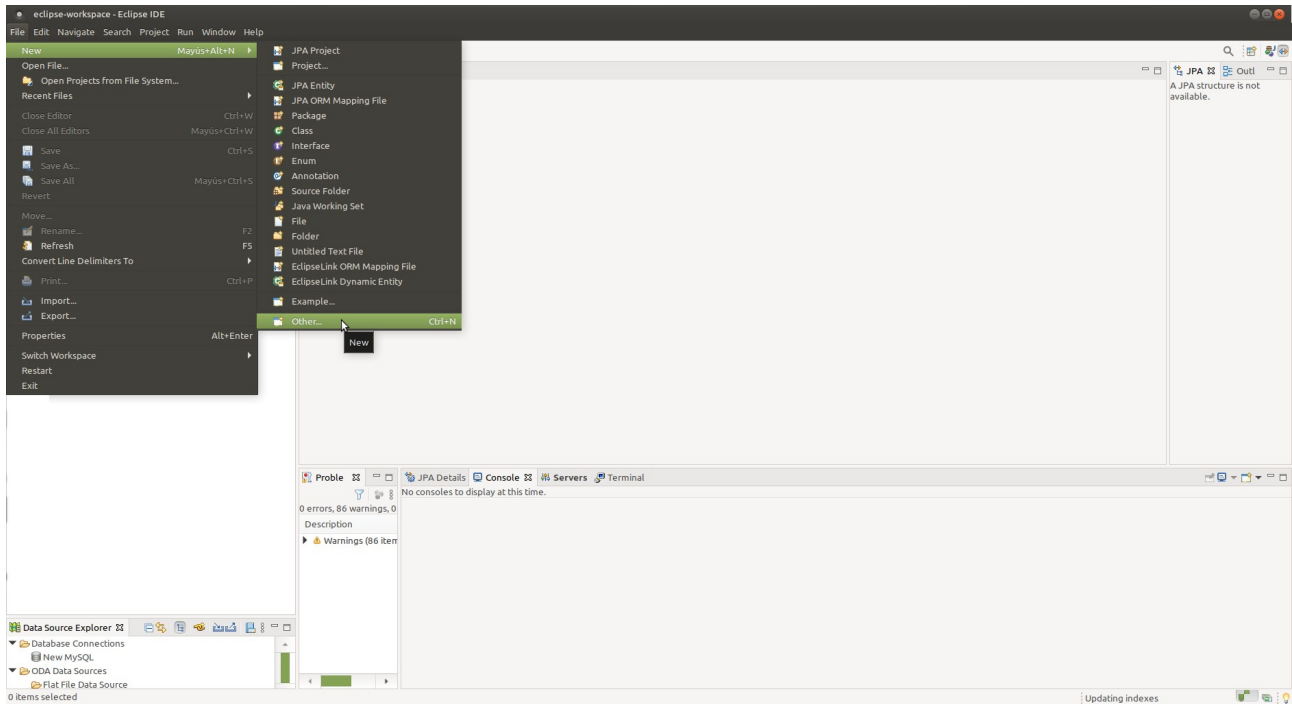
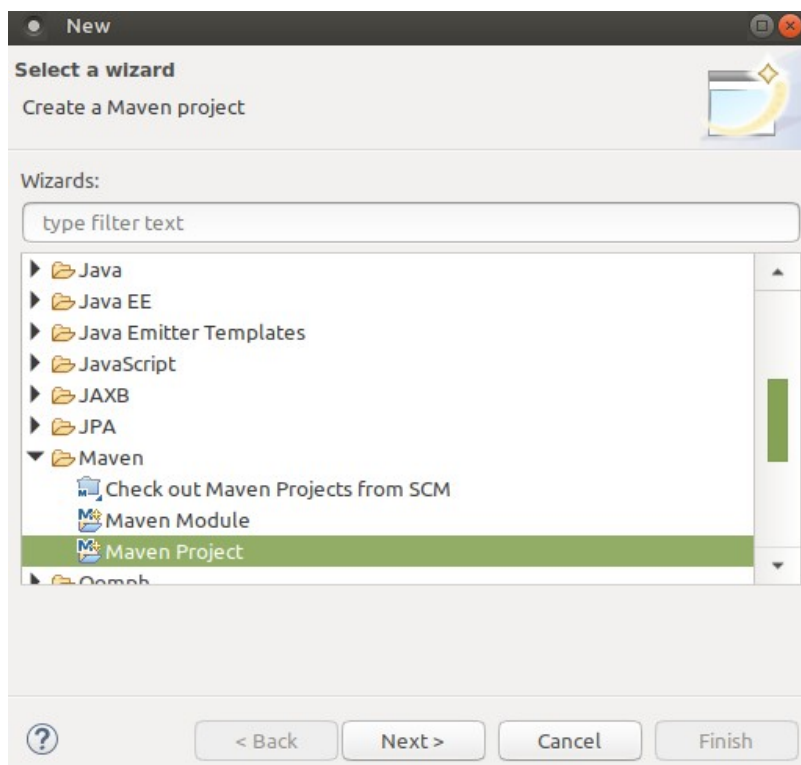


Paso a paso Proyecto maven-eclipse-jpa con mysql

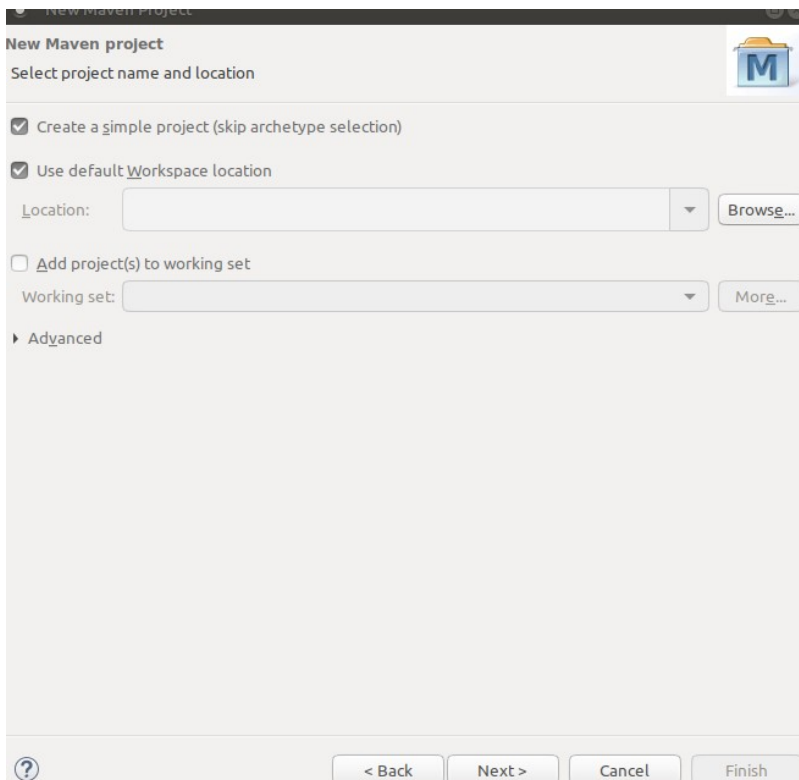
Creamos un proyecto maven en eclipse: File->New->Other



En la selección de wizard: Maven → Maven Project

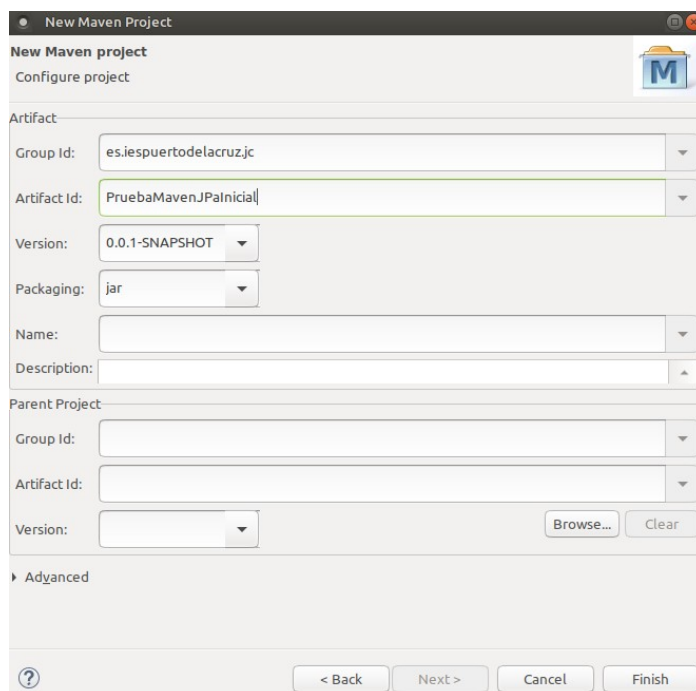


Marcamos el checkbox: Create a simple project (skip archetype selection)



