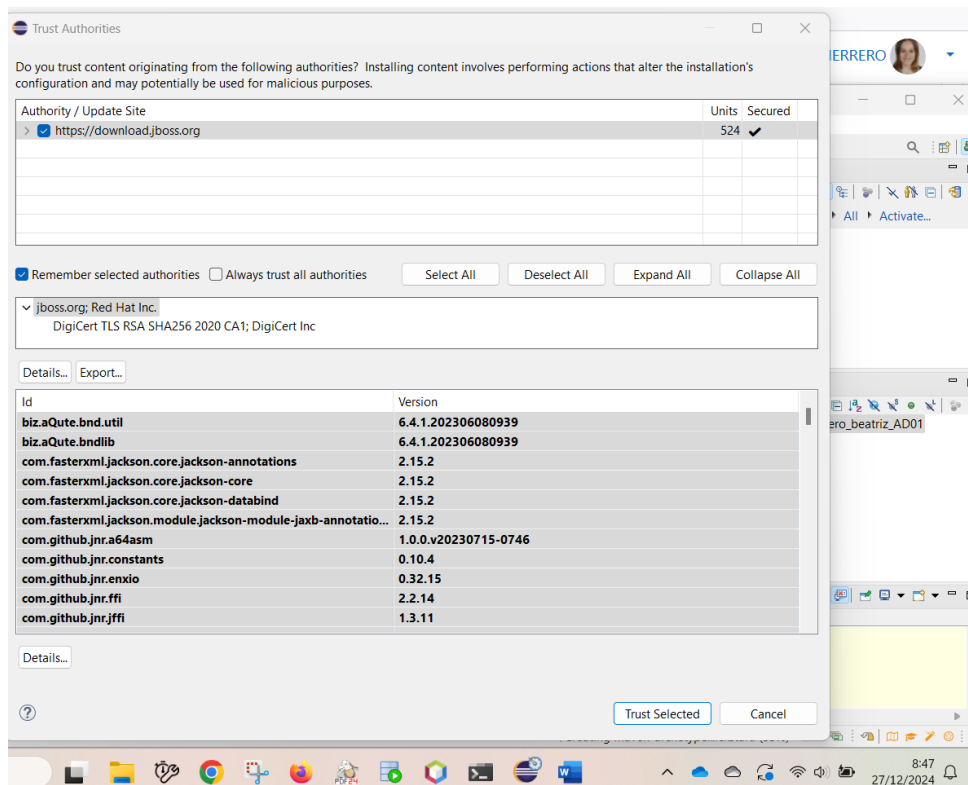
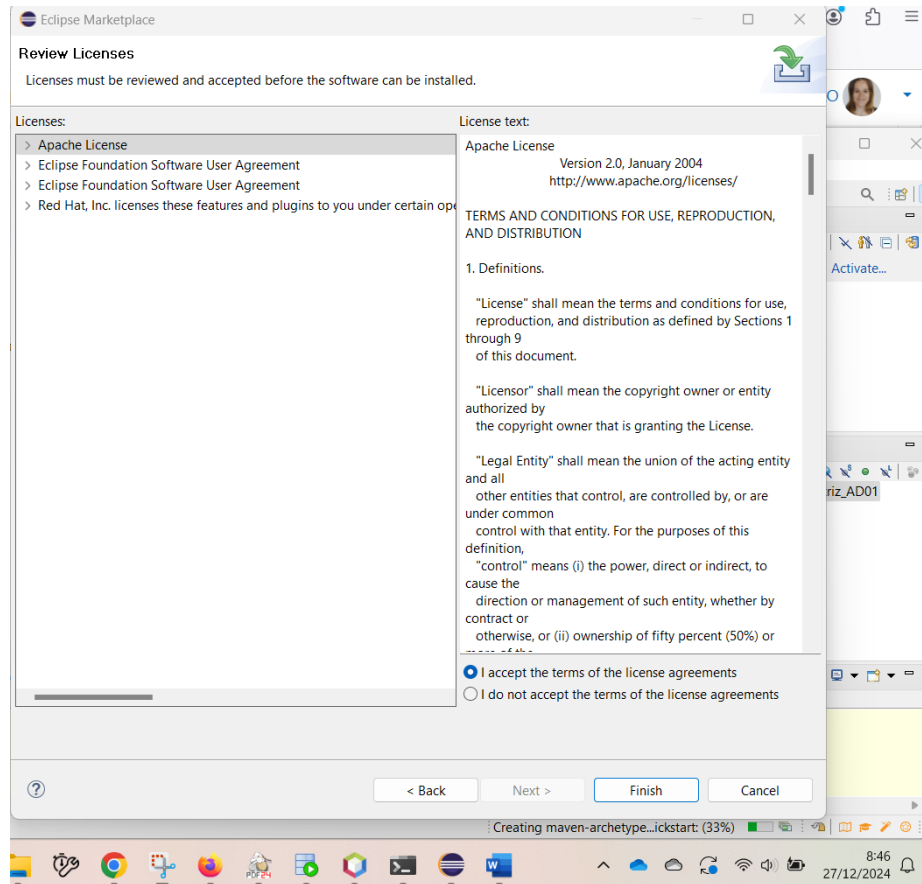
Instalo la herramienta JBoss Tool desde el Marketplace de Eclipse: Help → Eclipse Marketplace



Lo instalo sin los paquetes obsoletos

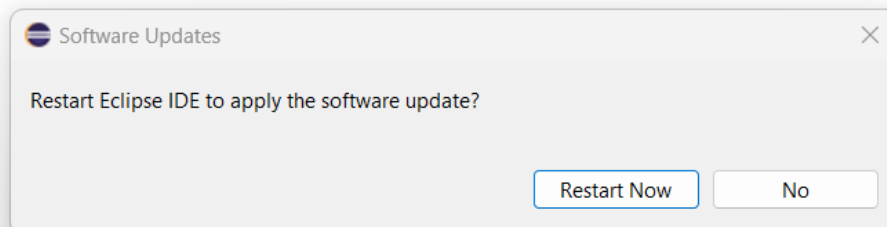


Y continúo con la instalación.

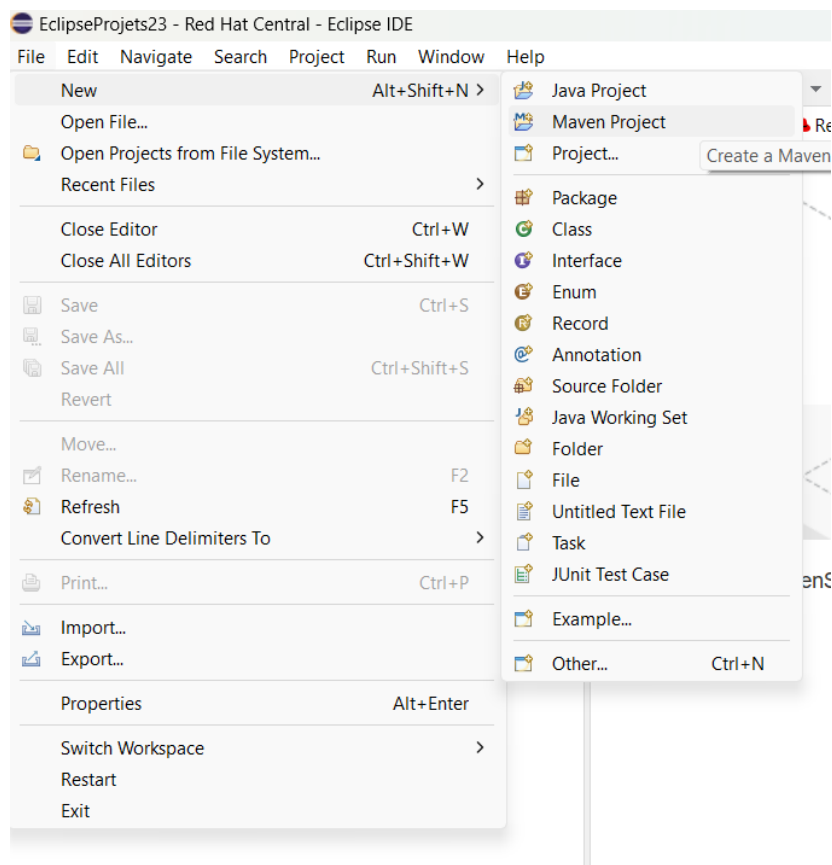


Reinicio Eclipse para que se guarden los cambios.

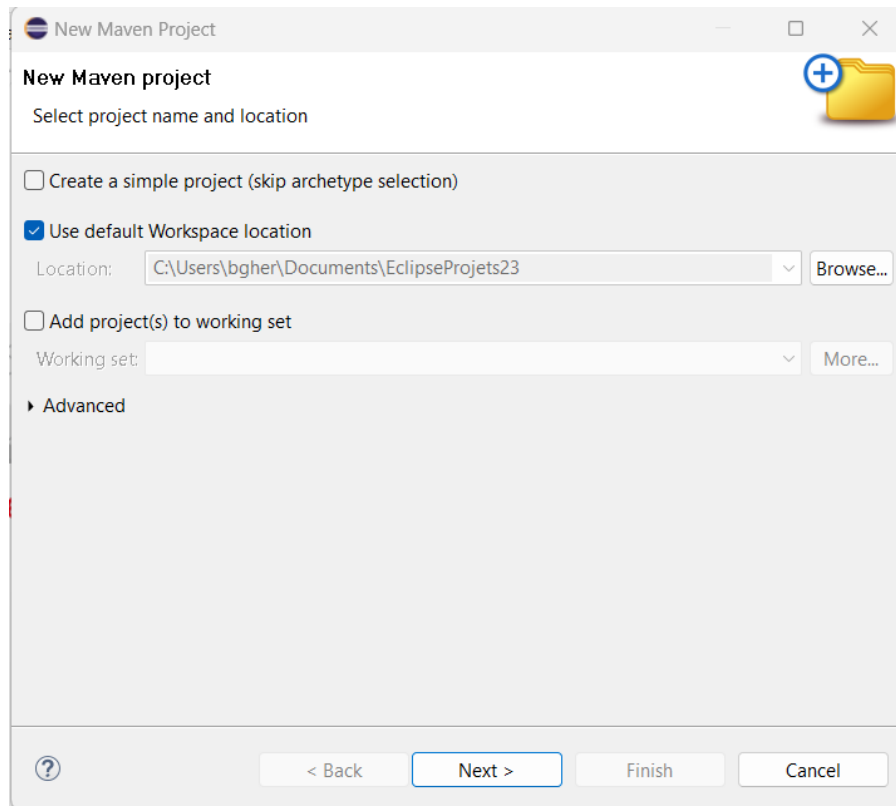
```
6  */  
7  module garcia_herrero_beatriz_AD01 {  
8  }
```