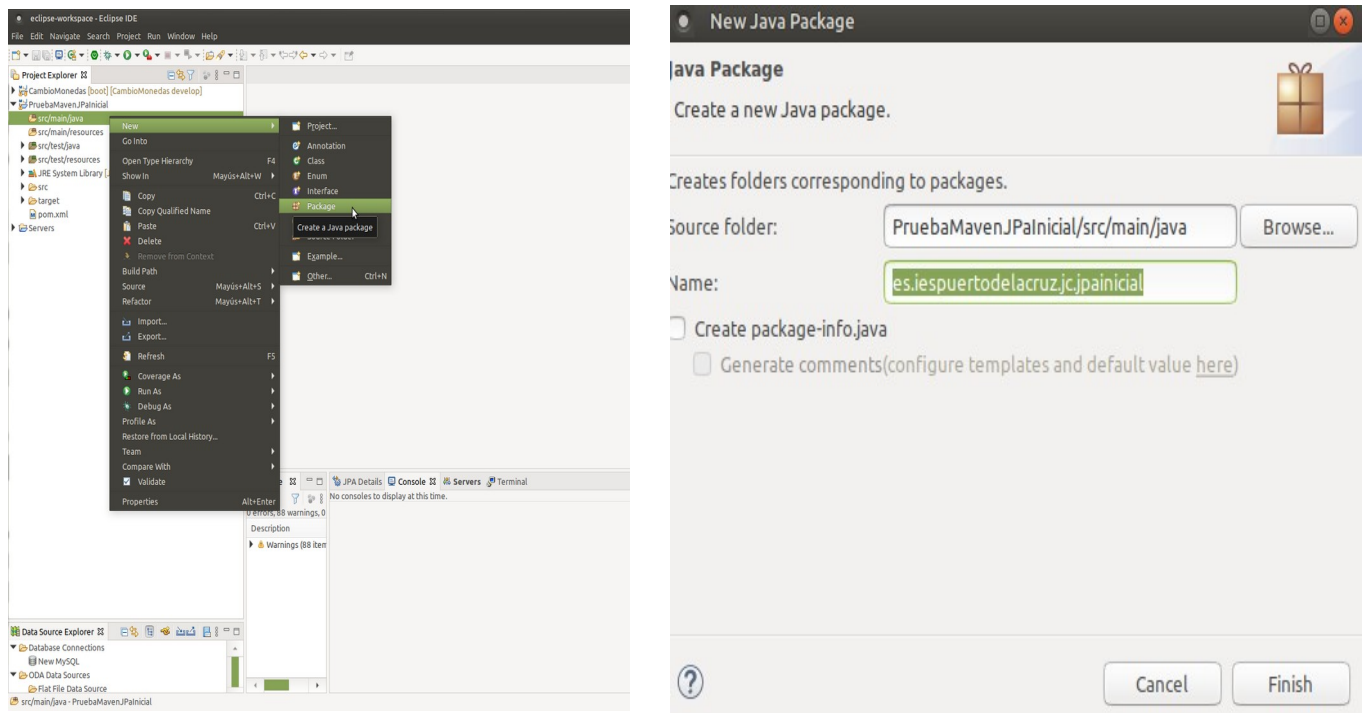
The screenshot shows the 'New Maven Project' dialog box with the title 'New Maven project' and subtitle 'Select project name and location'. The 'Create a simple project (skip archetype selection)' checkbox is checked. The 'Use default Workspace location' checkbox is also checked. The 'Location' field is empty with a 'Browse...' button next to it. The 'Add project(s) to working set' checkbox is unchecked. The 'Working set' field is empty with a 'More...' button next to it. The 'Advanced' section is collapsed. At the bottom, there are buttons for '< Back', 'Next >', 'Cancel', and 'Finish'.

Damos los datos maven de nuestro proyecto (group id y Artifact id) donde Artifact id es el nombre del proyecto en sí mismo

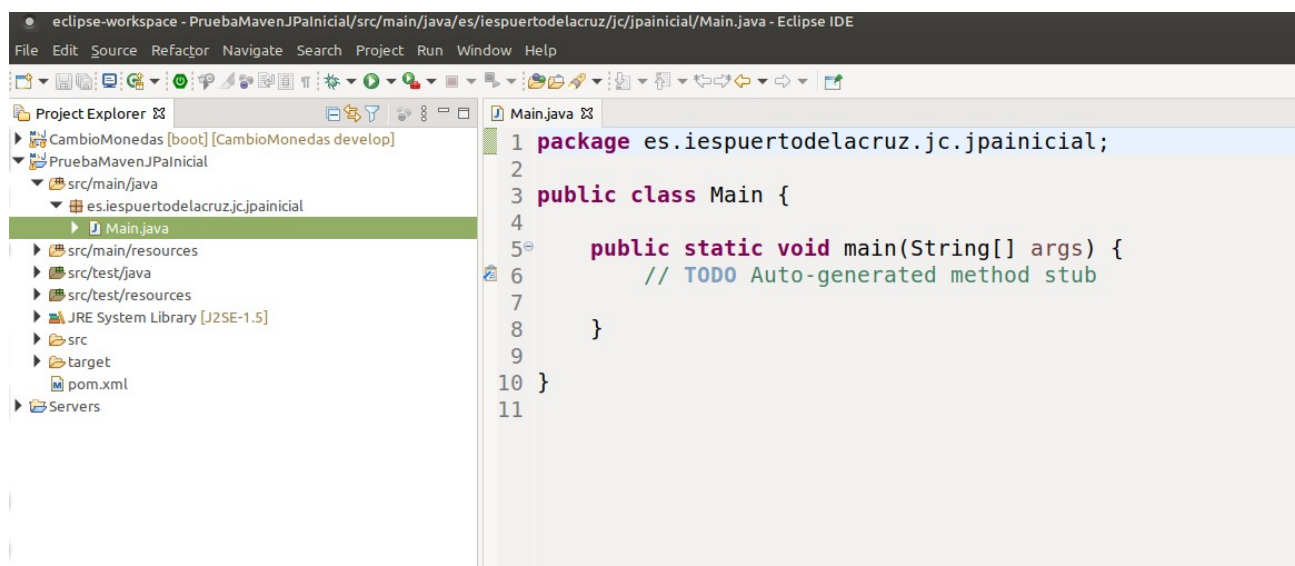


The screenshot shows the 'New Maven Project' dialog box with the title 'New Maven project' and subtitle 'Configure project'. The 'Artifact' section contains the following fields: 'Group Id' with the value 'es.iespuertodelacruz.jc', 'Artifact Id' with the value 'PruebaMavenJPalicial', 'Version' with the value '0.0.1-SNAPSHOT', and 'Packaging' with the value 'jar'. The 'Name' and 'Description' fields are empty. The 'Parent Project' section contains the following fields: 'Group Id', 'Artifact Id', and 'Version', all of which are empty. There are 'Browse...' and 'Clear' buttons next to the 'Version' field. The 'Advanced' section is collapsed. At the bottom, there are buttons for '?', '< Back', 'Next >', 'Cancel', and 'Finish'.

Creamos el paquete raíz a partir del cual vamos a trabajar. Nos ponemos encima de: src/main/java y le damos a agregar un paquete:



Vamos a establecer una clase con un método main (en este ejemplo que no tiene mayor interés se llamará: Main)



Ahora vamos a agregar al pom.xml que nos ha generado la información de mysql, hibernate:

pom.xml

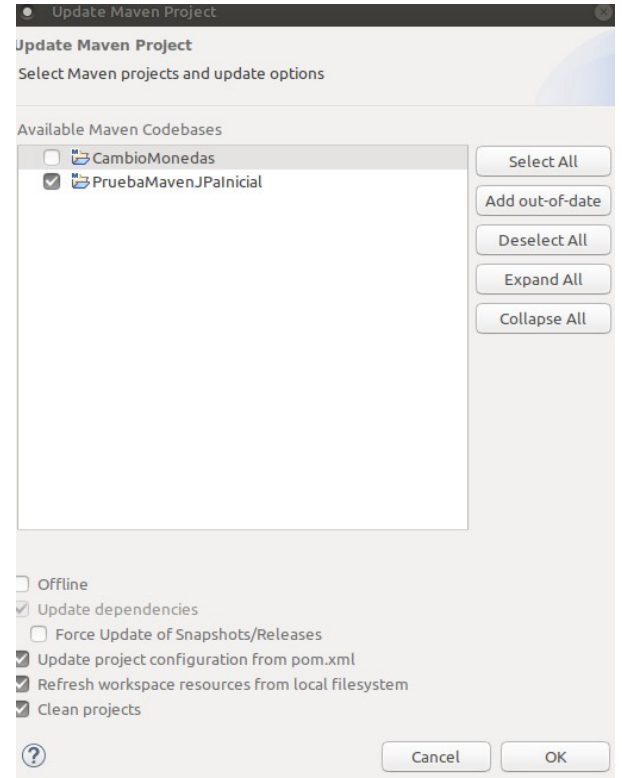
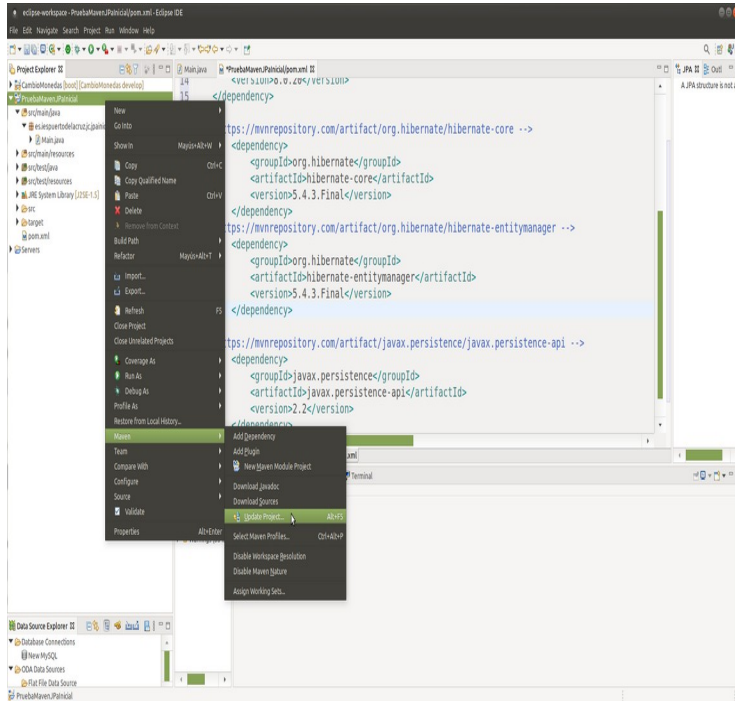
```
<project xmlns="http://maven.apache.org/POM/4.0.0"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://maven.apache.org/POM/4.0.0
https://maven.apache.org/xsd/maven-4.0.0.xsd">
  <modelVersion>4.0.0</modelVersion>
  <groupId>es.iespuertodelacruz.jc</groupId>
  <artifactId>PruebaMavenJPaInicial</artifactId>
  <version>0.0.1-SNAPSHOT</version>
  <properties>
    <maven.compiler.target>1.8</maven.compiler.target>
    <maven.compiler.source>1.8</maven.compiler.source>
  </properties>
  <dependencies>
    <dependency>
      <groupId>mysql</groupId>
      <artifactId>mysql-connector-java</artifactId>
      <version>8.0.20</version>
    </dependency>

<!-- https://mvnrepository.com/artifact/org.hibernate/hibernate-core -->
    <dependency>
      <groupId>org.hibernate</groupId>
      <artifactId>hibernate-core</artifactId>
      <version>5.4.3.Final</version>
    </dependency>
<!-- https://mvnrepository.com/artifact/org.hibernate/hibernate-entitymanager -->
    <dependency>
      <groupId>org.hibernate</groupId>
      <artifactId>hibernate-entitymanager</artifactId>
      <version>5.4.3.Final</version>
    </dependency>

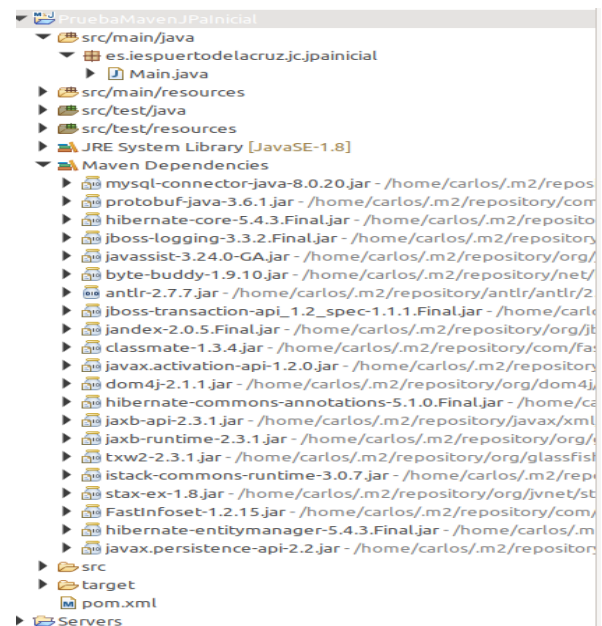
<!-- https://mvnrepository.com/artifact/javax.persistence/javax.persistence-api -->
    <dependency>
      <groupId>javax.persistence</groupId>
      <artifactId>javax.persistence-api</artifactId>
      <version>2.2</version>
    </dependency>
  </dependencies>
</project>
```

Para que actualice correctamente le decimos a eclipse que actualice Maven

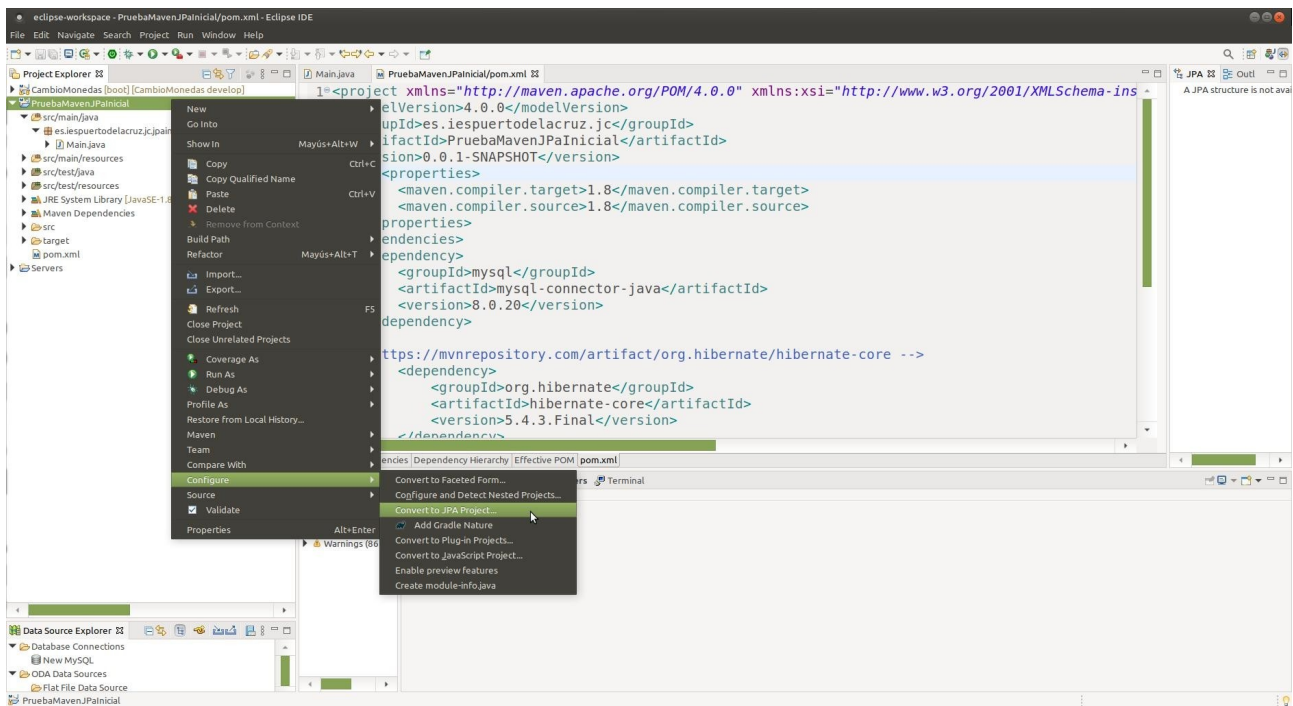
Para ello nos posicionamos sobre el proyecto botón derecho → Maven → update project