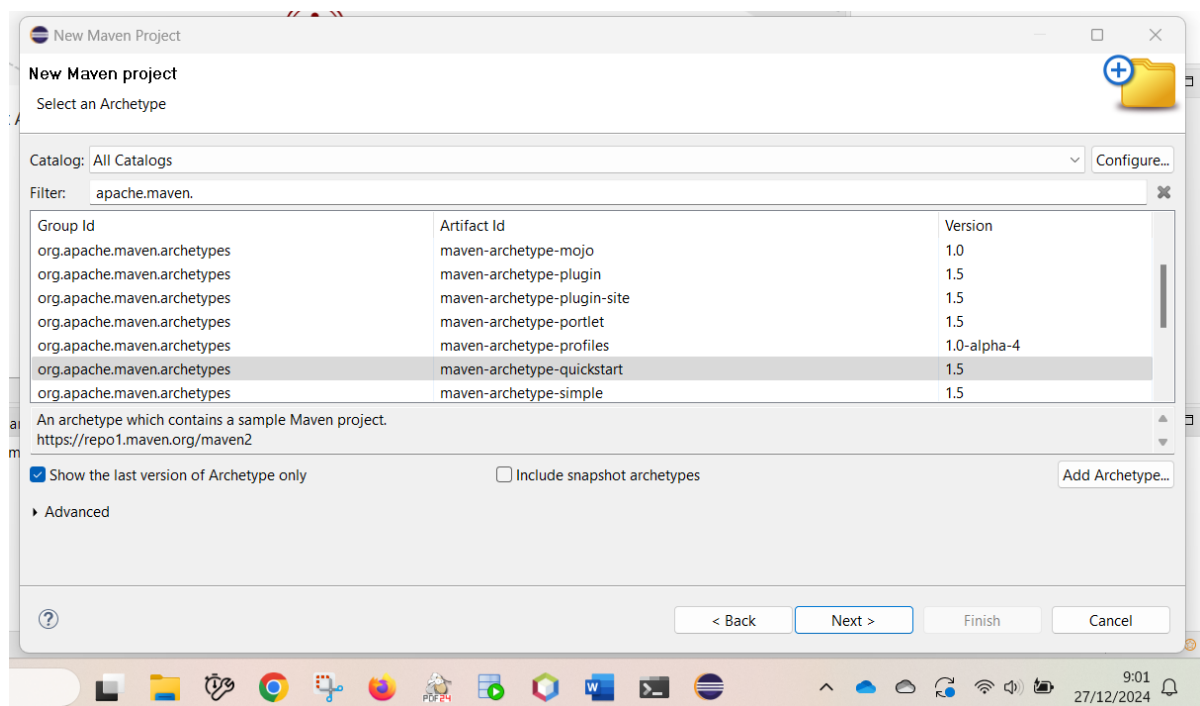
Ahora creo un nuevo proyecto Java Maven en Eclipse: File → New → Other... → Maven Project:

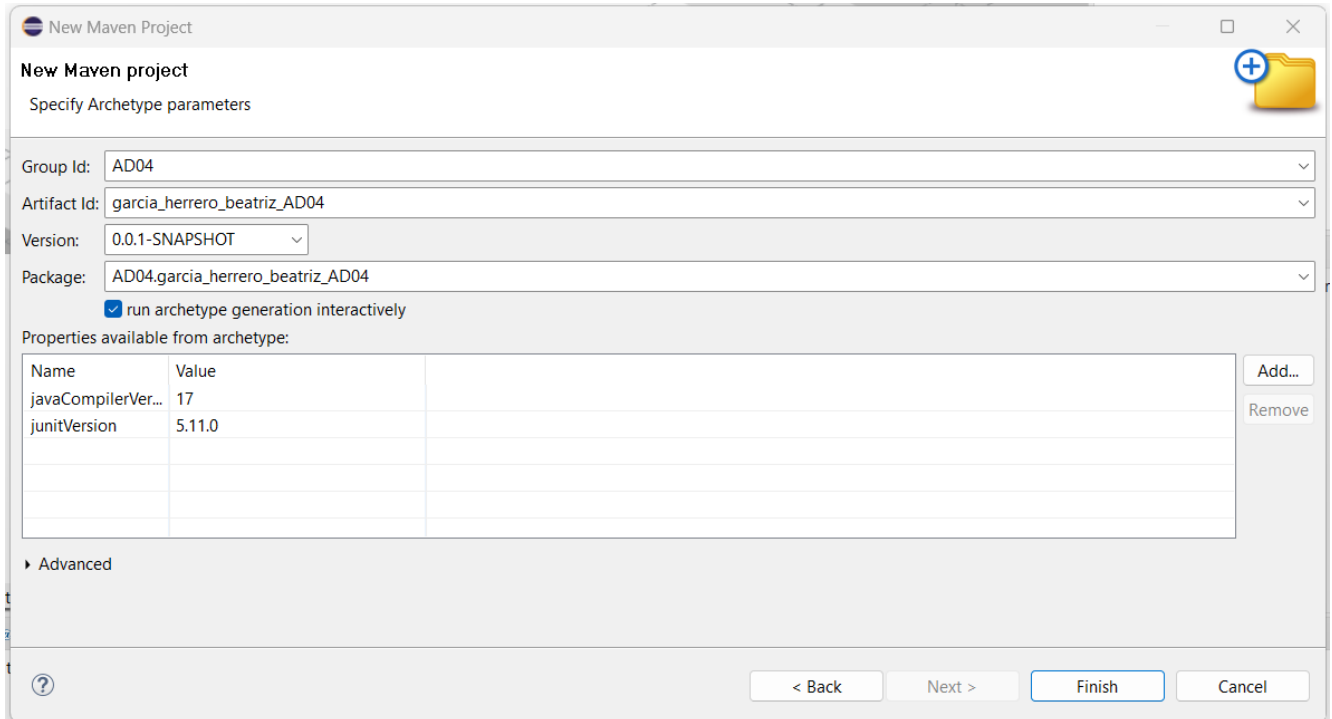


Escojo el espacio de trabajo por defecto.



Creo el proyecto con el artefacto de fácil inicio "quickstart".





New Maven Project

Specify Archetype parameters

Group Id: AD04

Artifact Id: garcia\_herrero\_beatriz\_AD04

Version: 0.0.1-SNAPSHOT

Package: AD04.garcia\_herrero\_beatriz\_AD04

☒ run archetype generation interactively

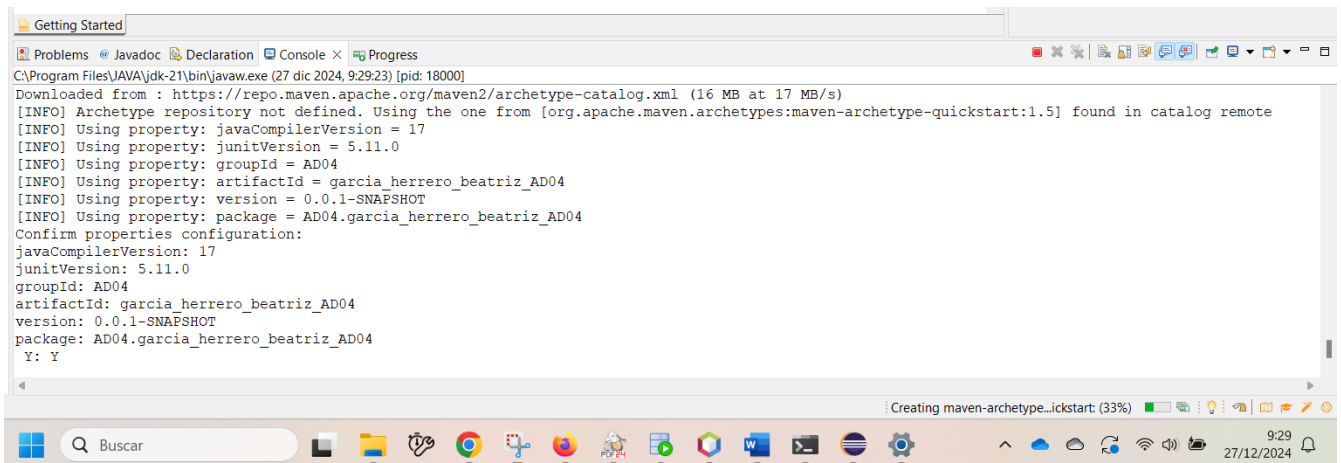
Properties available from archetype:

Name	Value
javaCompilerVer...	17
junitVersion	5.11.0

Advanced

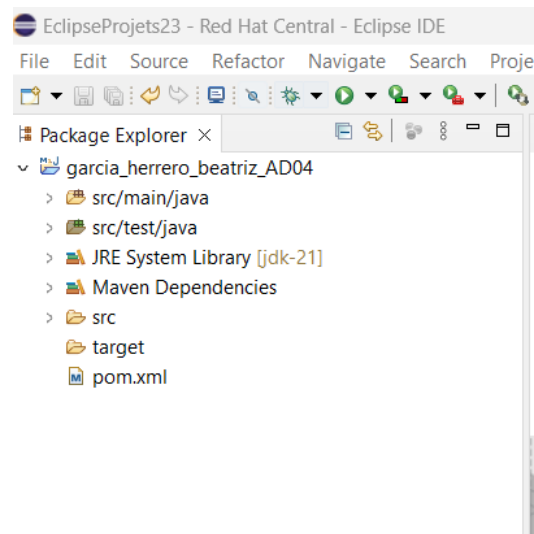
< Back Next > Finish Cancel

Escribo Y, para continuar y crear el nuevo proyecto.



```
Getting Started
Problems Javadoc Declaration Console × Progress
C:\Program Files\JAVA\jdk-21\bin\javaw.exe (27 dic 2024, 9:29:23) [pid: 18000]
Downloaded from : https://repo.maven.apache.org/maven2/archetype-catalog.xml (16 MB at 17 MB/s)
[INFO] Archetype repository not defined. Using the one from [org.apache.maven.archetypes:maven-archetype-quickstart:1.5] found in catalog remote
[INFO] Using property: javaCompilerVersion = 17
[INFO] Using property: junitVersion = 5.11.0
[INFO] Using property: groupId = AD04
[INFO] Using property: artifactId = garcia_herrero_beatriz_AD04
[INFO] Using property: version = 0.0.1-SNAPSHOT
[INFO] Using property: package = AD04.garcia_herrero_beatriz_AD04
Confirm properties configuration:
javaCompilerVersion: 17
junitVersion: 5.11.0
groupId: AD04
artifactId: garcia_herrero_beatriz_AD04
version: 0.0.1-SNAPSHOT
package: AD04.garcia_herrero_beatriz_AD04
Y: Y
Creating maven-archetype...ickstart: (33%)
```

El nuevo proyecto me ha generado la siguiente estructura de carpetas.



En el siguiente enlace tenemos la información de la dependencia de Hibertane.

<https://mvnrepository.com/artifact/org.hibernate.orm/hibernate-core/6.6.4.Final>

The screenshot shows the Maven Repository page for the artifact `org.hibernate.orm:hibernate-core:6.6.4.Final`. The page includes a sidebar with popular categories, a main content area with metadata, and a code block for the dependency declaration.