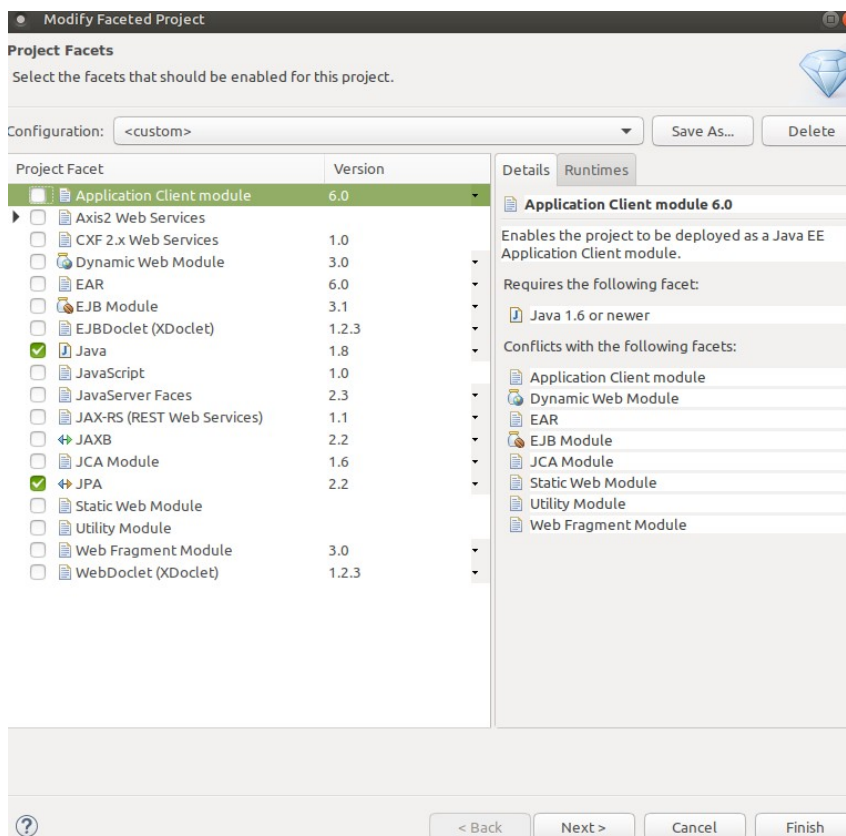
Con lo anterior veremos que en: Maven Dependencies nos aparece mysql, hibernate, etc



Ahora para la generación automática de las Entities vamos a convertir el proyecto en un proyecto JPA: Botón derecho sobre proyecto- → Configure → Convert to jpa project

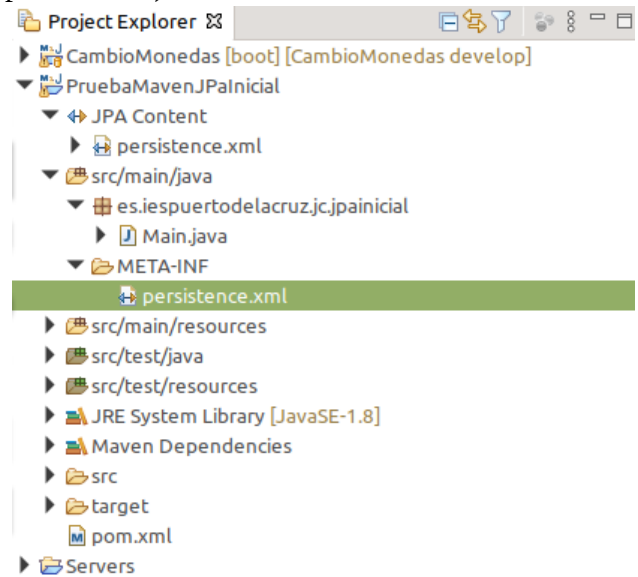


En la ventana seleccionamos Java 1.8 y que esté seleccionado JPA

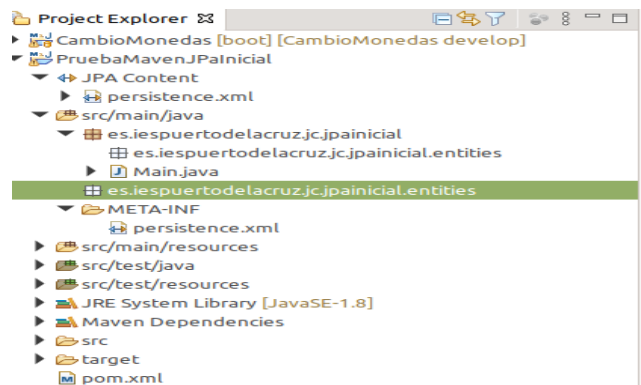


Pulsamos en el botón finish (no hace falta pulsar next)

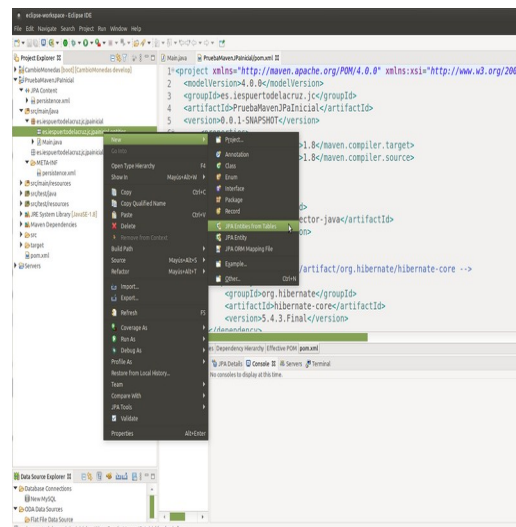
Nos habrá creado un fichero persistence.xml



Ahora crearemos el paquete donde vamos a poner nuestras entities. Por ejemplo: es.iespuertodelacruz.jc.jpainicial.entities



Nos ponemos encima del paquete, botón derecho
→ New → JPA entities from tables

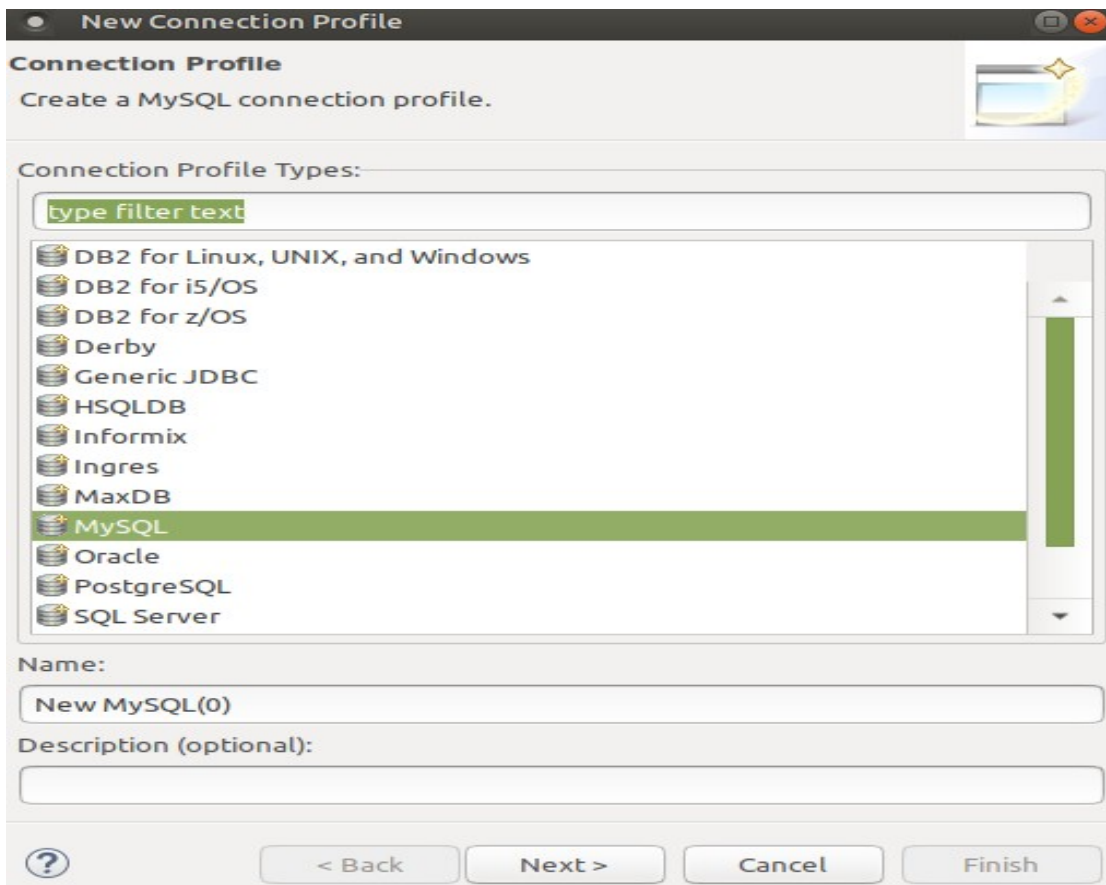


Nos aparece un wizard donde habrá que seleccionar una conexión. Como no tenemos aún ninguna pulsamos sobre el icono amarillo de nueva conexión (observar que lo hemos rodeado con un círculo para detectarlo fácilmente)

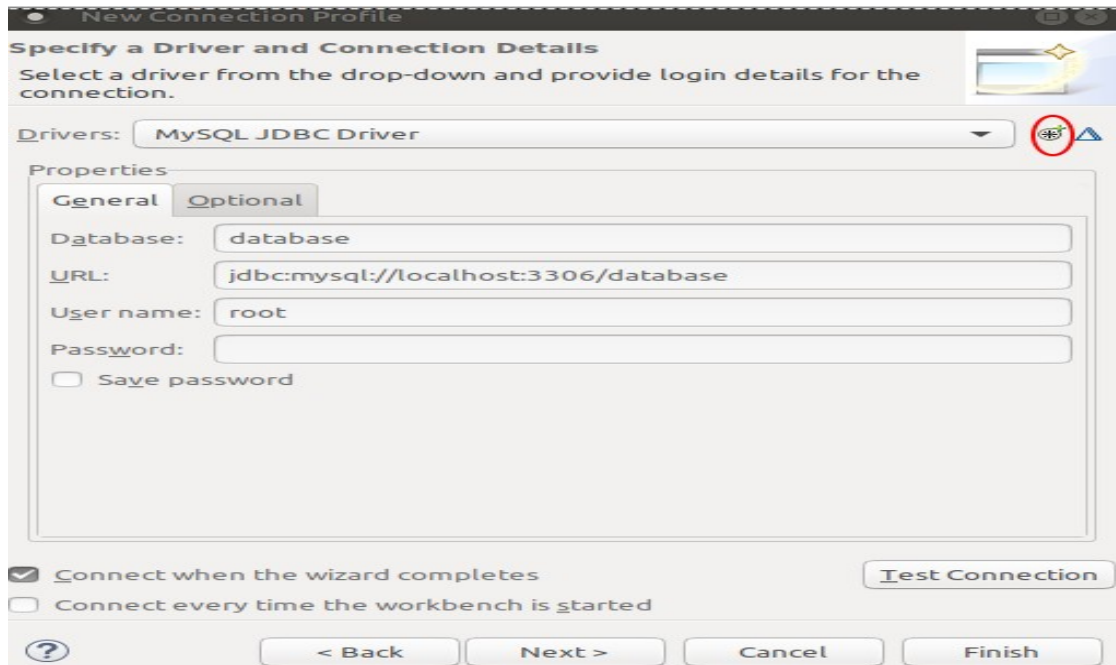
nota: cuidado! Si no tenemos arrancado el servicio de mysql no va a terminar conectando nuestro proyecto



Abre un nuevo wizard para crear la nueva conexión. Seleccionamos Mysql y pulsamos next

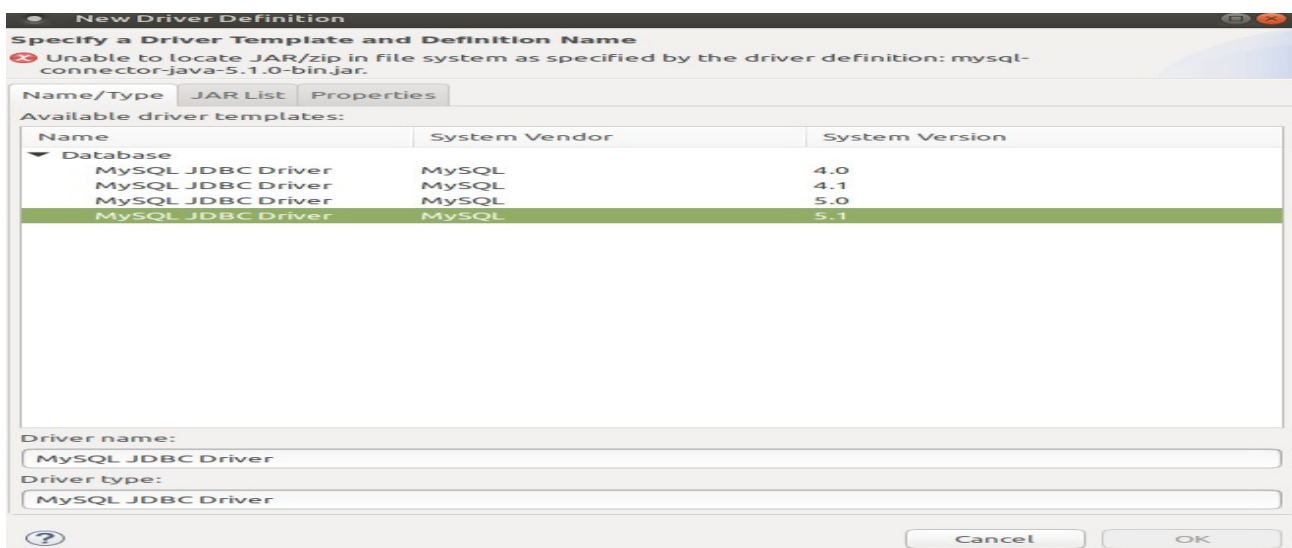


En la siguiente ventana del wizard se detallarán las especificidades de la conexión (nombre de la base de datos, usuario, etc) pero adicionalmente nosotros vamos a tener que hacer un paso más ya que aún no hemos cargado un driver mysql para que Eclipse pueda conectar con la base de datos

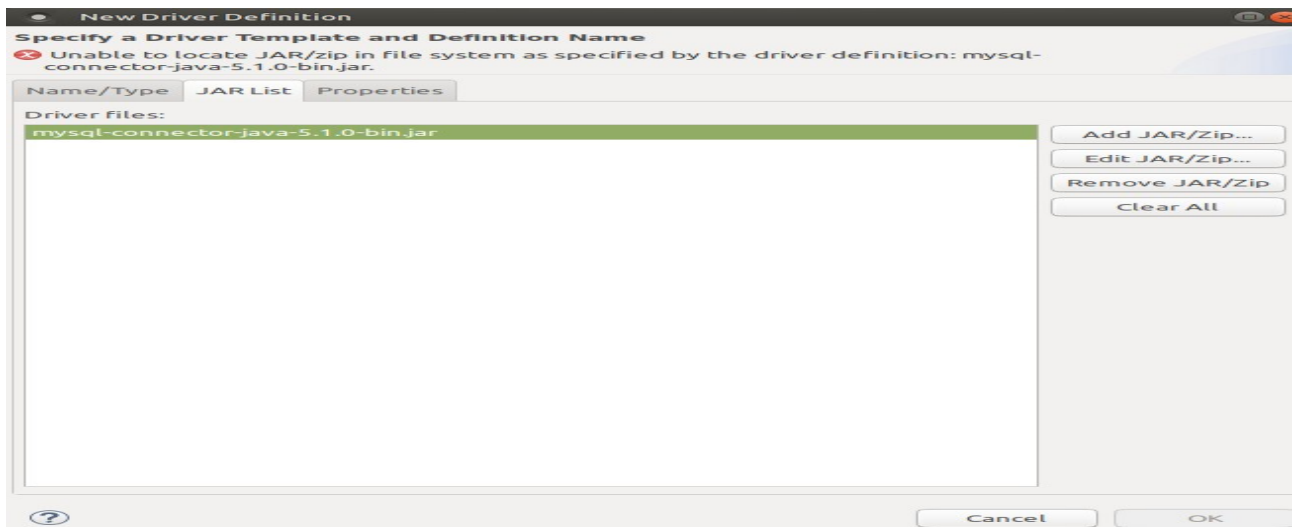


Se ha marcado con un círculo rojo el botón que hay que pulsar para decirle a eclipse donde tenemos el driver para que se pueda conectar a la base de datos (en contextual el icono dice: New driver definition)