**Popular Categories**

- Testing Frameworks & Tools
- Android Packages
- Logging Frameworks
- JVM Languages
- Java Specifications
- JSON Libraries
- Core Utilities
- Mocking
- Annotation Libraries
- Web Assets
- HTTP Clients
- Language Runtime
- Logging Bridges
- Dependency Injection
- XML Processing
- Web Frameworks
- Android Platform
- I/O Utilities

**Hibernate ORM Hibernate Core » 6.6.4.Final**

Hibernate's core ORM functionality

License	LGPL 2.1
Categories	Object/Relational Mapping
Tags	persistence mapping orm hibernate relational
Organization	Hibernate.org
HomePage	<a href="https://hibernate.org/orm">https://hibernate.org/orm</a>
Date	Dec 18, 2024
Files	<a href="#">pom (5 KB)</a> <a href="#">jar (11.5 MB)</a> <a href="#">View All</a>
Repositories	Central
Ranking	#1180 in MvnRepository (See Top Artifacts) #6 in Object/Relational Mapping
Used By	464 artifacts

**Note:** There is a new version for this artifact

**New Version** [7.0.0.Beta3](#)

**Builds**

- Maven
- Gradle
- Gradle (Short)
- Gradle (Kotlin)
- SBT
- Ivy
- Grape
- Leiningen
- Buildr

```
<!-- https://mvnrepository.com/artifact/org.hibernate.orm/hibernate-core -->
<dependency>
  <groupId>org.hibernate.orm</groupId>
  <artifactId>hibernate-core</artifactId>
  <version>6.6.4.Final</version>
</dependency>
```

Y la dependencia del conector de Oracle.

<https://mvnrepository.com/artifact/com.oracle.database.jdbc/ojdbc11/23.6.0.24.10>

mvnrepository.com/artifact/com.oracle.database.jdbc/ojdbc11/23.6.0.24.10

Indexed Artifacts (48.5M)

Popular Categories

- Testing Frameworks & Tools
- Android Packages
- Logging Frameworks
- JVM Languages
- Java Specifications
- JSON Libraries
- Core Utilities
- Mocking
- Annotation Libraries
- Web Assets
- HTTP Clients
- Language Runtime
- Logging Bridges

Home » com.oracle.database.jdbc » ojdbc11 » 23.6.0.24.10

**Ojdbc11 » 23.6.0.24.10**

Oracle JDBC Driver compatible with JDK11, JDK17, JDK19, and JDK21

Categories	JDBC Drivers
Tags	database sql oracle jdbc data driver rdbms
HomePage	<a href="https://www.oracle.com/database/technologies/maven-central-g...">https://www.oracle.com/database/technologies/maven-central-g...</a>
Date	Nov 06, 2024
Files	<a href="#">pom (1 KB)</a> <a href="#">jar (6.9 MB)</a> <a href="#">View All</a>
Repositories	Central Mulesoft
Ranking	#2821 in MvnRepository (See Top Artifacts) #15 in JDBC Drivers
Used By	178 artifacts

Maven Gradle Gradle (Short) Gradle (Kotlin) SBT Ivy Grape Leiningen Buildr

```
<!-- https://mvnrepository.com/artifact/com.oracle.database.jdbc/ojdbc11 -->
<dependency>
  <groupId>com.oracle.database.jdbc</groupId>
  <artifactId>ojdbc11</artifactId>
  <version>23.6.0.24.10</version>
</dependency>
```

☒ Include comment with link to declaration

Dentro del proyecto añado las dependencias de Hibernate y de la base de datos de Oracle en el fichero “pom.xml”

EclipseProjets23 - garcia\_herrero\_beatriz\_AD04/pom.xml - Eclipse IDE

File Edit Source Navigate Search Project Run Design Window Help

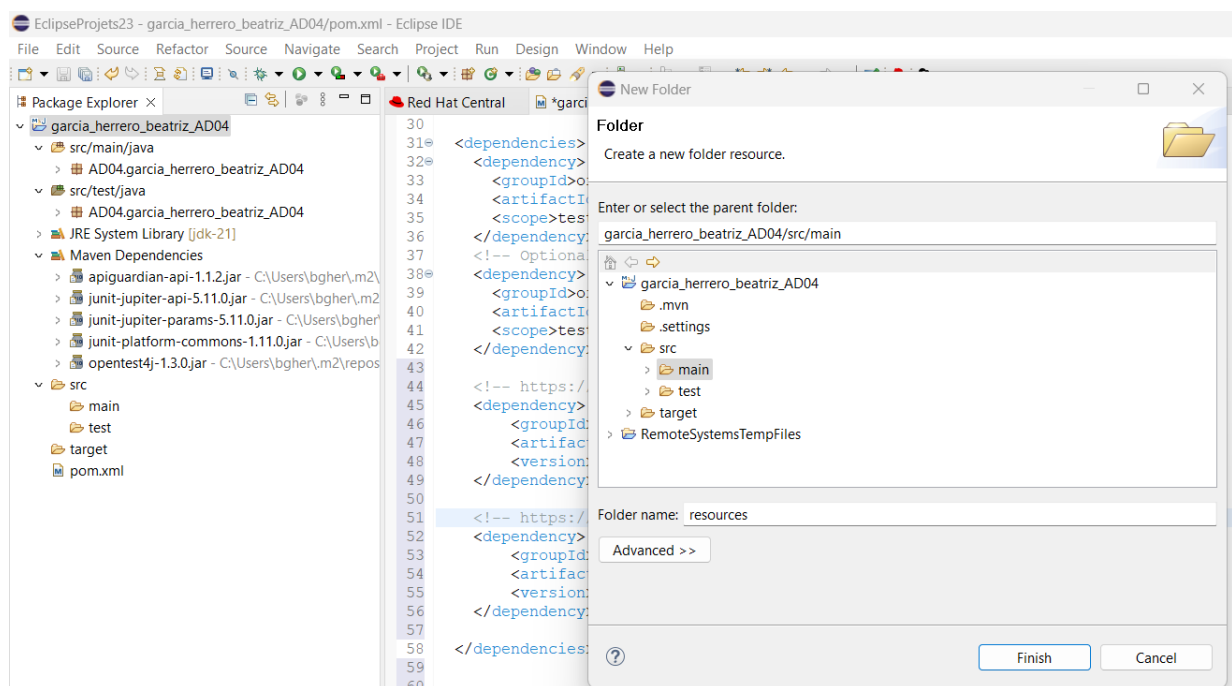
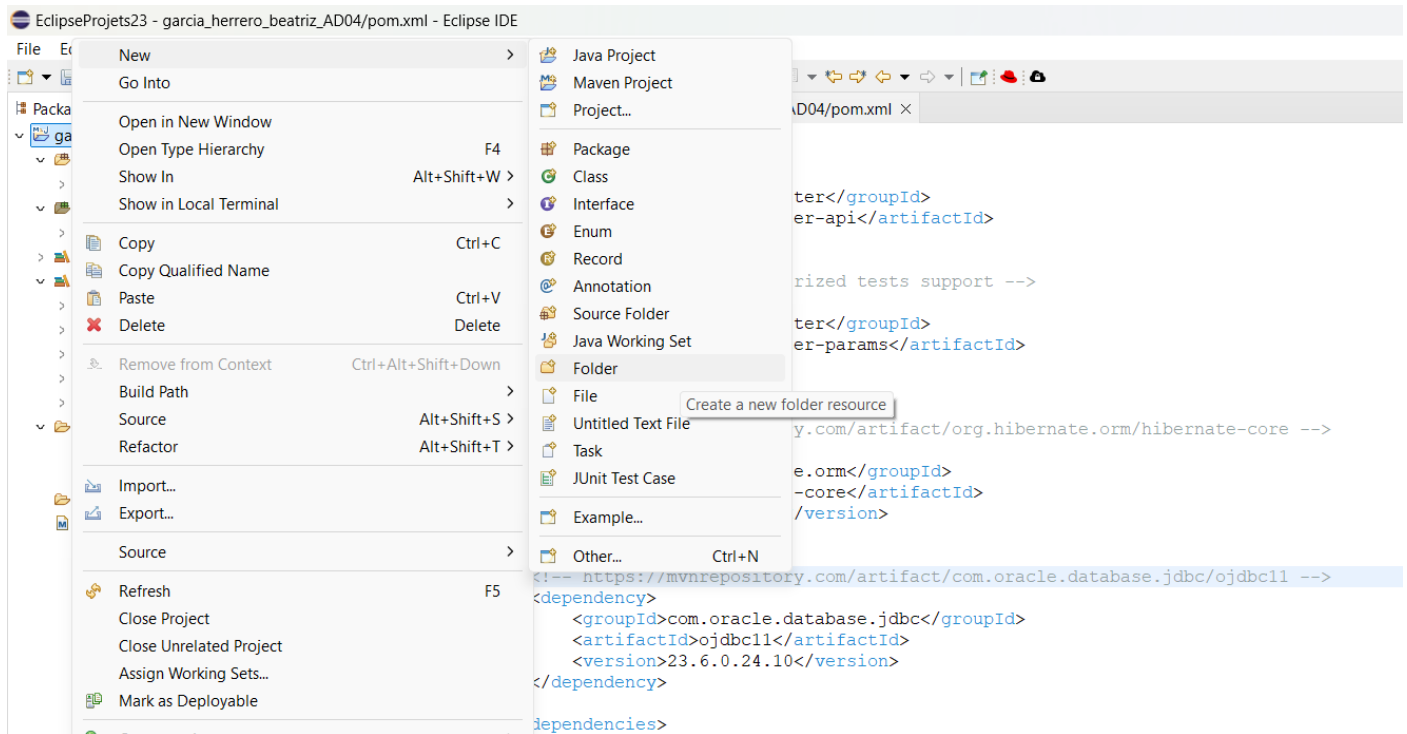
Package Explorer

- garcia\_herrero\_beatriz\_AD04
  - src/main/java
    - AD04.garcia\_herrero\_beatriz\_AD04
  - src/test/java
    - AD04.garcia\_herrero\_beatriz\_AD04
  - JRE System Library [jdk-21]
  - Maven Dependencies
    - apiguardian-api-1.1.2.jar - C:\Users\bgher\m2
    - junit-jupiter-api-5.11.0.jar - C:\Users\bgher\m2
    - junit-jupiter-params-5.11.0.jar - C:\Users\bgher\m2
    - junit-platform-commons-1.11.0.jar - C:\Users\bgher\m2
    - opentest4j-1.3.0.jar - C:\Users\bgher\m2\repos
  - src
    - main
    - test
    - target
    - pom.xml**

```
30
31 <dependencies>
32 <dependency>
33   <groupId>org.junit.jupiter</groupId>
34   <artifactId>junit-jupiter-api</artifactId>
35   <scope>test</scope>
36 </dependency>
37 <!-- Optionally: parameterized tests support -->
38 <dependency>
39   <groupId>org.junit.jupiter</groupId>
40   <artifactId>junit-jupiter-params</artifactId>
41   <scope>test</scope>
42 </dependency>
43
44 <!-- https://mvnrepository.com/artifact/org.hibernate.orm/hibernate-core -->
45 <dependency>
46   <groupId>org.hibernate.orm</groupId>
47   <artifactId>hibernate-core</artifactId>
48   <version>6.6.4.Final</version>
49 </dependency>
50
51 <!-- https://mvnrepository.com/artifact/com.oracle.database.jdbc/ojdbc11 -->
52 <dependency>
53   <groupId>com.oracle.database.jdbc</groupId>
54   <artifactId>ojdbc11</artifactId>
55   <version>23.6.0.24.10</version>
56 </dependency>
57
58 </dependencies>
59
60
```