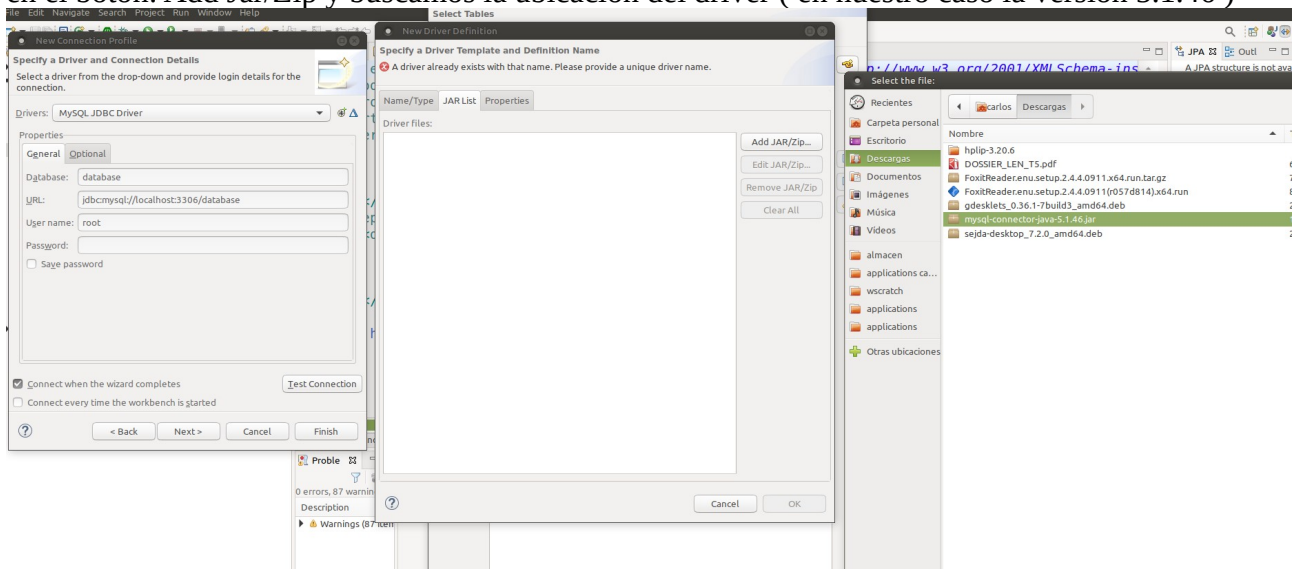
Al pulsar en el botón nos aparece un nuevo wizard. En la primera pestaña seleccionamos la versión 5.1 del driver



Luego pulsamos sobre la pestaña: Jar List y borramos el driver que tiene establecido:



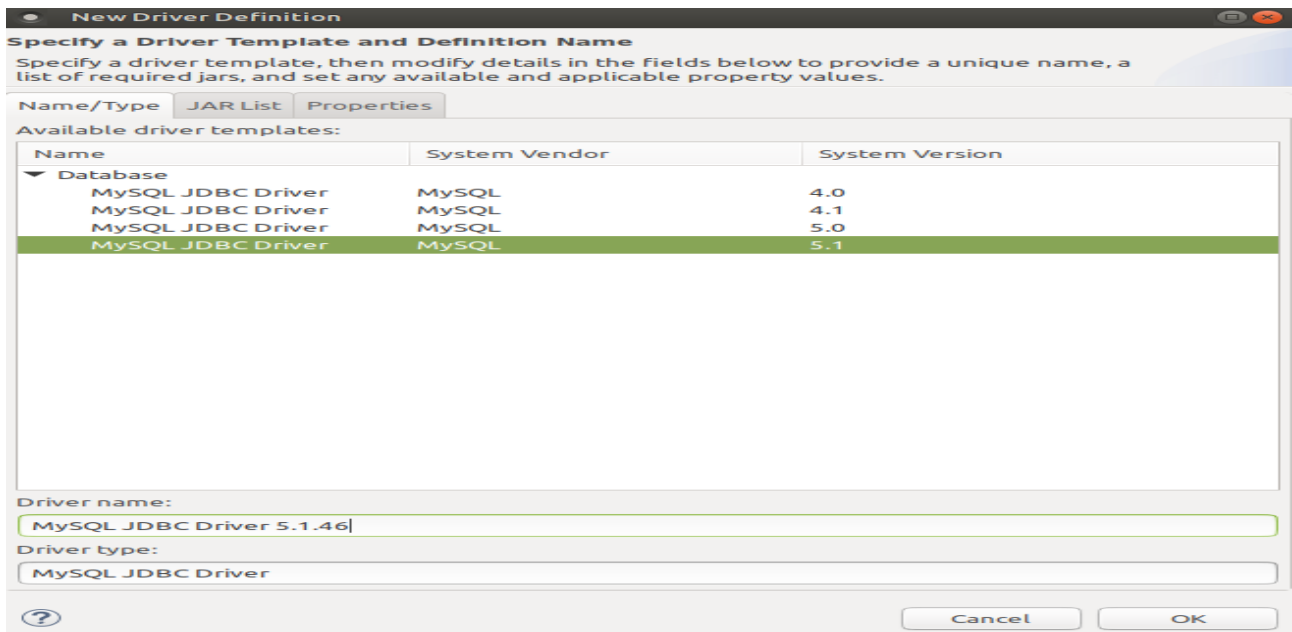
Después agregamos el driver que tengamos descargado en el disco duro. Para eso pulsamos en el botón: Add Jar/Zip y buscamos la ubicación del driver (en nuestro caso la versión 5.1.46)



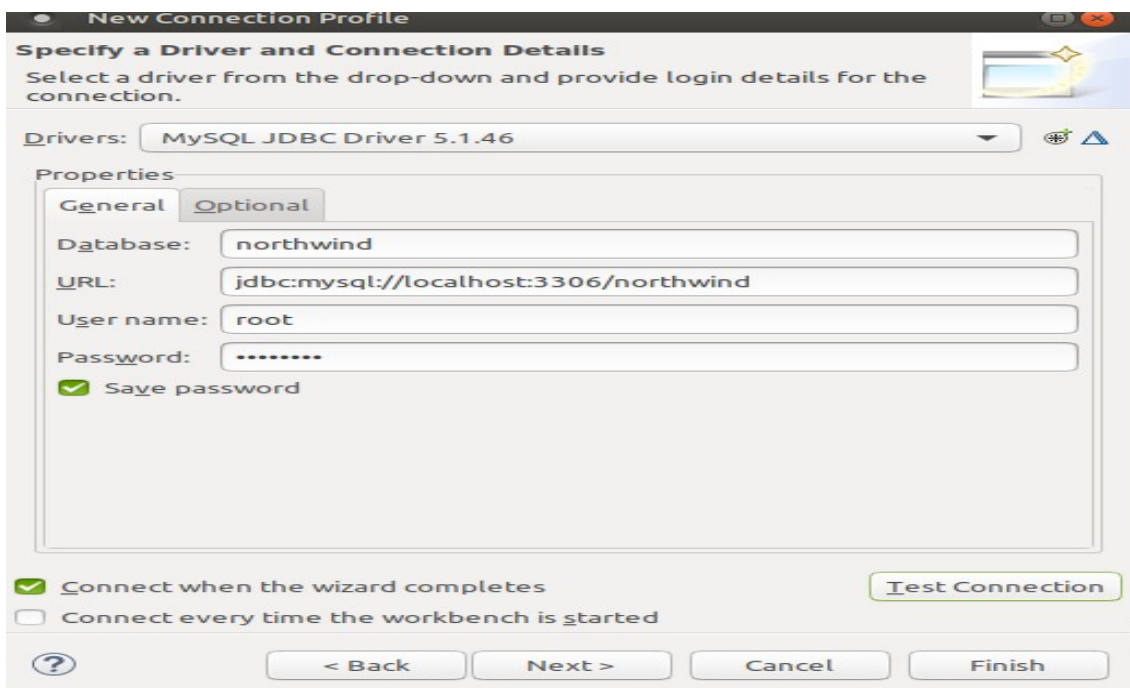
Ahora la pestaña Jar List nos quedará:



Observar que no nos permitirá pulsar en OK si no le hemos puesto un nombre correcto al Driver. En Este caso se muestra un ejemplo con el nombre 5.1.46:



Una vez cargado el driver para que eclipse se conecte terminamos de rellenar los datos de conexión en el wizard. Vamos a ver un ejemplo de conexión a base de datos northwind en local (Mysql está instalado en el mismo ordenador que tenemos eclipse)



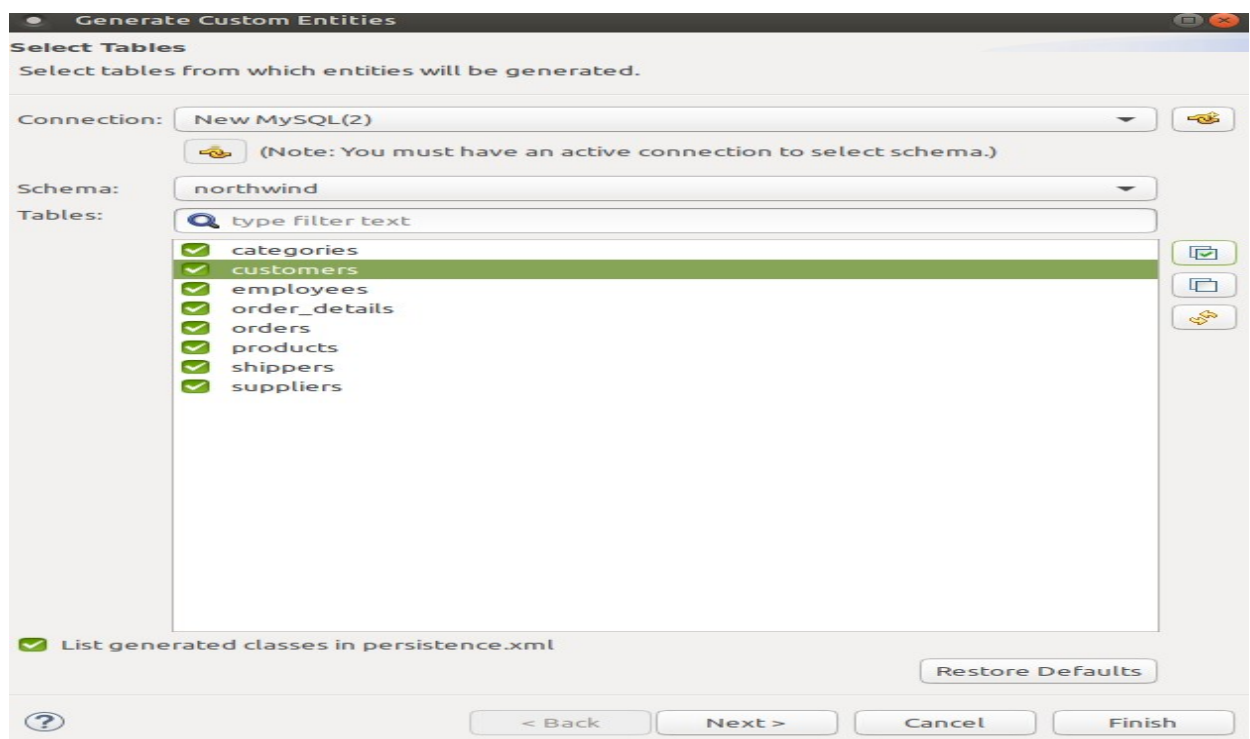
Una vez establecido no es necesario pulsar en next. Basta con pulsar en: Finish

Con lo anterior se cierra el wizard y nos devuelve al wizard inicial que era el de generación de entities desde la base de datos.

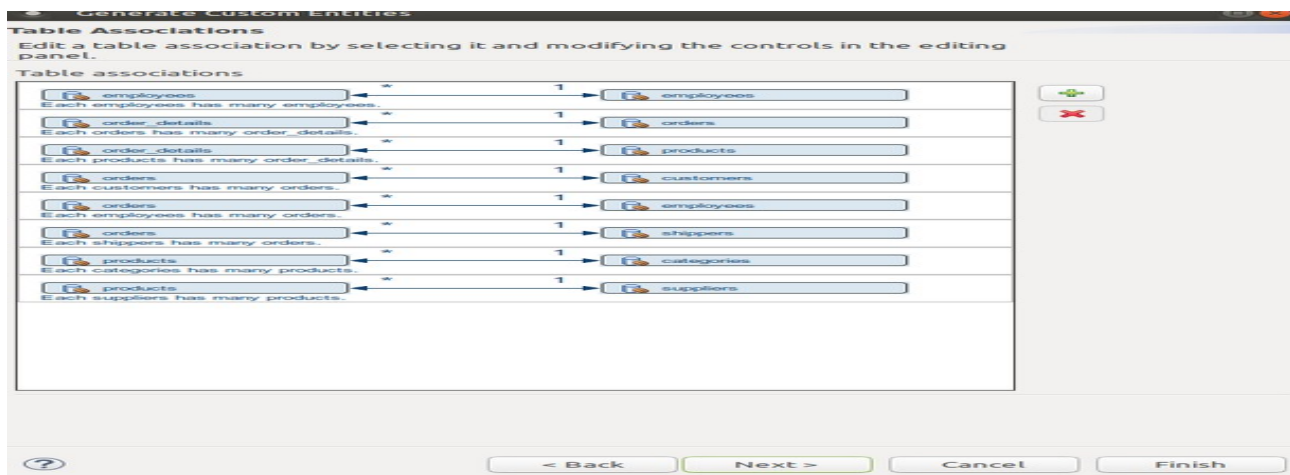
Tenemos que tener seleccionado en: Connection la conexión que hemos creado hace un momento (en el ejemplo se llamó New Mysql(2))

Tenemos que tener seleccionado en: Schema la base de datos que queremos atacar. En este caso de ejemplo: northwind

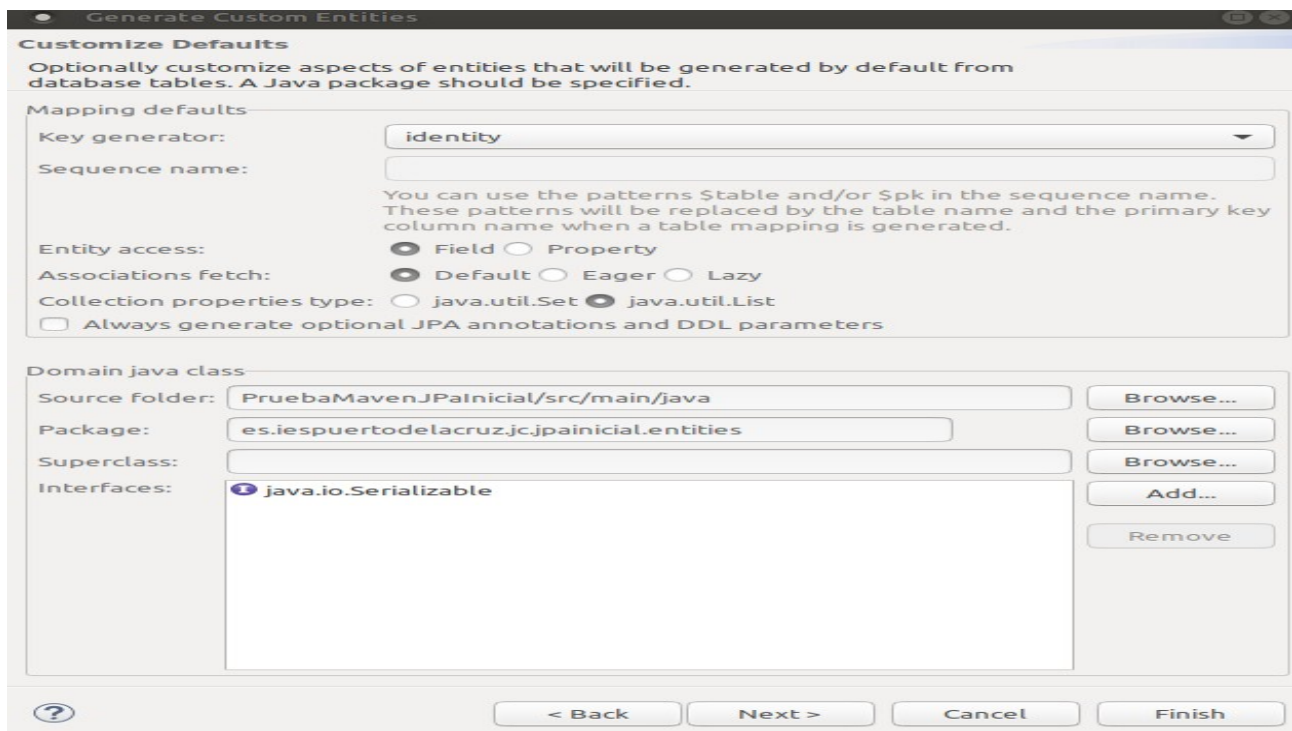
Nos mostrará las diferentes tablas de la base de datos. Nosotros seleccionaremos aquellas que queramos incluir. En este ejemplo las tomaremos todas