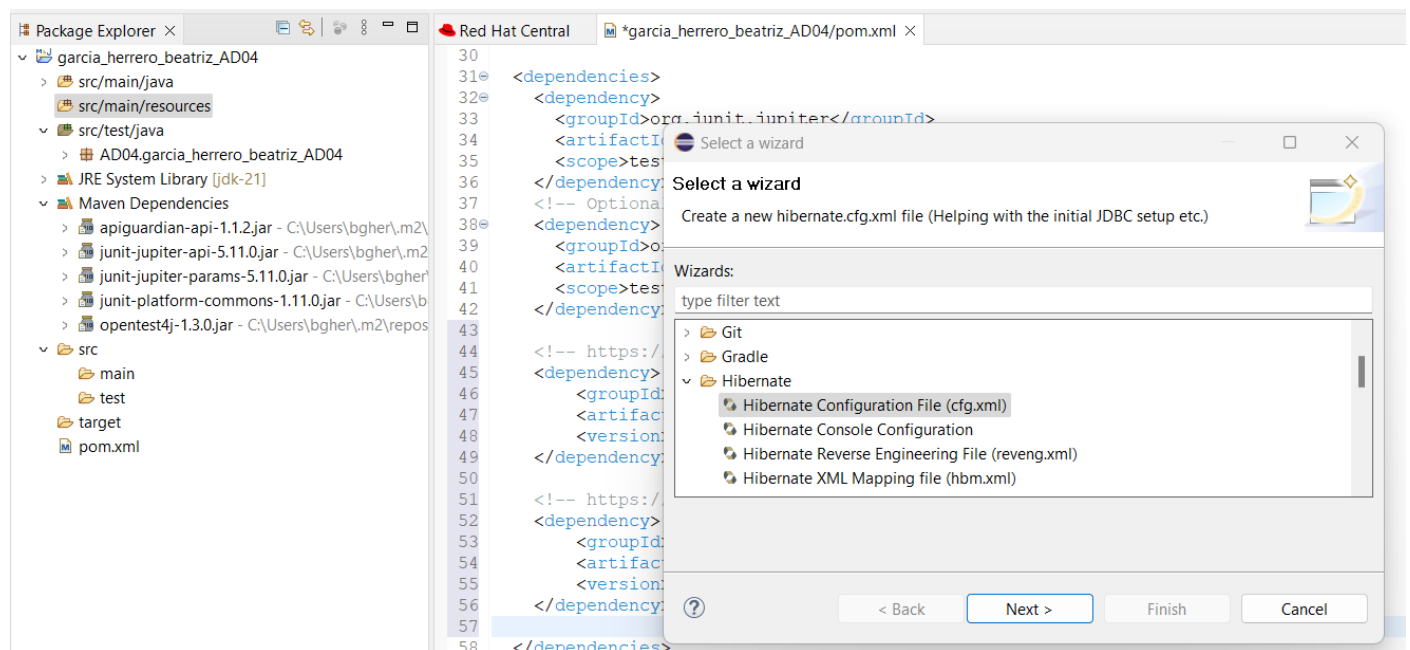
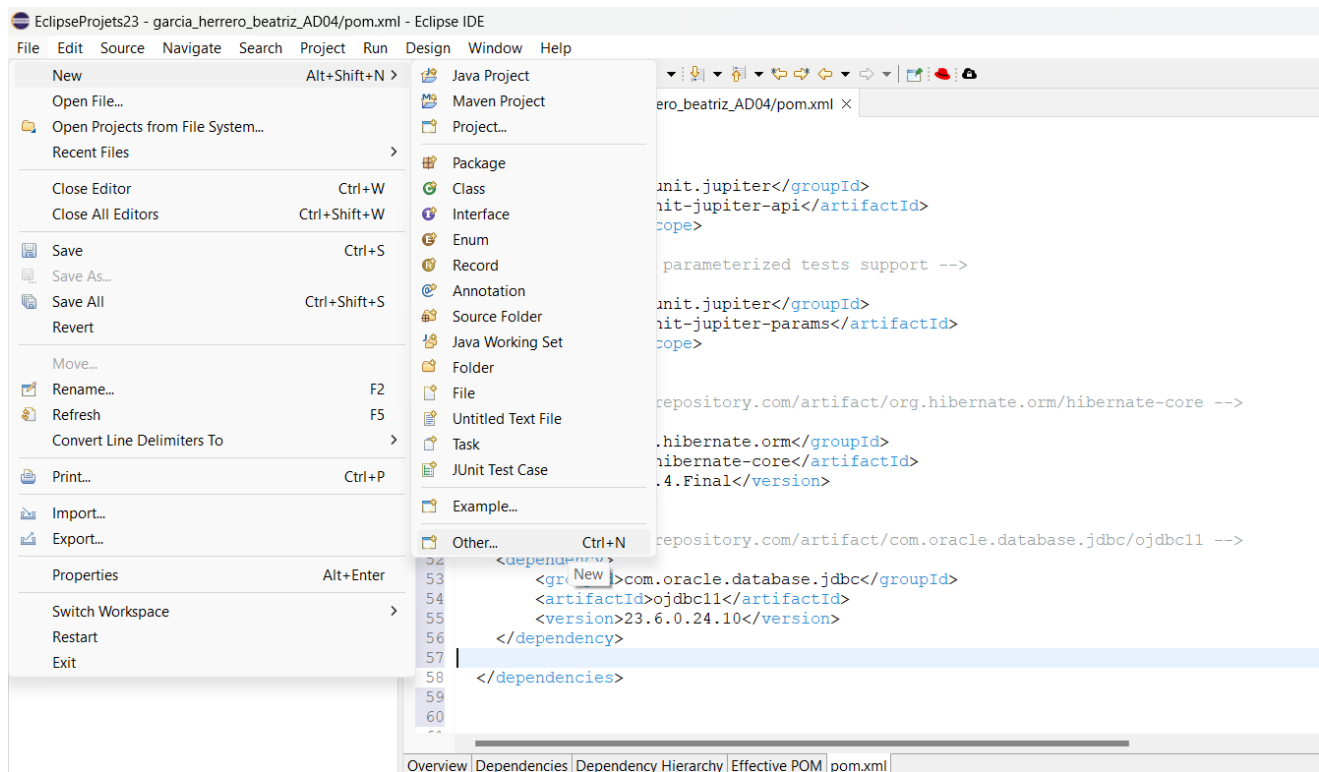


Creo una nueva carpeta “New” → “Folder”, para guardar los recursos de Hibernate:

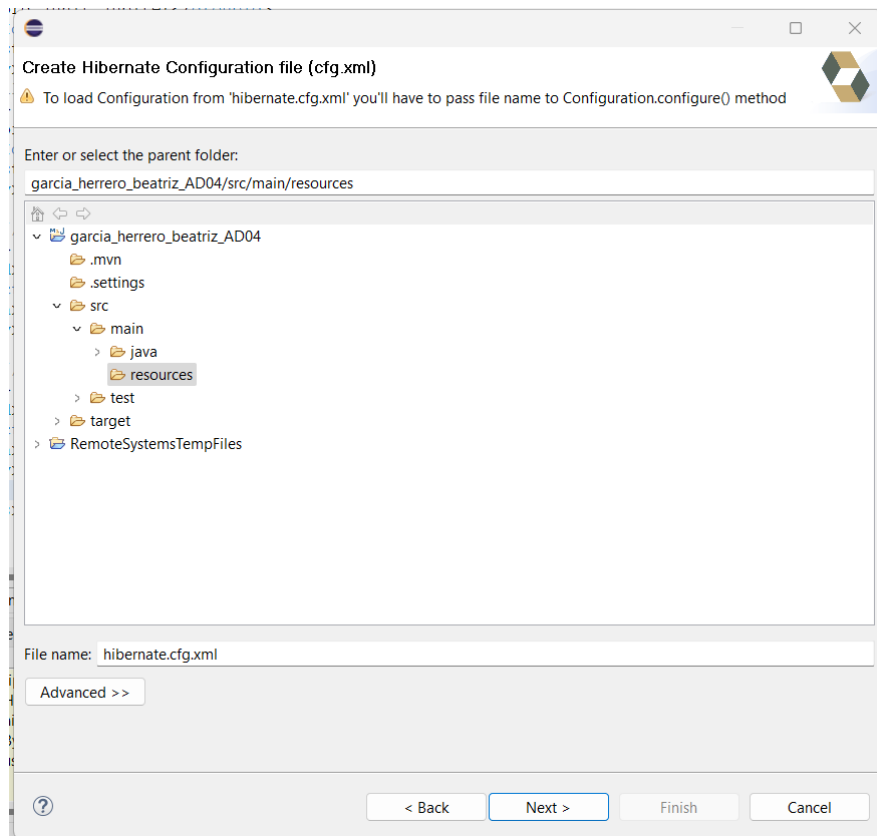


Creo un fichero de configuración de Hibernate: “New” → “Other” → Hibernate

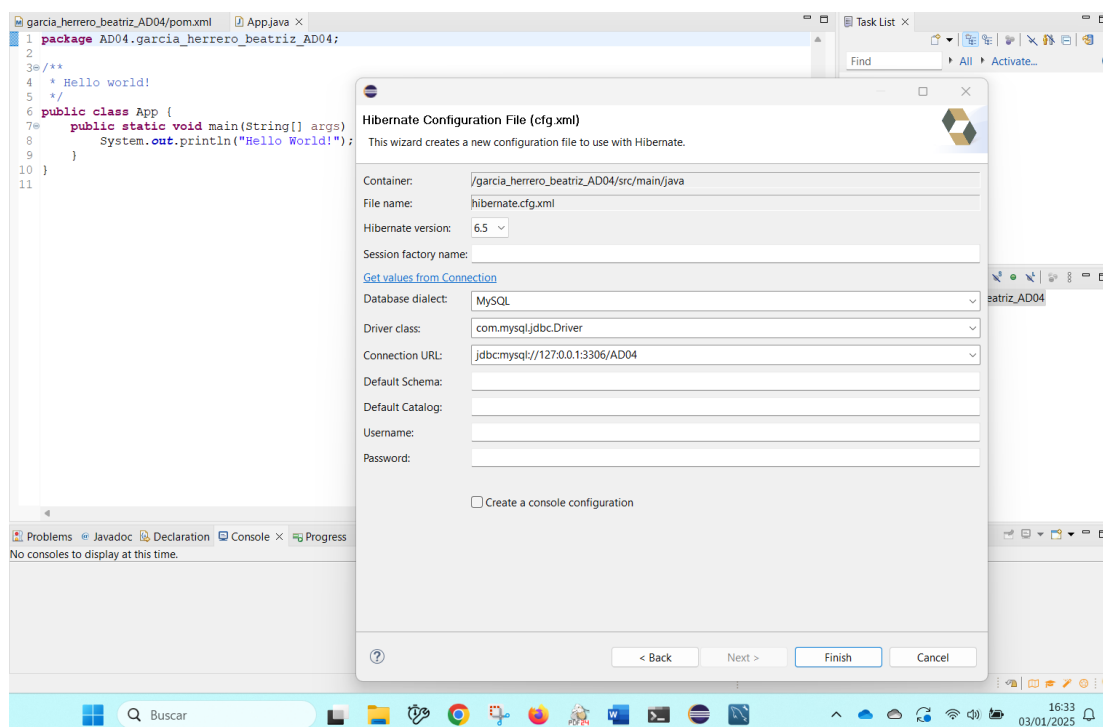
Añadimos al proyecto: Hibernate Configuration File (cfg.xml)



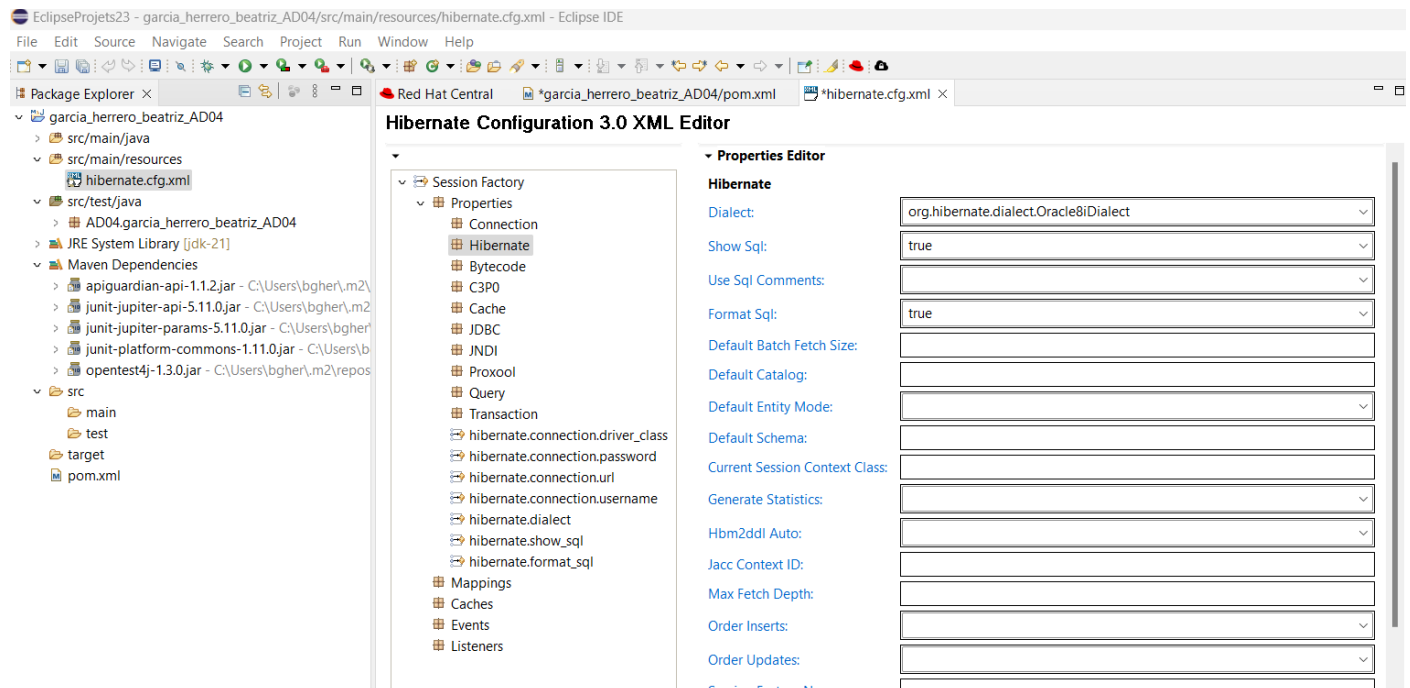
Indico que se almacene en el “src/main/resources” del proyecto que acabo de crear.



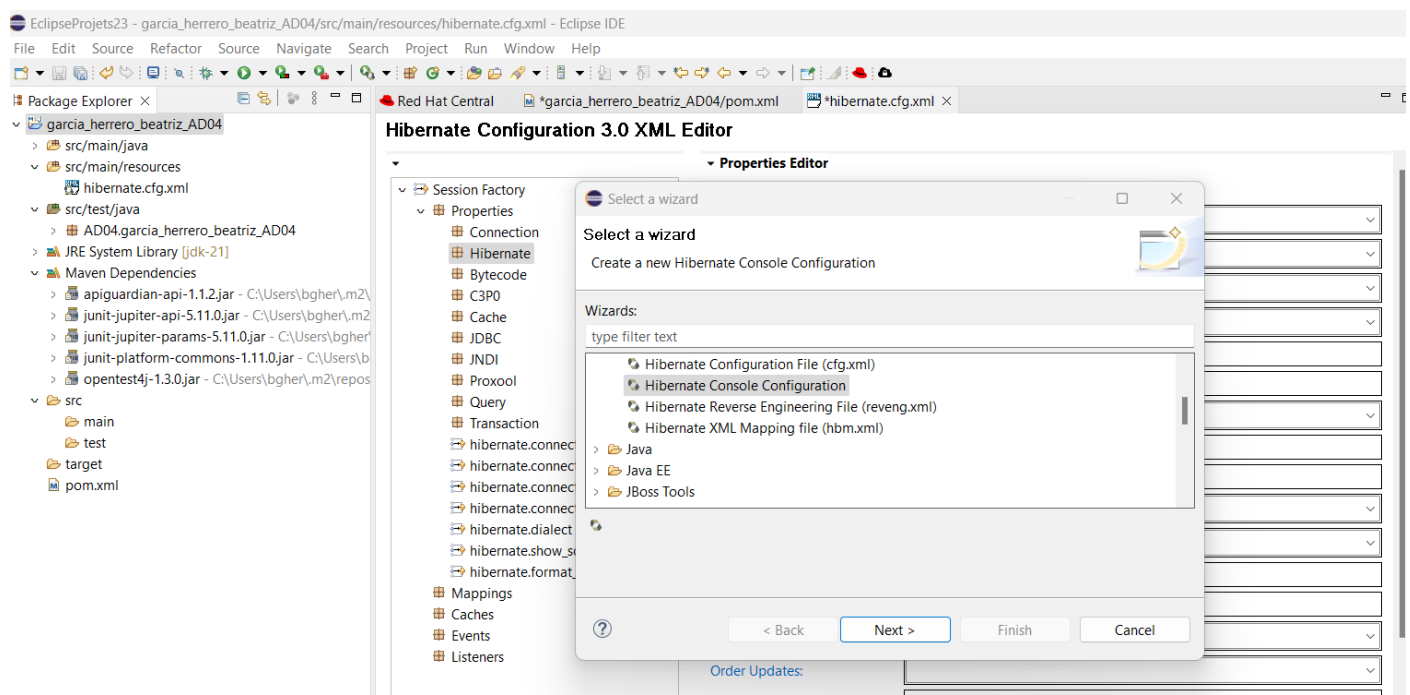
Realizo la configuración a la base de datos Oracle.

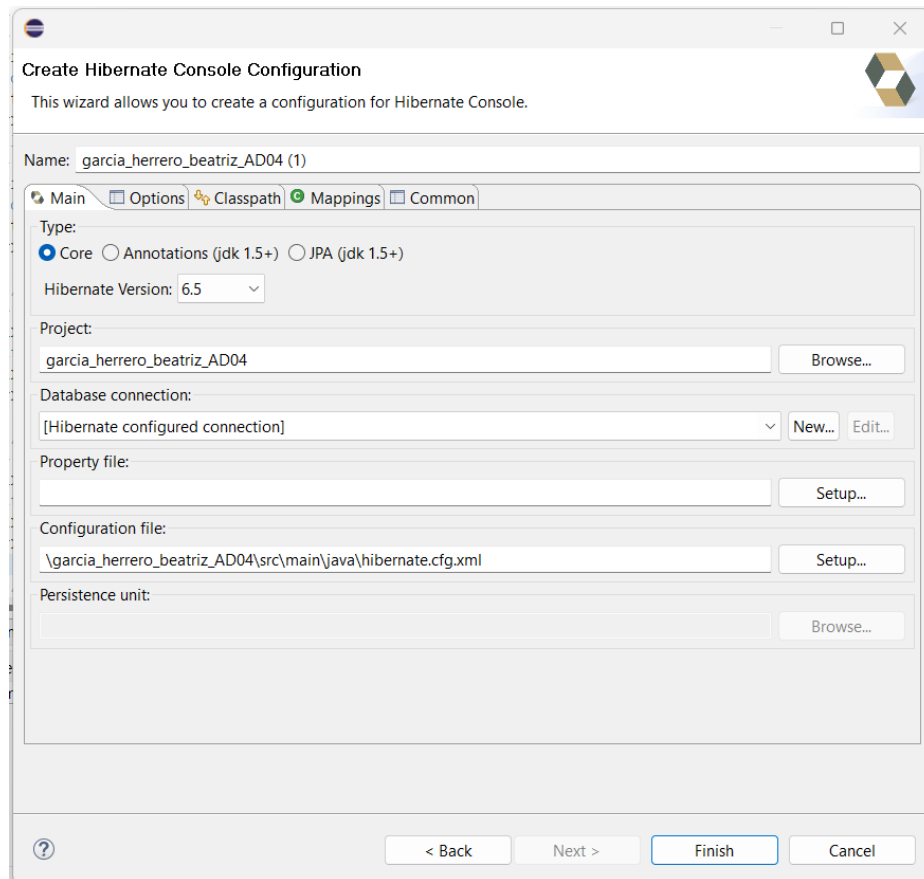


En el fichero de configuración, elijo que se muestre el SQL (Show Sql) y el formato SQL (Format Sql).

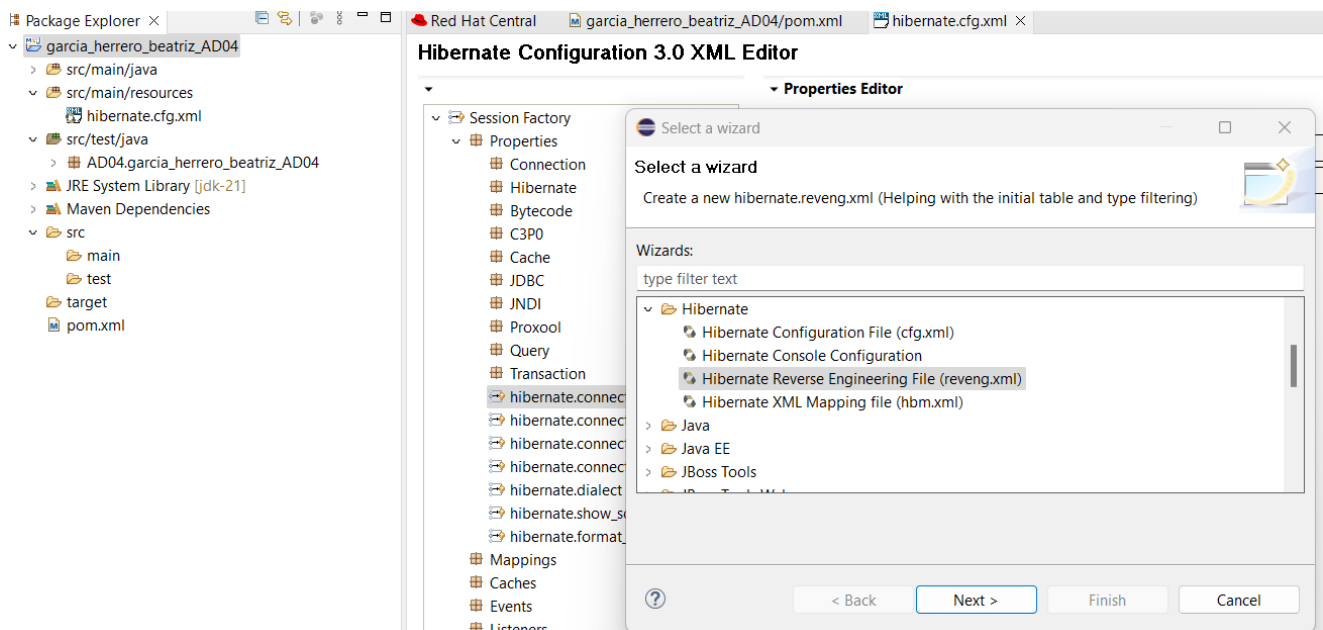


El siguiente paso es crear un archivo de configuración de consola. Nos vamos a New → Other → Hibernate → Hibernate Console Configuration.

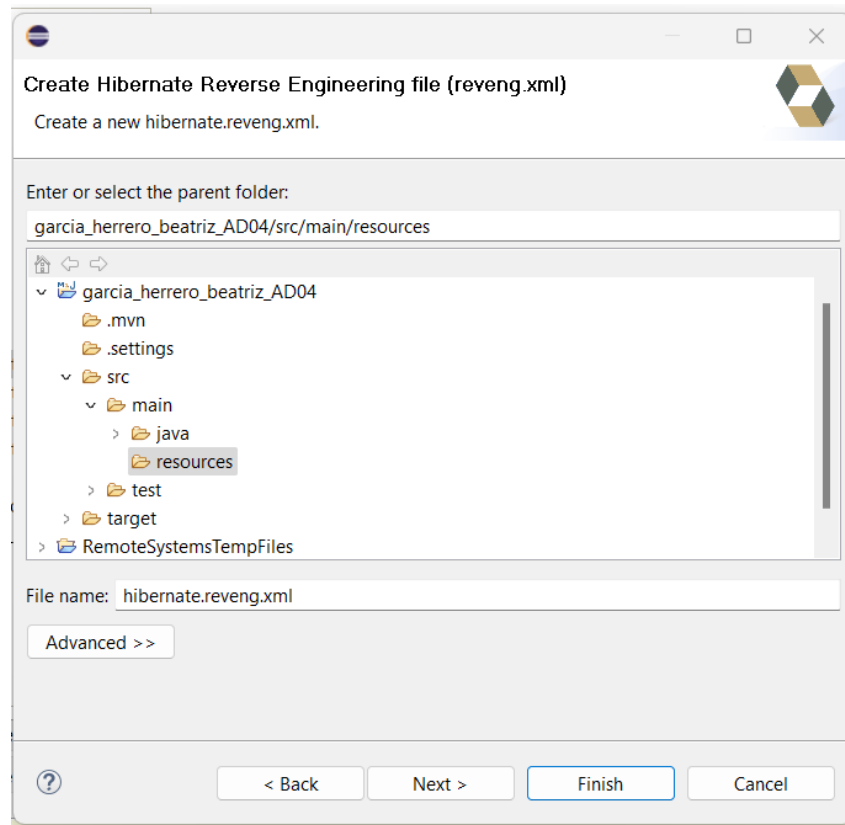




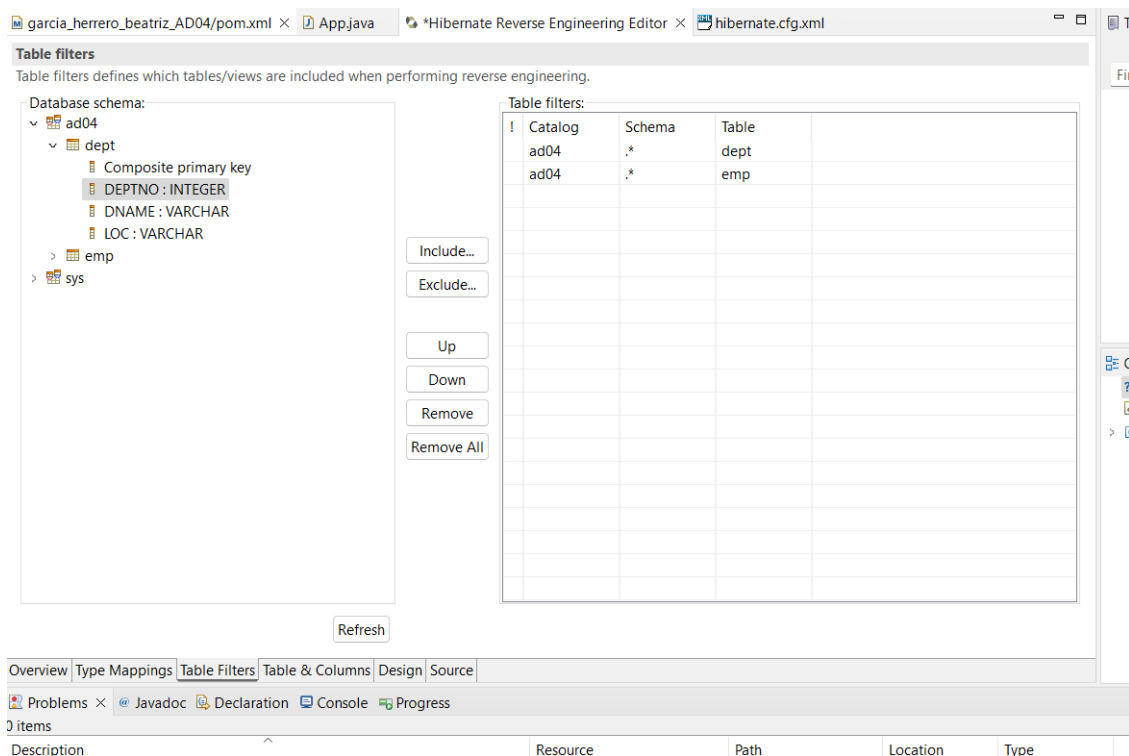
Lo siguiente será generar un archivo para la ingeniería inversa (la que transforme los datos de la base de datos relacional en objetos POO). Vamos a New → Other → Hibernate → Hibernate Reverse Engineering File



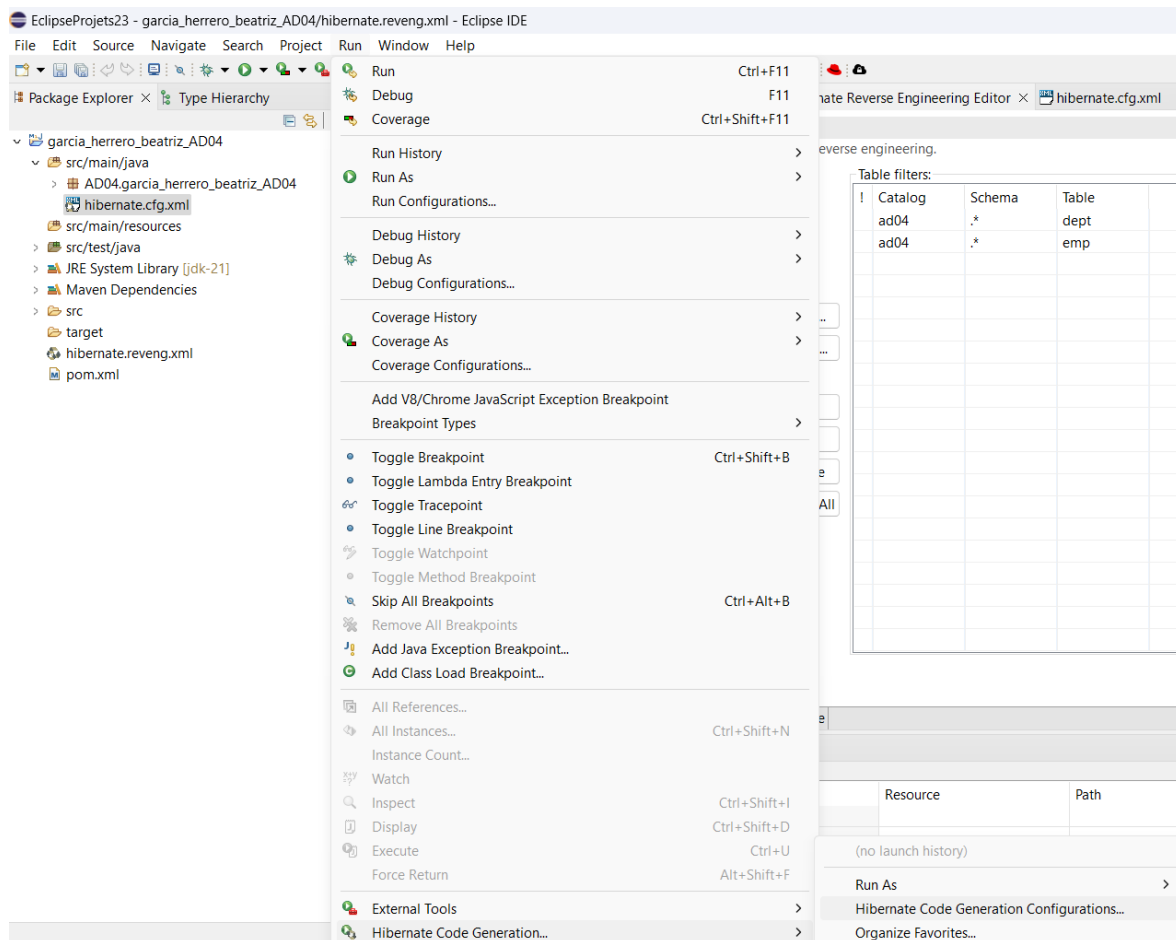
De nuevo, deberemos indicar la ruta donde se almacena (parent folder), marcando la carpeta “src/main/resources” de nuestro proyecto, y el nombre lo dejamos por defecto.



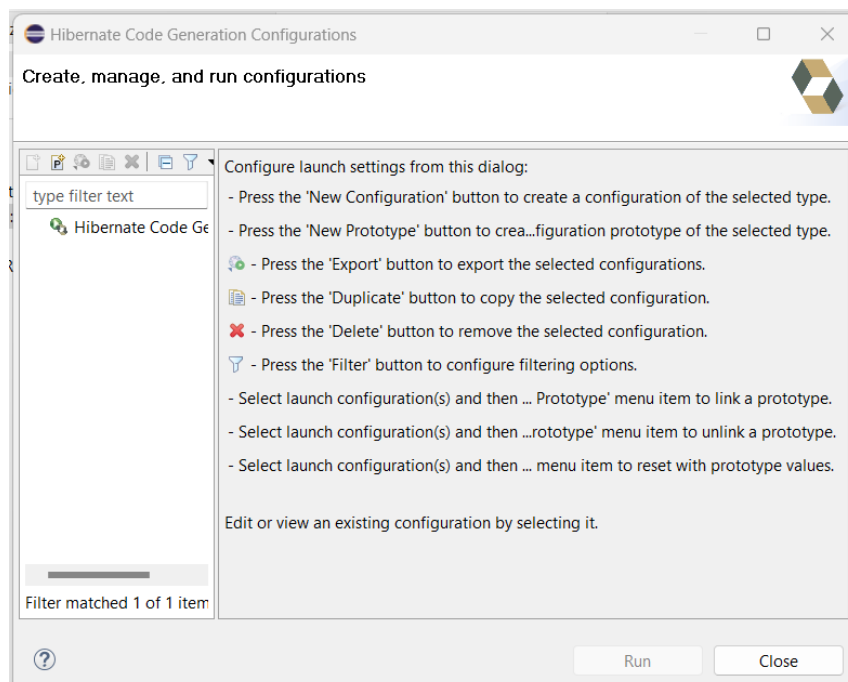
Ahora configuro las tablas que quiero traduzca a objetos.



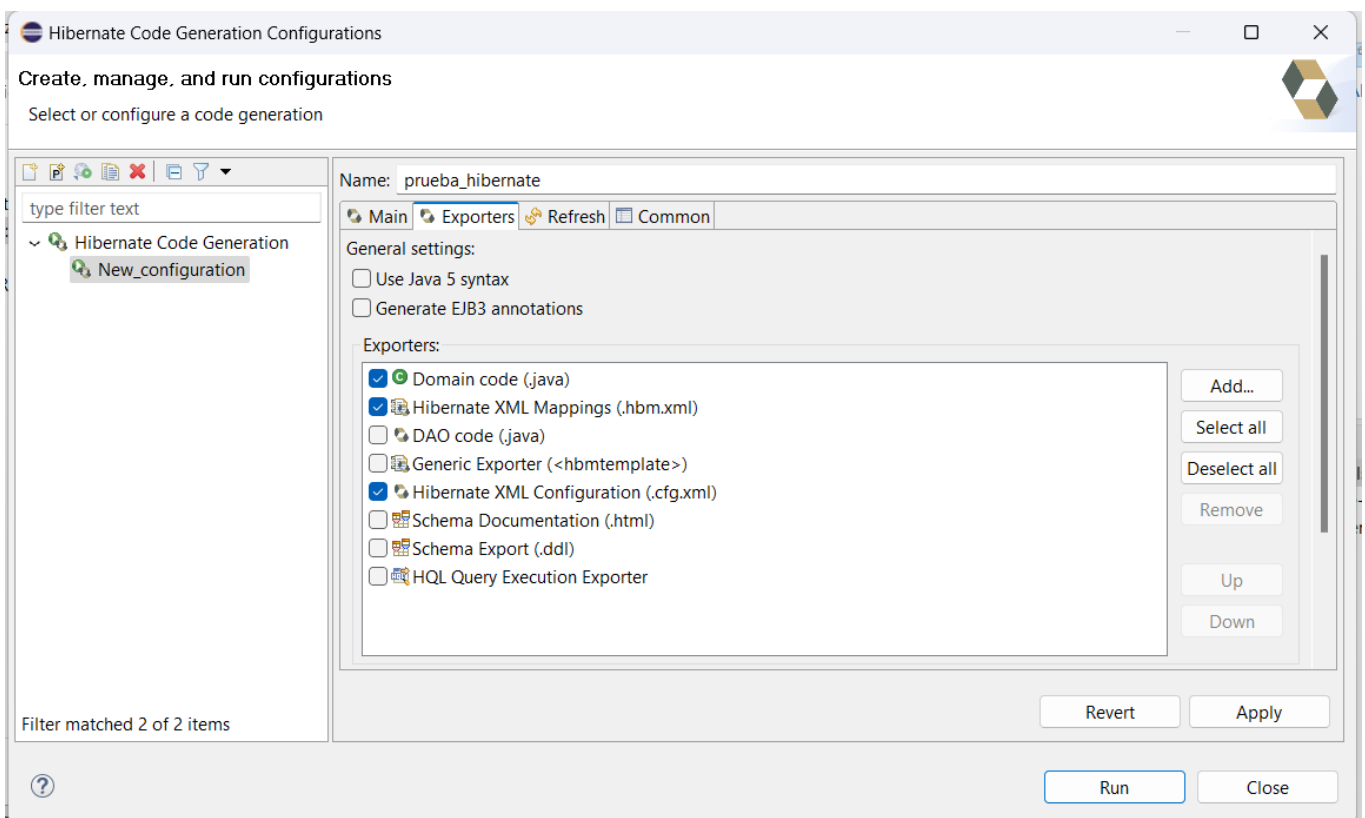
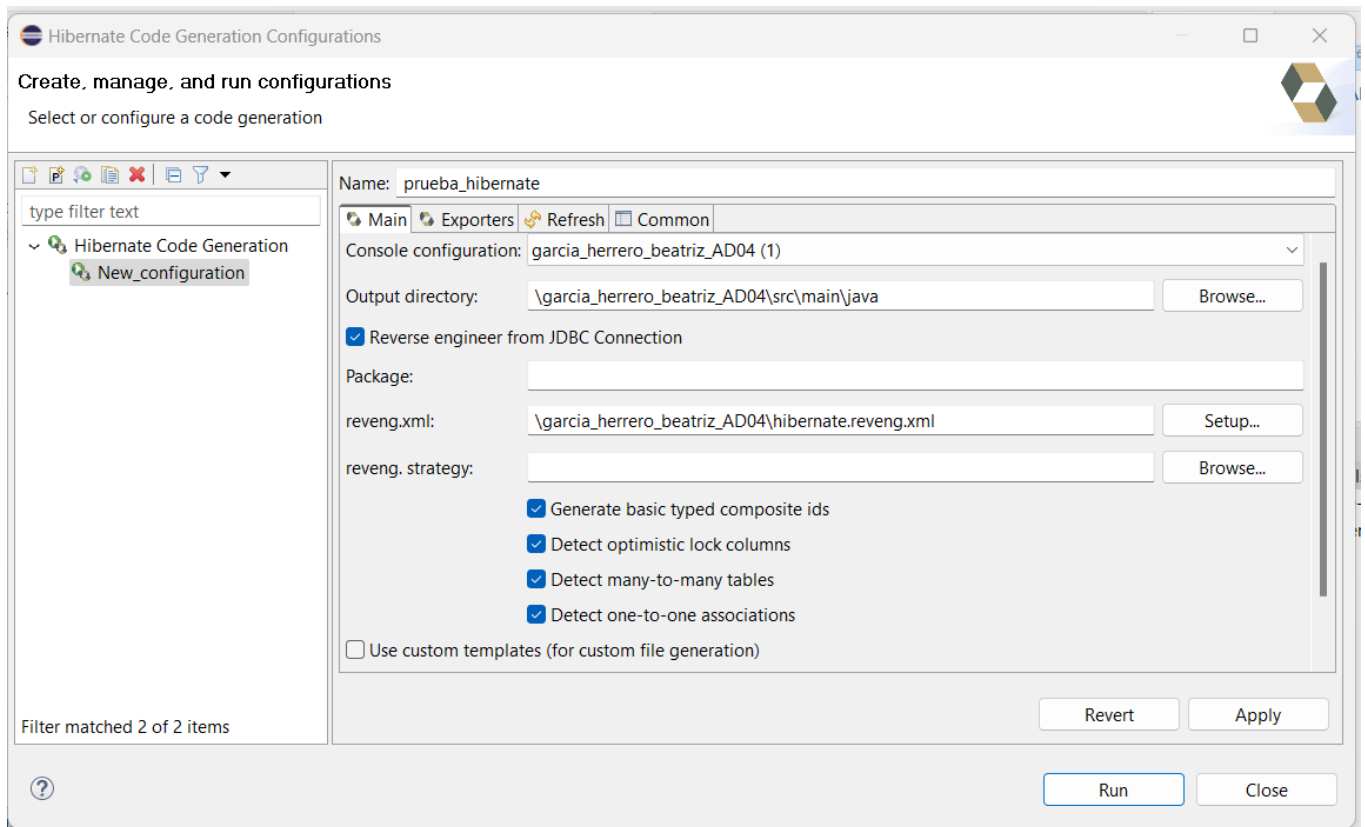
Debo indicar a Hibernate que genere el código en Java. Nos iremos a Run → Hibernate Code Generation → “Hibernate Code Generation Configurations...”



Y aparece la siguiente pantalla



## Creo una nueva configuración





Aplico los cambios y pulso Run.

```

1 // default package
2 package AD04.garcia_herrero_beatriz_AD04;
3
4
5 import java.math.BigDecimal;
6
7
8 /**
9  * Emp generated by hbm2java
10  */
11 public class Emp implements java.io.Serializable {
12
13     /**
14      *
15      */
16     private static final long serialVersionUID = 1L;
17     private int empno;
18     private int deptno;
19     private String ename;
20     private String job;
21     private Integer mgr;
22     private Date hiredate;
23     private double sal;
24     private BigDecimal comm;
25
26     public Emp() {
27     }
28
29     public Emp(int empno, int deptno) {
30         this.empno = empno;
31         this.deptno = deptno;
32     }
33
34     public Emp(int empno, int deptno, String ename, String job, Integer mgr, Date hiredate, double sal,
35         BigDecimal comm) {
36         this.empno = empno;
37         this.deptno = deptno;
38         this.ename = ename;
39         this.job = job;
40         this.mgr = mgr;
41         this.hiredate = hiredate;
42         this.sal = sal;
43         this.comm = comm;
44     }
45 }

```

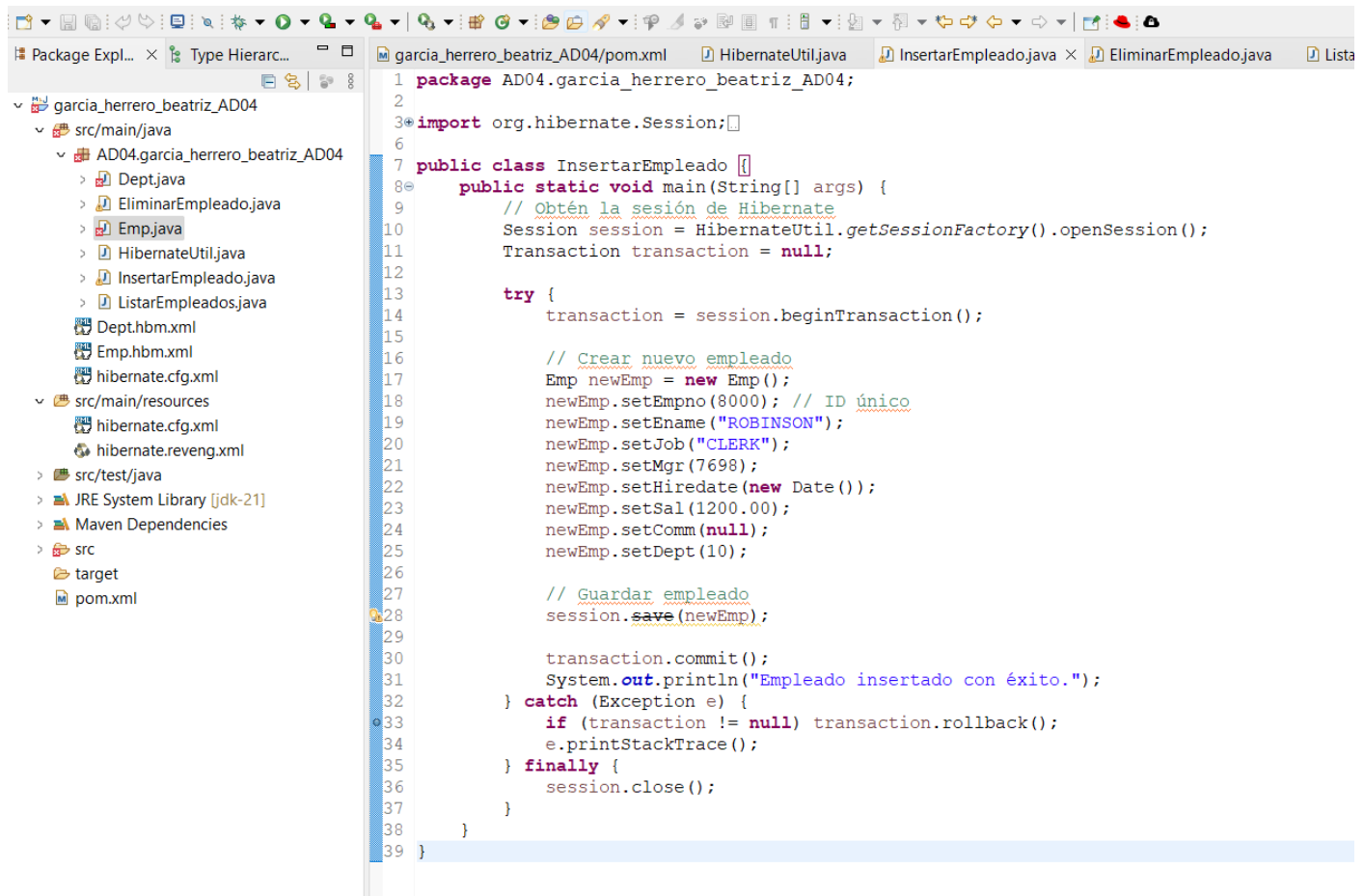
El código para eliminar un empleado.

```

1 package AD04.garcia_herrero_beatriz_AD04;
2
3 import org.hibernate.Session;
4
5
6 public class EliminarEmpleado {
7
8     public static void main(String[] args) {
9         // Obtén la sesión de Hibernate
10         Session session = HibernateUtil.getSessionFactory().openSession();
11         Transaction transaction = null;
12
13         try {
14             transaction = session.beginTransaction();
15
16             // Obtener el empleado a eliminar
17             Emp empToDelete = session.get(Emp.class, 8000); // ID del empleado a eliminar
18
19             if (empToDelete != null) {
20                 session.delete(empToDelete);
21                 System.out.println("Empleado eliminado con éxito.");
22             } else {
23                 System.out.println("Empleado no encontrado.");
24             }
25
26             transaction.commit();
27         } catch (Exception e) {
28             if (transaction != null) transaction.rollback();
29             e.printStackTrace();
30         } finally {
31             session.close();
32         }
33     }
34 }
35
36

```

El código para insertar un empleado.

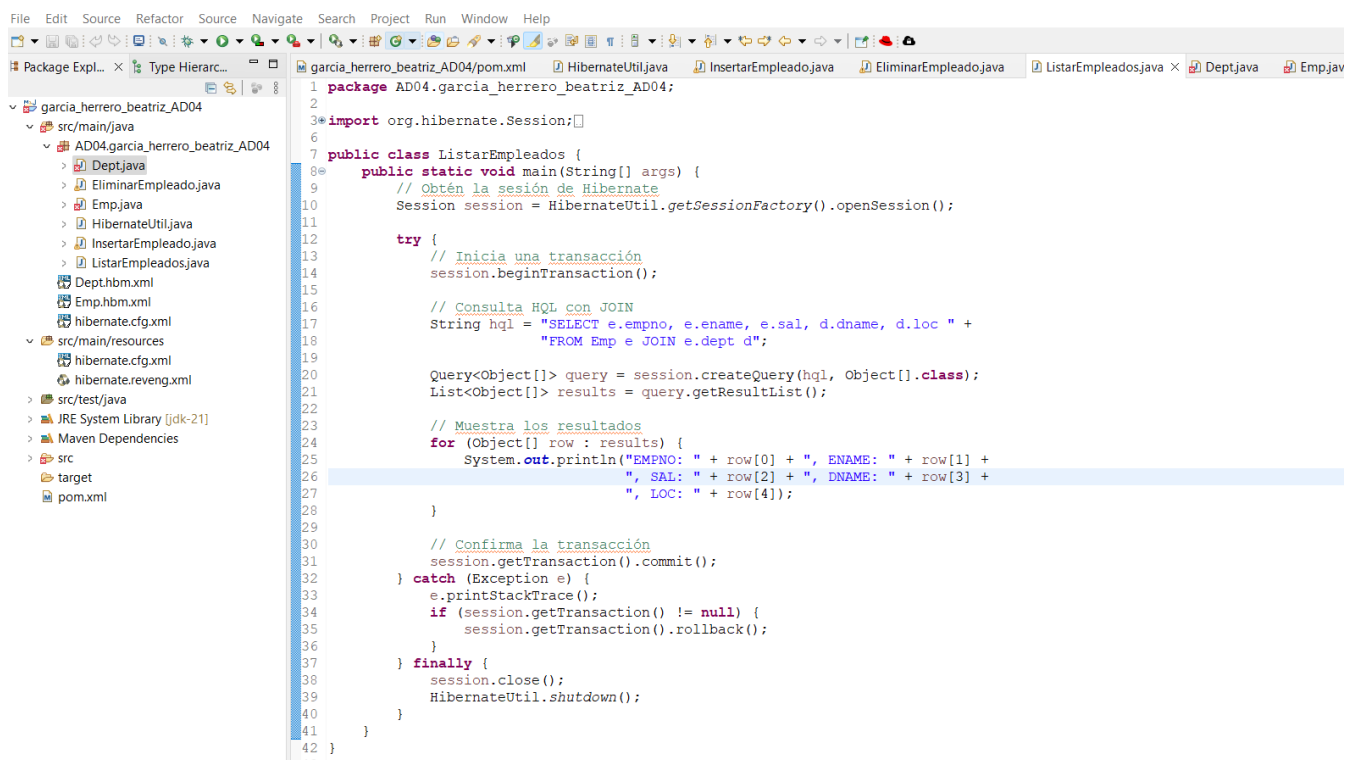


```

1 package AD04.garcia_herrero_beatriz_AD04;
2
3 import org.hibernate.Session;
4
5
6
7 public class InsertarEmpleado {
8     public static void main(String[] args) {
9         // Obtén la sesión de Hibernate
10        Session session = HibernateUtil.getSessionFactory().openSession();
11        Transaction transaction = null;
12
13        try {
14            transaction = session.beginTransaction();
15
16            // Crear nuevo empleado
17            Emp newEmp = new Emp();
18            newEmp.setEmpno(8000); // ID único
19            newEmp.setName("ROBINSON");
20            newEmp.setJob("CLERK");
21            newEmp.setMgr(7698);
22            newEmp.setHiredate(new Date());
23            newEmp.setSal(1200.00);
24            newEmp.setComm(null);
25            newEmp.setDept(10);
26
27            // Guardar empleado
28            session.save(newEmp);
29
30            transaction.commit();
31            System.out.println("Empleado insertado con éxito.");
32        } catch (Exception e) {
33            if (transaction != null) transaction.rollback();
34            e.printStackTrace();
35        } finally {
36            session.close();
37        }
38    }
39 }

```

Mostrar listado.



```

1 package AD04.garcia_herrero_beatriz_AD04;
2
3 import org.hibernate.Session;
4
5
6
7 public class ListarEmpleados {
8     public static void main(String[] args) {
9         // Obtén la sesión de Hibernate
10        Session session = HibernateUtil.getSessionFactory().openSession();
11
12        try {
13            // Inicia una transacción
14            session.beginTransaction();
15
16            // Consulta HQL con JOIN
17            String hql = "SELECT e.empno, e.ename, e.sal, d.dname, d.loc " +
18                "FROM Emp e JOIN e.dept d";
19
20            Query<Object[]> query = session.createQuery(hql, Object[].class);
21            List<Object[]> results = query.getResultList();
22
23            // Muestra los resultados
24            for (Object[] row : results) {
25                System.out.println("EMPNO: " + row[0] + ", ENAME: " + row[1] +
26                    ", SAL: " + row[2] + ", DNAME: " + row[3] +
27                    ", LOC: " + row[4]);
28            }
29
30            // Confirma la transacción
31            session.getTransaction().commit();
32        } catch (Exception e) {
33            e.printStackTrace();
34            if (session.getTransaction() != null) {
35                session.getTransaction().rollback();
36            }
37        } finally {
38            session.close();
39            HibernateUtil.shutdown();
40        }
41    }
42 }

```

## El código de HibernateUtil.java