Avanzamos el wizard con next:



En esa ventana no hay nada que hacer ahora y pulsamos en next:



Esta ventana tiene varias opciones importantes. Como sabemos que northwind tiene autonuméricos vamos a seleccionar en: Key generator la opción: identity

También vemos que podemos elegir entre Eager y Lazy eso es algo importante pero ahora no lo vamos a tocar

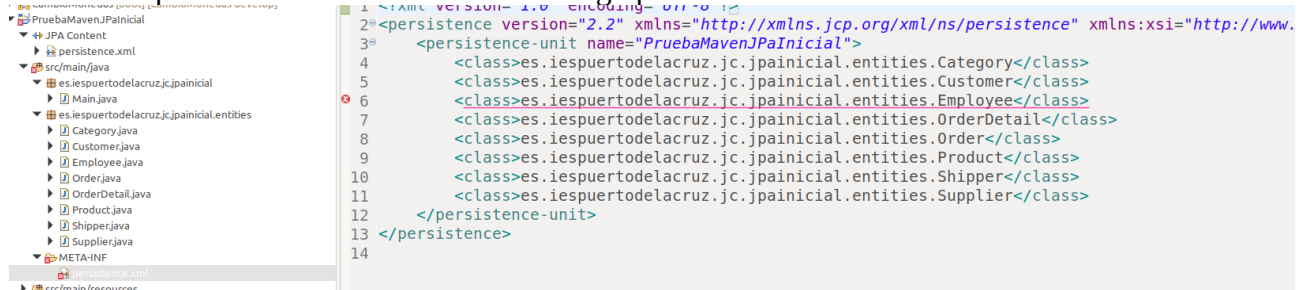
De igual forma también podemos elegir entre listas y conjuntos (list, set) de momento también lo dejamos todo como viene y pulsamos en: Finish

Ahora podemos observar que nos ha creado las Entities:



Para que todo funcione correctamente nos falta agregar un par de cosas en el fichero **persistence.xml**

Después de los wizard nos mostrará algo parecido a:



Nosotros le vamos a agregar la información de hibernate y los datos de conexión:

En este caso que nos conectamos a la propia máquina (localhost) la base da datos northwind y le ponemos el parámetro de serverTimezone:

```
<property name="javax.persistence.jdbc.url" value="jdbc:mysql://localhost:3306/northwind?serverTimezone=UTC"/>
```

Una imagen de como quedaría es:

```
1 <?xml version="1.0" encoding="UTF-8"?>
2 <persistence version="2.1" xmlns="http://xmlns.jcp.org/xml/ns/persistence" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
3   xsi:schemaLocation="http://xmlns.jcp.org/xml/ns/persistence http://xmlns.jcp.org/xml/ns/persistence/persistence_2_1.xsd">
4   <persistence-unit name="unidadPersistencia" transaction-type="RESOURCE_LOCAL">
5     <provider>org.hibernate.jpa.HibernatePersistenceProvider</provider>
6     <class>es.iespuertodelacruz.jc.jpainicial.entities.Category</class>
7     <class>es.iespuertodelacruz.jc.jpainicial.entities.Customer</class>
8     <class>es.iespuertodelacruz.jc.jpainicial.entities.Employee</class>
9     <class>es.iespuertodelacruz.jc.jpainicial.entities.OrderDetail</class>
10    <class>es.iespuertodelacruz.jc.jpainicial.entities.Order</class>
11    <class>es.iespuertodelacruz.jc.jpainicial.entities.Product</class>
12    <class>es.iespuertodelacruz.jc.jpainicial.entities.Shipper</class>
13    <class>es.iespuertodelacruz.jc.jpainicial.entities.Supplier</class>
14    <properties>
15      <property name="javax.persistence.jdbc.url" value="jdbc:mysql://localhost:3306/northwind?serverTimezone=UTC"/>
16      <property name="javax.persistence.jdbc.user" value="root"/>
17      <property name="javax.persistence.jdbc.driver" value="com.mysql.cj.jdbc.Driver"/>
18      <property name="javax.persistence.jdbc.password" value="1q2w3e4r"/>
19      <property name="hibernate.dialect" value="org.hibernate.dialect.MySQLDialect" />
20      <property name="hibernate.show_sql" value="true" />
21    </properties>
22  </persistence-unit>
</persistence>
```

Para poder copiar lo vemos como texto el fichero persistence.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<persistence version="2.1" xmlns="http://xmlns.jcp.org/xml/ns/persistence"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://xmlns.jcp.org/xml/ns/persistence
http://xmlns.jcp.org/xml/ns/persistence/persistence_2_1.xsd">
  <persistence-unit name="unidadPersistencia" transaction-type="RESOURCE_LOCAL">
    <provider>org.hibernate.jpa.HibernatePersistenceProvider</provider>
    <class>es.iespuertodelacruz.jc.jpainicial.entities.Category</class>
    <class>es.iespuertodelacruz.jc.jpainicial.entities.Customer</class>
    <class>es.iespuertodelacruz.jc.jpainicial.entities.Employee</class>
    <class>es.iespuertodelacruz.jc.jpainicial.entities.OrderDetail</class>
    <class>es.iespuertodelacruz.jc.jpainicial.entities.Order</class>
    <class>es.iespuertodelacruz.jc.jpainicial.entities.Product</class>
    <class>es.iespuertodelacruz.jc.jpainicial.entities.Shipper</class>
    <class>es.iespuertodelacruz.jc.jpainicial.entities.Supplier</class>
    <properties>
      <property name="javax.persistence.jdbc.url" value="jdbc:mysql://localhost:3306/northwind?serverTimezone=UTC"/>
      <property name="javax.persistence.jdbc.user" value="root"/>
      <property name="javax.persistence.jdbc.driver" value="com.mysql.cj.jdbc.Driver"/>
      <property name="javax.persistence.jdbc.password" value="1q2w3e4r"/>
      <property name="hibernate.dialect" value="org.hibernate.dialect.MySQLDialect" />
      <property name="hibernate.show_sql" value="true" />
    </properties>
  </persistence-unit>
</persistence>
```

Hemos marcado en amarillo aquellas cosas que nosotros podemos modificar habitualmente

Por ejemplo, se ha elegido ponerle el nombre: unidadPersistencia

De esa forma cuando queramos crear un entityManagerfactory escribimos:

```
EntityManagerFactory emf = Persistence.createEntityManagerFactory("unidadPersistencia");
```

También aparece marcado en amarillo los datos de la conexión que antes detallamos:

```
jdbc:mysql://localhost:3306/northwind?serverTimezone=UTC"
```


Hemos marcado en amarillo la forma en la que ponemos el nombre del usuario que conectará el programa a la base de datos:

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

La contraseña para ese usuario:

```
<property name="javax.persistence.jdbc.password" value="1q2w3e4r" />
```

Y finalmente hemos marcado una línea que se pone para hacer debug. Lo que hace es mostrarnos que ejecuta hibernate al actuar sobre la base de datos:

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

Con lo anterior ya lo tenemos todo configurado. Ahora para comprobarlo vamos a mostrar en pantalla los datos del primer empleado (empleado con id 1)

Podemos poner en el main():

```
import javax.persistence.EntityManager;
import javax.persistence.EntityManagerFactory;
import javax.persistence.Persistence;

import es.iespuertodelacruz.jc.jpainicial.entities.Employee;

public class Main {

    public static void main(String[] args) {
        EntityManagerFactory emf =
Persistence.createEntityManagerFactory("unidadPersistencia");
        EntityManager em = emf.createEntityManager();

        Employee emp = em.find(Employee.class, 1);

        System.out.println( emp.getFirstName() + " " + emp.getLastName());

    }
}
```

Para ejecutar la aplicación elegimos: Run as: Java application y buscamos la clase Main que tenemos en nuestro proyecto:

