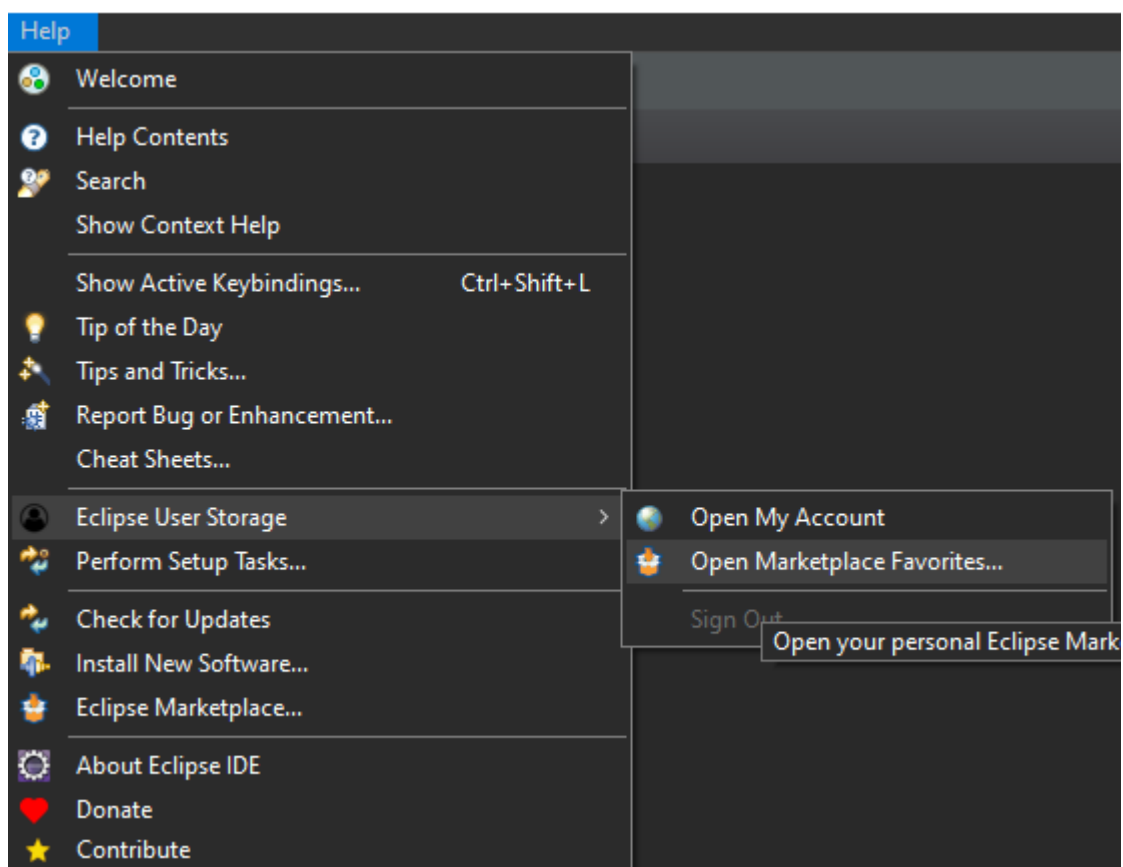


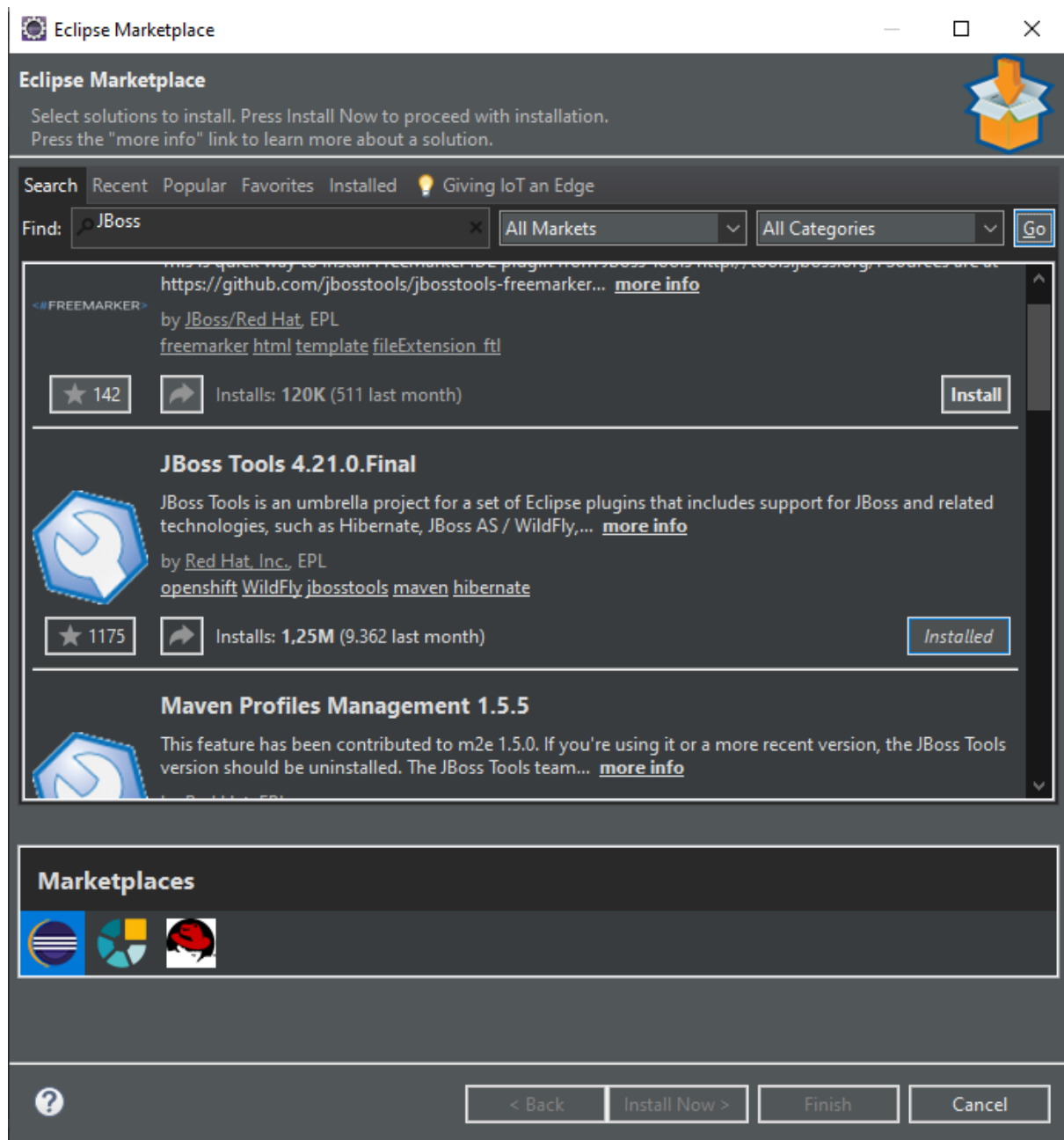
Manual instalación Hibernate

En este manual vamos a aprender a crear un proyecto con el framework Hibernate en Eclipse.

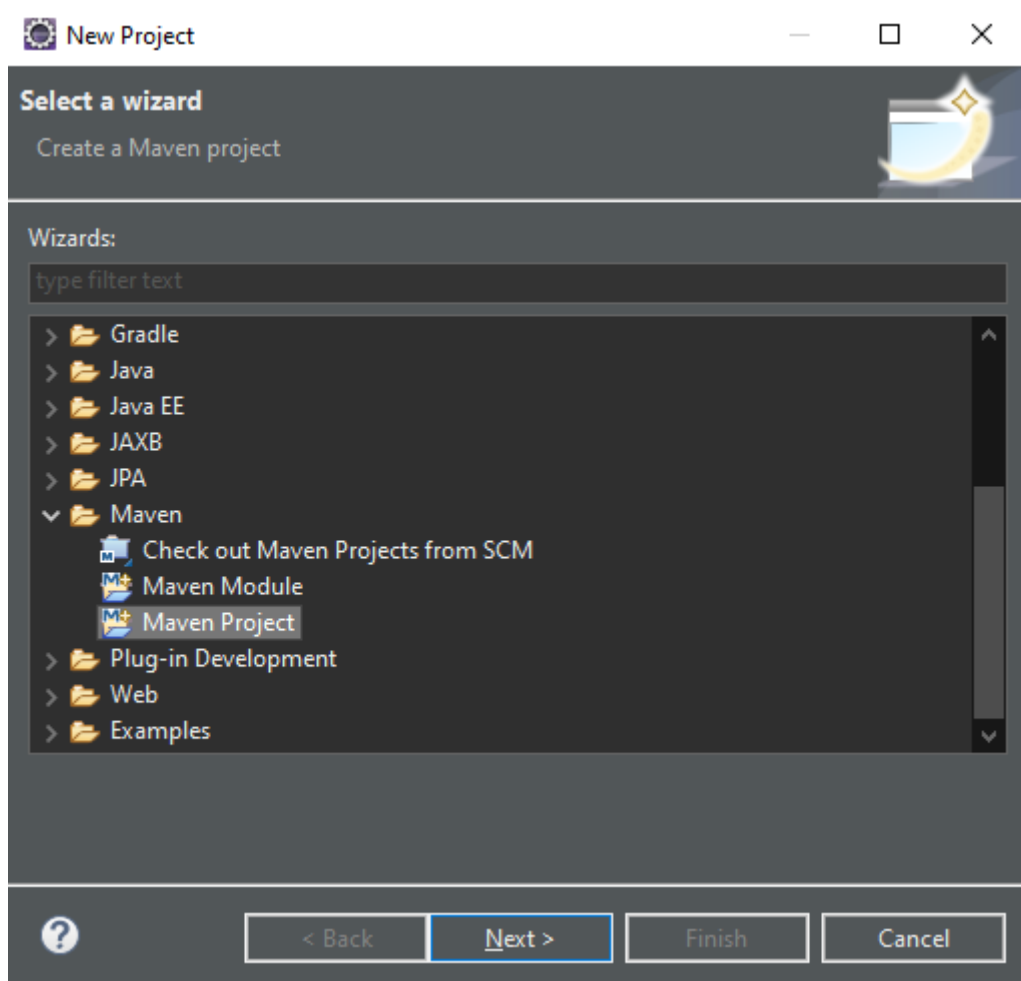
1- En primer lugar vamos a instalar los plugins necesarios de hibernate y crearemos el proyecto



En primer lugar vamos al botón 'help' en la barra de herramientas y buscamos Eclipse Marketplace o Eclipse User Storage → Open Marketplace Favorites... Ambos te llevarán a la pantalla del marketplace.

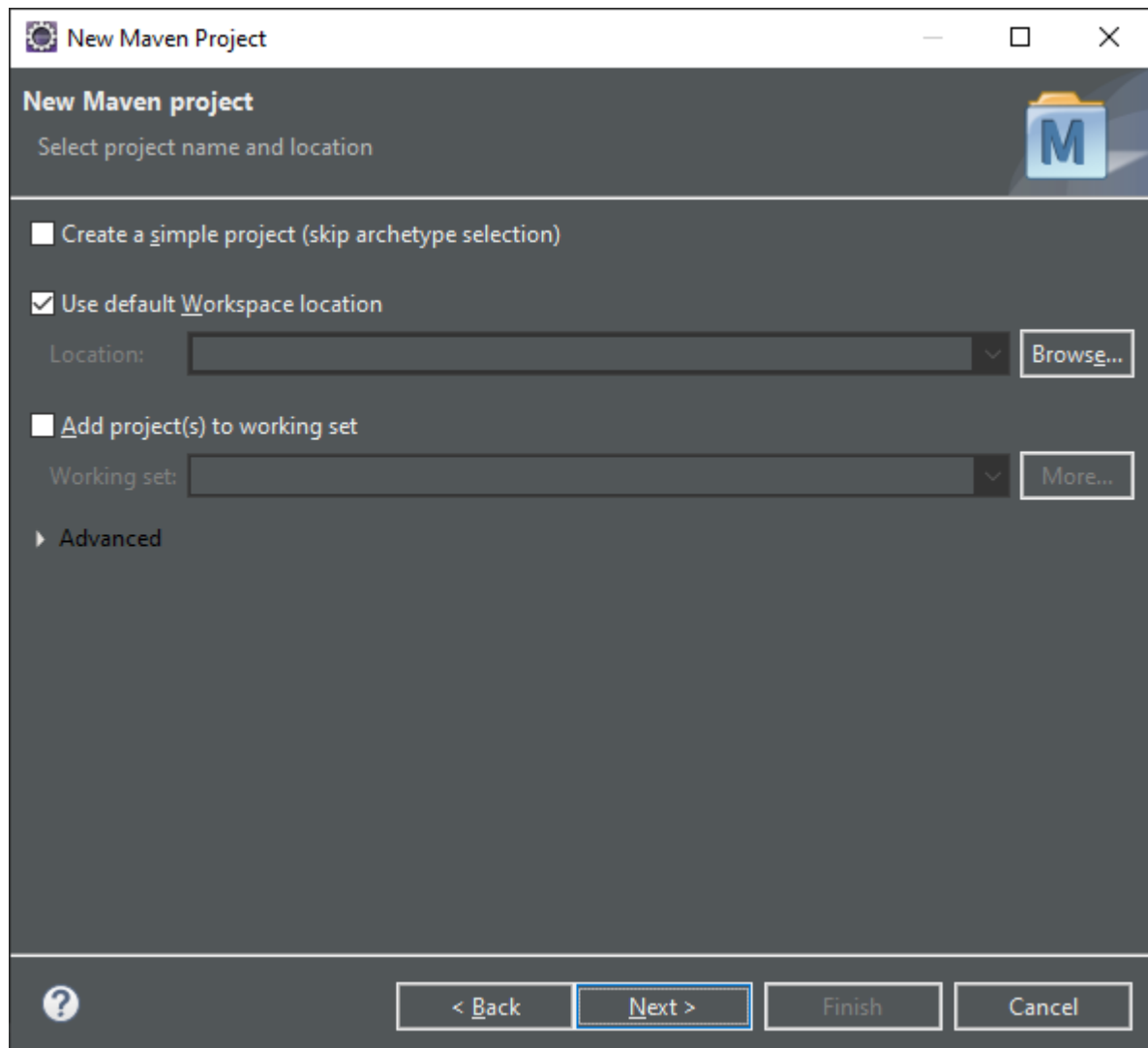


Una vez en el marketplace vamos a la pestaña 'search' y buscamos JBoss, tenemos que buscar el JBoss Tools y pulsamos en 'Install Now >'. Después le damos a 'Next> → Finish' para finalizar la instalación.

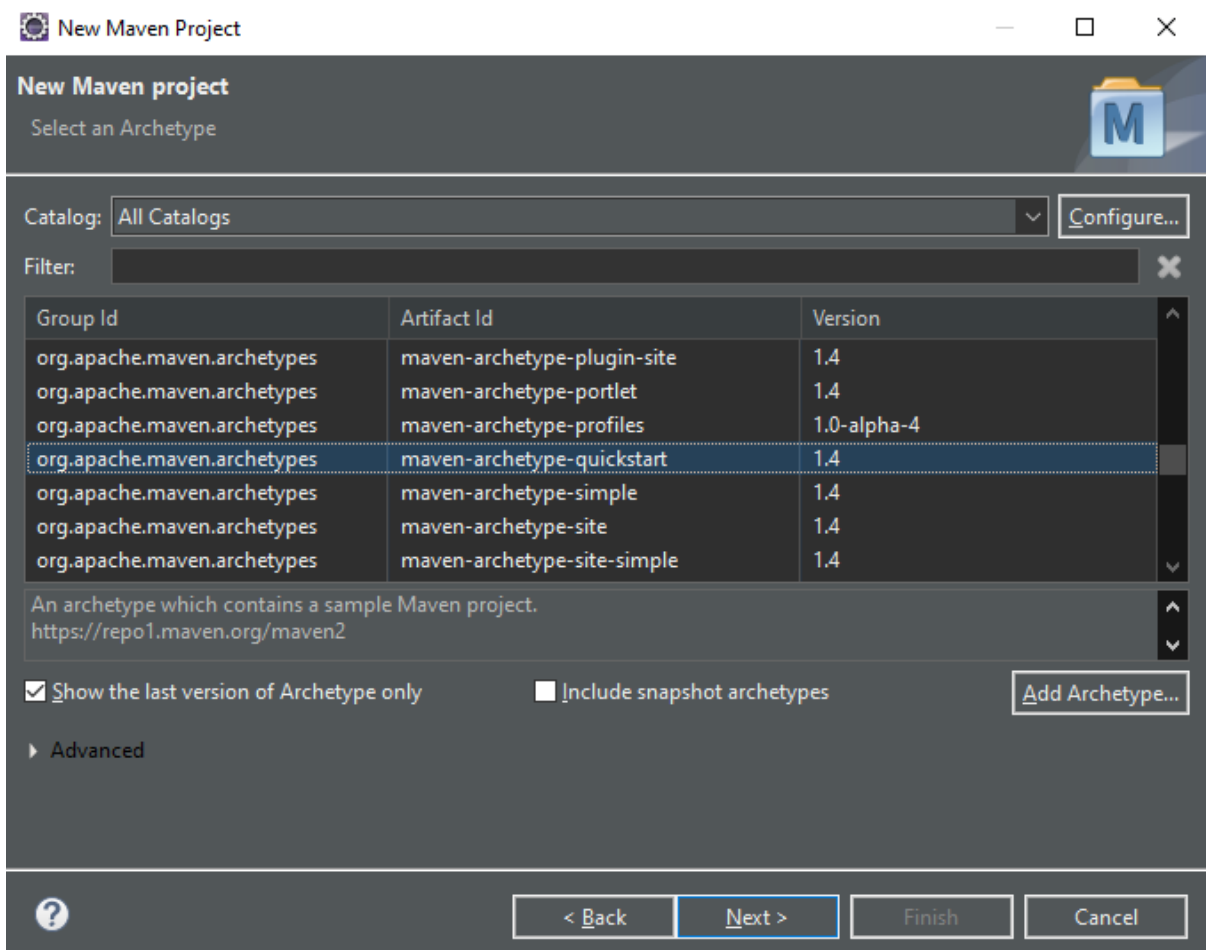


A continuación vamos a crear un proyecto Maven. En eclipse buscamos en la barra de herramientas 'File → New → Maven → Maven Project'.

Pulsamos 'Next>' y nos llevará a la siguiente ventana donde se especificará dónde vamos a guardar el proyecto.



En la ventana de localización vamos a dejar todas las opciones por defecto y pulsamos el botón 'Next>'.




En ésta ventana buscaremos la opción de 'Group ID' →
org.apache.maven.archetypes y en 'Artifact Id' → maven-archetype-quickstart.

A continuación pulsamos 'Next>' y vamos a la siguiente ventana.

New Maven Project

New Maven project

Specify Archetype parameters



Group Id:

com.conexionmaven

Artifact Id:

Hibernate

Version:

0.0.1-SNAPSHOT

Package:

com.conexionmaven.Hibernate

Properties available from archetype:

Name	Value	
		Add...
		Remove

Advanced

< Back

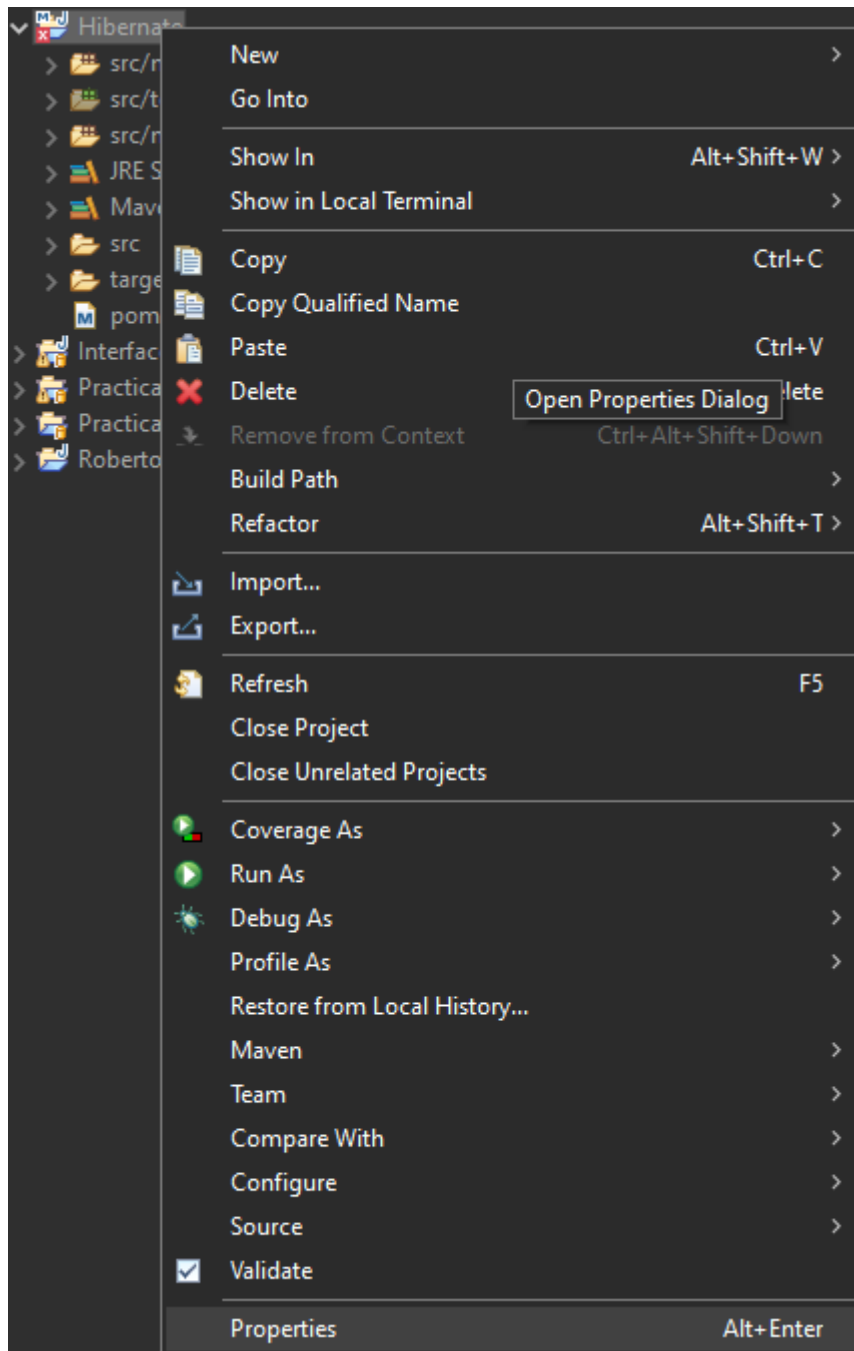
Next >

Finish

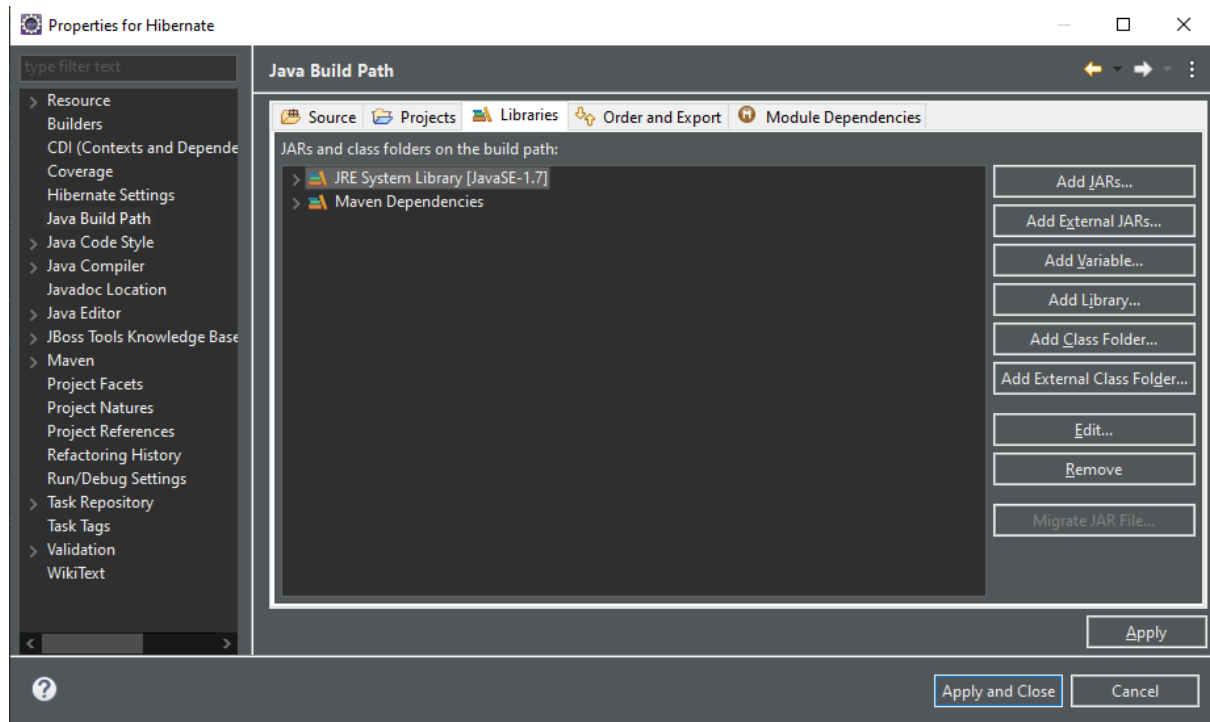
Cancel

En esta ventana simplemente vamos a pulsar el botón 'Finish'.

2- En este punto vamos a cambiar la versión del JDK del proyecto a JDK-17.

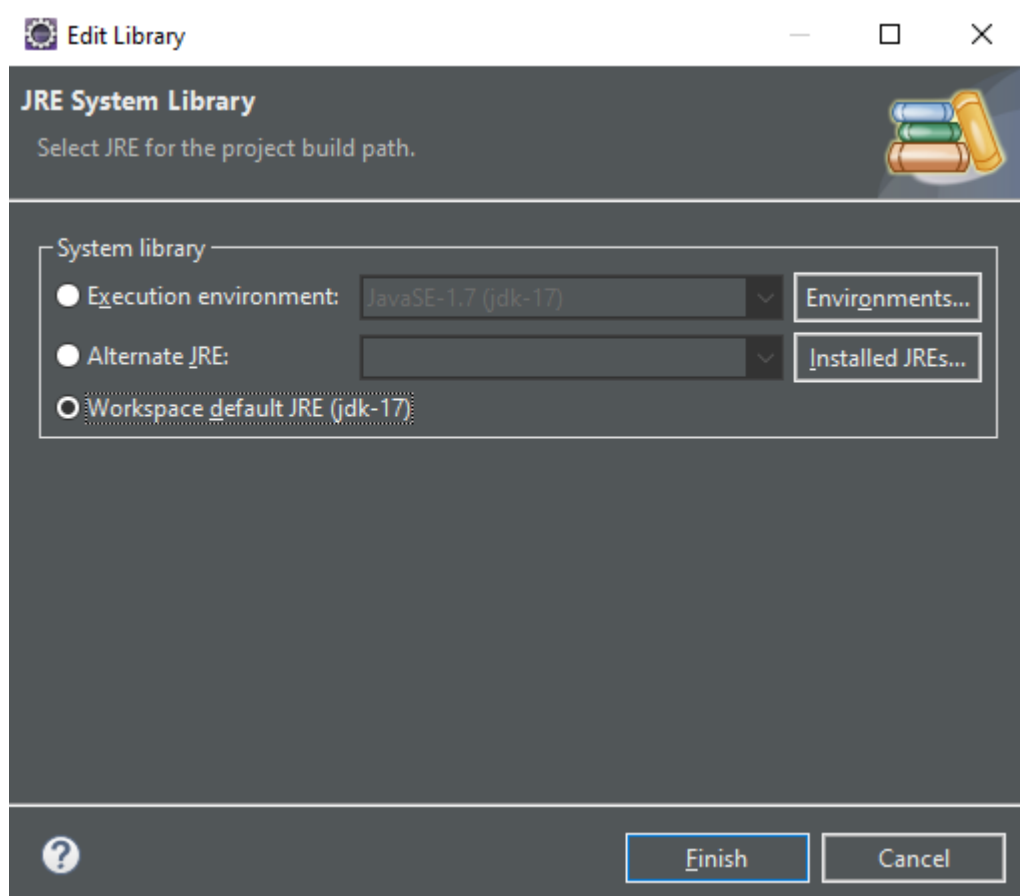


Pulsamos con el botón derecho del ratón en el proyecto y vamos a la última opción 'Properties'.



Vamos a Java Build Path en la parte izquierda de la ventana y nos vamos a la pestaña 'Libraries'.

Pinchamos en la carpeta 'JRE System Library' y en la parte derecha vamos al botón 'Edit...' y vamos a la siguiente ventana.



En ésta ventana vamos a pinchar en la opción 'Workspace default JRE (jdk-17)' y pulsamos 'Finish'.

Asegúrate de que tienes instalado el jdk-17.

3- A continuación vamos a modificar el archivo POM del proyecto para actualizar a la versión 17 de java.

```
<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>com.conexionmaven</groupId>
  <artifactId>Hibernate</artifactId>
  <version>0.0.1-SNAPSHOT</version>

  <name>Hibernate</name>
  <!-- FIXME change it to the project's website -->
  <url>http://www.example.com</url>

  <properties>
    <maven.compiler.release>17</maven.compiler.release>
  </properties>

  <dependencies>
    <dependency>
      <groupId>junit</groupId>
      <artifactId>junit</artifactId>
      <version>4.11</version>
      <scope>test</scope>
    </dependency>
  </dependencies>
```

Vamos a sobrescribir la etiqueta <properties> con el siguiente código:

```
<properties>
  <maven.compiler.release>17</maven.compiler.release>
</properties>
```

```

</plugin>
<!-- default lifecycle, jar packaging: see https://maven.apache.org
<plugin>
  <artifactId>maven-resources-plugin</artifactId>
  <version>3.0.2</version>
</plugin>
<plugin>
  <artifactId>maven-compiler-plugin</artifactId>
  <version>3.8.0</version>
  <configuration>
    <release>17</release>
  </configuration>
</plugin>
<plugin>
  <artifactId>maven-surefire-plugin</artifactId>
  <version>2.22.1</version>
</plugin>
<plugin>
  <artifactId>maven-jar-plugin</artifactId>
  <version>3.0.2</version>
</plugin>
<plugin>
  <artifactId>maven-install-plugin</artifactId>
  <version>2.5.2</version>

```

A continuación buscamos el plugin 'maven-compiler.plugin' y lo sobreescribimos con el siguiente código:

```

<plugin>
  <artifactId>maven-compiler-plugin</artifactId>
  <version>3.8.0</version>
  <configuration>
    <release>17</release>
  </configuration>
</plugin>

```



MySQL Connector/J » 8.0.27

JDBC Type 4 driver for MySQL

License	GPL 2.0
Categories	MySQL Drivers
Organization	Oracle Corporation
HomePage	http://dev.mysql.com/doc/connector-j/en/
Date	(Oct 18, 2021)
Files	pom (2 KB) jar (2.4 MB) View All
Repositories	Central
Used By	5,718 artifacts

[Maven](#) [Gradle](#) [Gradle \(Short\)](#) [Gradle \(Kotlin\)](#) [SBT](#) [Ivy](#) [Grape](#) [Leiningen](#) [Bu](#)

```
<!-- https://mvnrepository.com/artifact/mysql/mysql-connector-java -->
<dependency>
  <groupId>mysql</groupId>
  <artifactId>mysql-connector-java</artifactId>
  <version>8.0.27</version>
</dependency>
```

Vamos a buscar las dependencia de MySQL Connector/J en la página:

<https://mvnrepository.com/>

Cogemos la última versión y copiamos el código que aparece en la parte inferior.

```
<dependencies>
  <dependency>
    <groupId>junit</groupId>
    <artifactId>junit</artifactId>
    <version>4.11</version>
    <scope>test</scope>
  </dependency>
  <!-- https://mvnrepository.com/artifact/mysql/mysql-connector-java -->
  <dependency>
    <groupId>mysql</groupId>
    <artifactId>mysql-connector-java</artifactId>
    <version>8.0.27</version>
  </dependency>
</dependencies>
```

Pegamos la dependencia que hemos cogido dentro de la etiqueta '`<dependencies>`'.



Hibernate Core Relocation » 5.6.1.Final

Hibernate's core ORM functionality

License	LGPL 2.1
Categories	Object/Relational Mapping
Organization	Hibernate.org
HomePage	https://hibernate.org/orm
Date	(Oct 27, 2021)
Files	pom (5 KB) jar (7.1 MB) View All
Repositories	Central
Used By	3,590 artifacts

Note: There is a new version for this artifact

New Version

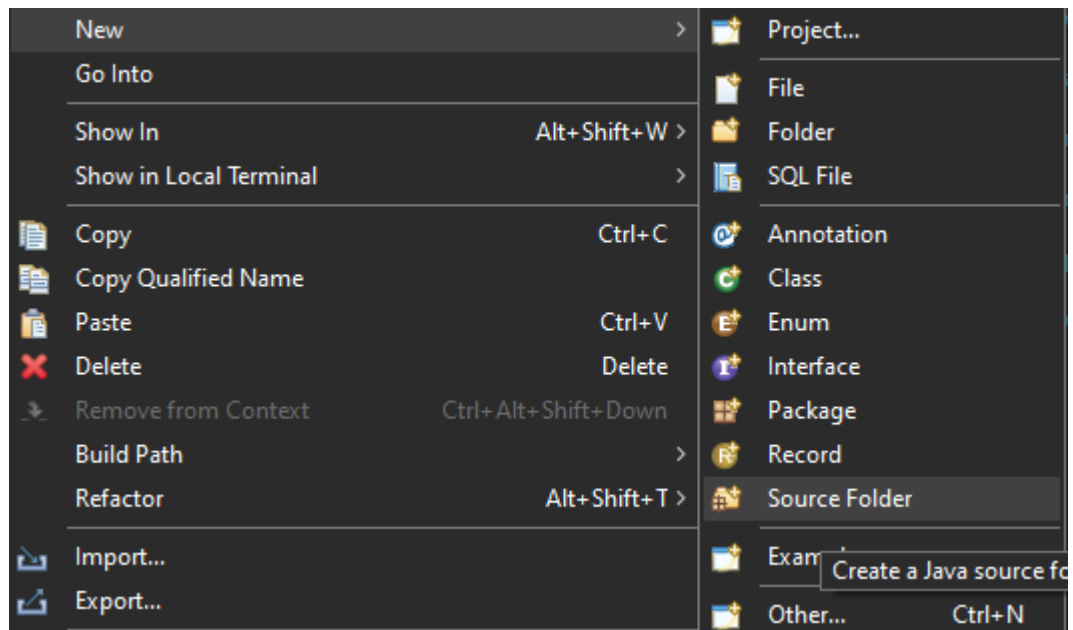
Maven Gradle Gradle (Short) Gradle (Kotlin) SBT Ivy Grape Leiningen

```
<!-- https://mvnrepository.com/artifact/org.hibernate/hibernate-core -->
<dependency>
  <groupId>org.hibernate</groupId>
  <artifactId>hibernate-core</artifactId>
  <version>5.6.1.Final</version>
</dependency>
```

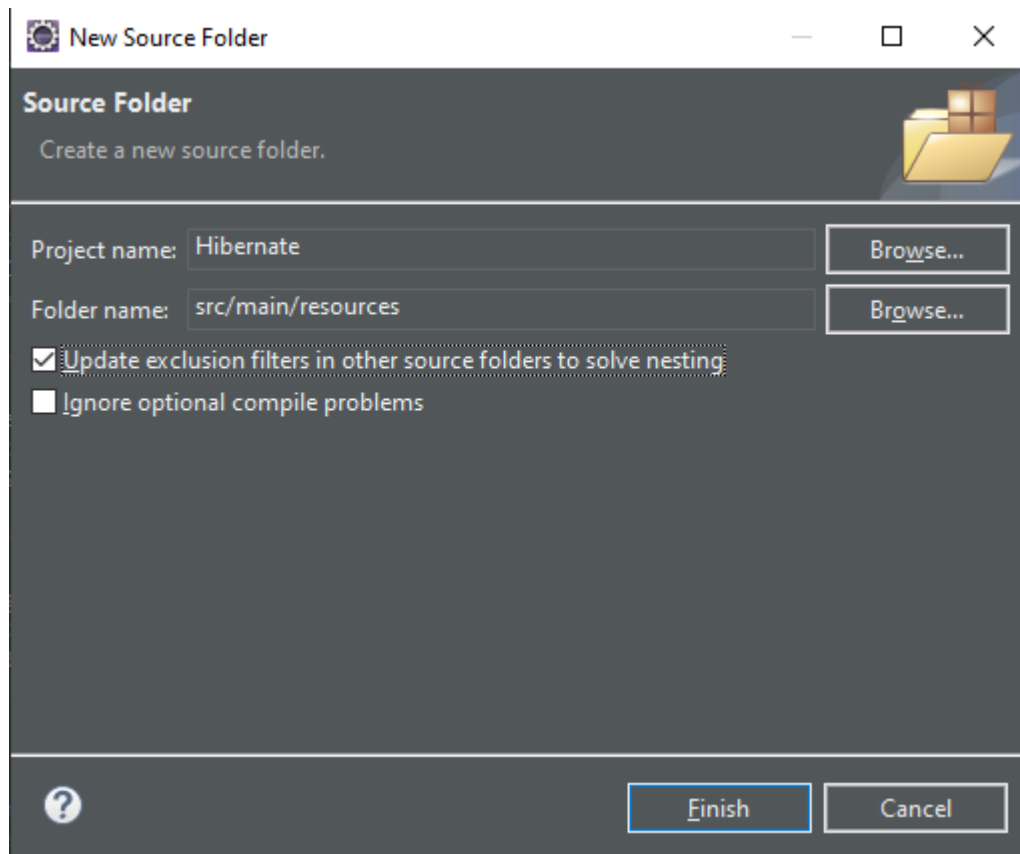
Ahora vamos a buscar la dependencia de Hibernate Core Relocation. Buscamos la última versión y copiamos la dependencia de maven como hemos hecho anteriormente.

```
<dependencies>
  <dependency>
    <groupId>junit</groupId>
    <artifactId>junit</artifactId>
    <version>4.11</version>
    <scope>test</scope>
  </dependency>
  <!-- https://mvnrepository.com/artifact/mysql/mysql-connector-java -->
  <dependency>
    <groupId>mysql</groupId>
    <artifactId>mysql-connector-java</artifactId>
    <version>8.0.27</version>
  </dependency>
  <!-- https://mvnrepository.com/artifact/org.hibernate/hibernate-core -->
  <dependency>
    <groupId>org.hibernate</groupId>
    <artifactId>hibernate-core</artifactId>
    <version>5.6.1.Final</version>
  </dependency>
</dependencies>
```

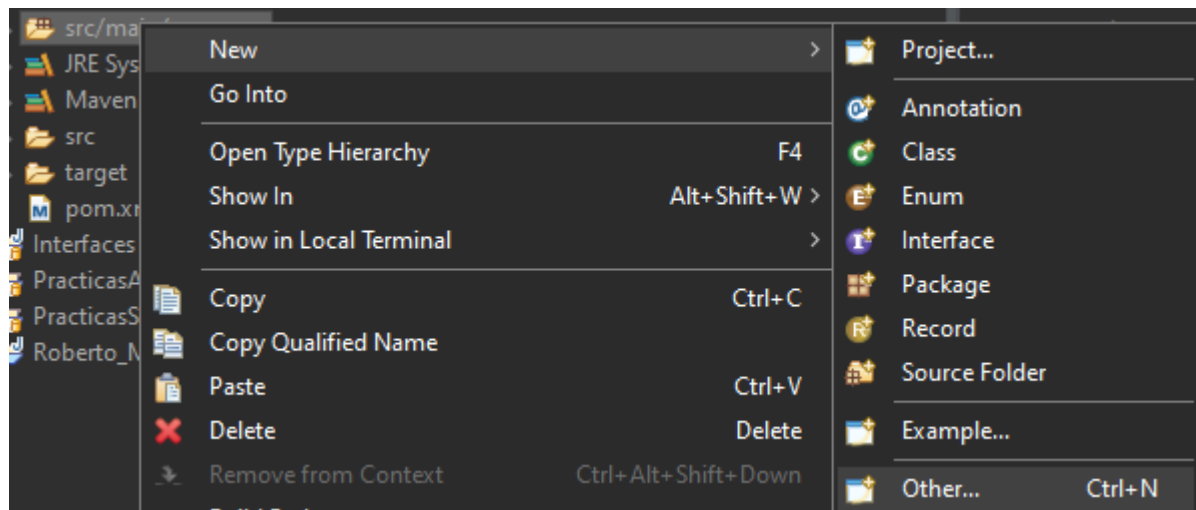
Pegamos la dependencia que hemos cogido dentro de la etiqueta '`<dependencies>`' como hemos hecho anteriormente.



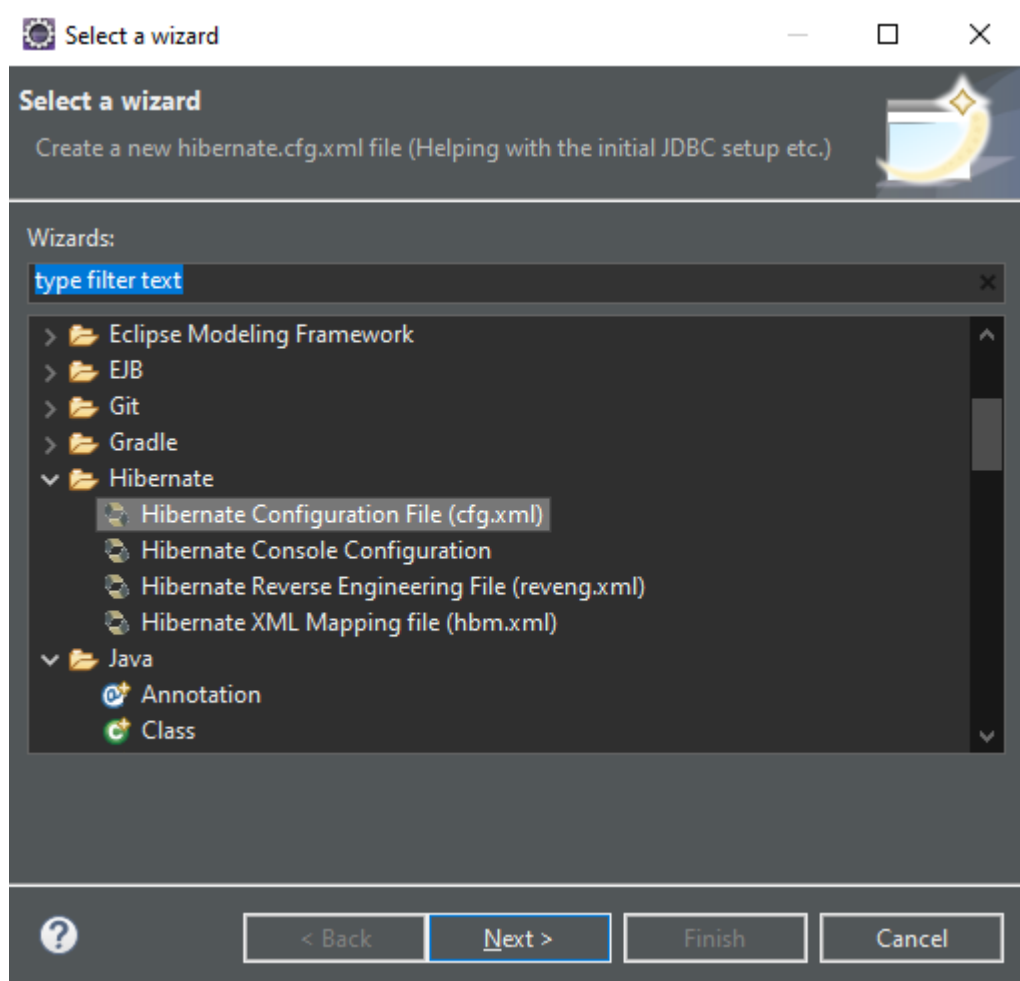
En éste punto vamos a volver al proyecto y vamos a crear pulsando con el botón derecho del ratón sobre el proyecto 'New → Source Folder' para alojar el fichero de configuración de hibernate.



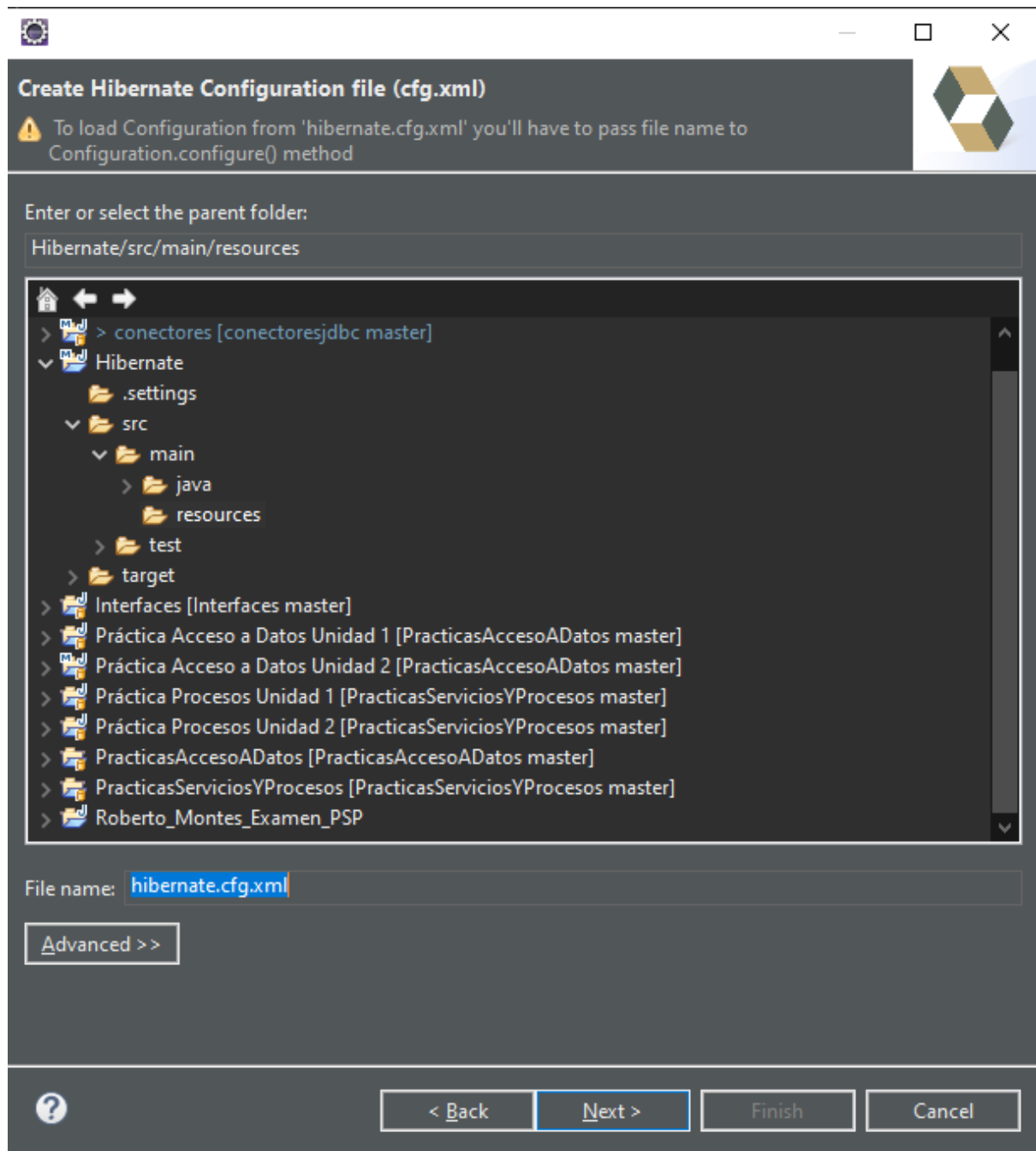
Vamos a llamarlo 'src/main/resources' y vamos a activar la opción 'Update exclusion...' como aparece en la imagen.



A continuación vamos a pinchar en la carpeta que acabamos de crear 'src/main/resources' con el botón derecho del ratón y vamos a crear un 'New → Other'.



En esta ventana vamos a pinchar en 'Hibernate → Configuration → File' y pulsamos 'Next>'



Cuando lleguemos a esta ventana dejaremos todas las opciones por defecto y pulsamos 'Next>'.

Hibernate Configuration File (cfg.xml)
This wizard creates a new configuration file to use with Hibernate.

Container:

File name:

Hibernate version:

Session factory name:

[Get values from Connection](#)

Database dialect:

Driver class:

Connection URL:

Default Schema:

Default Catalog:

Username:

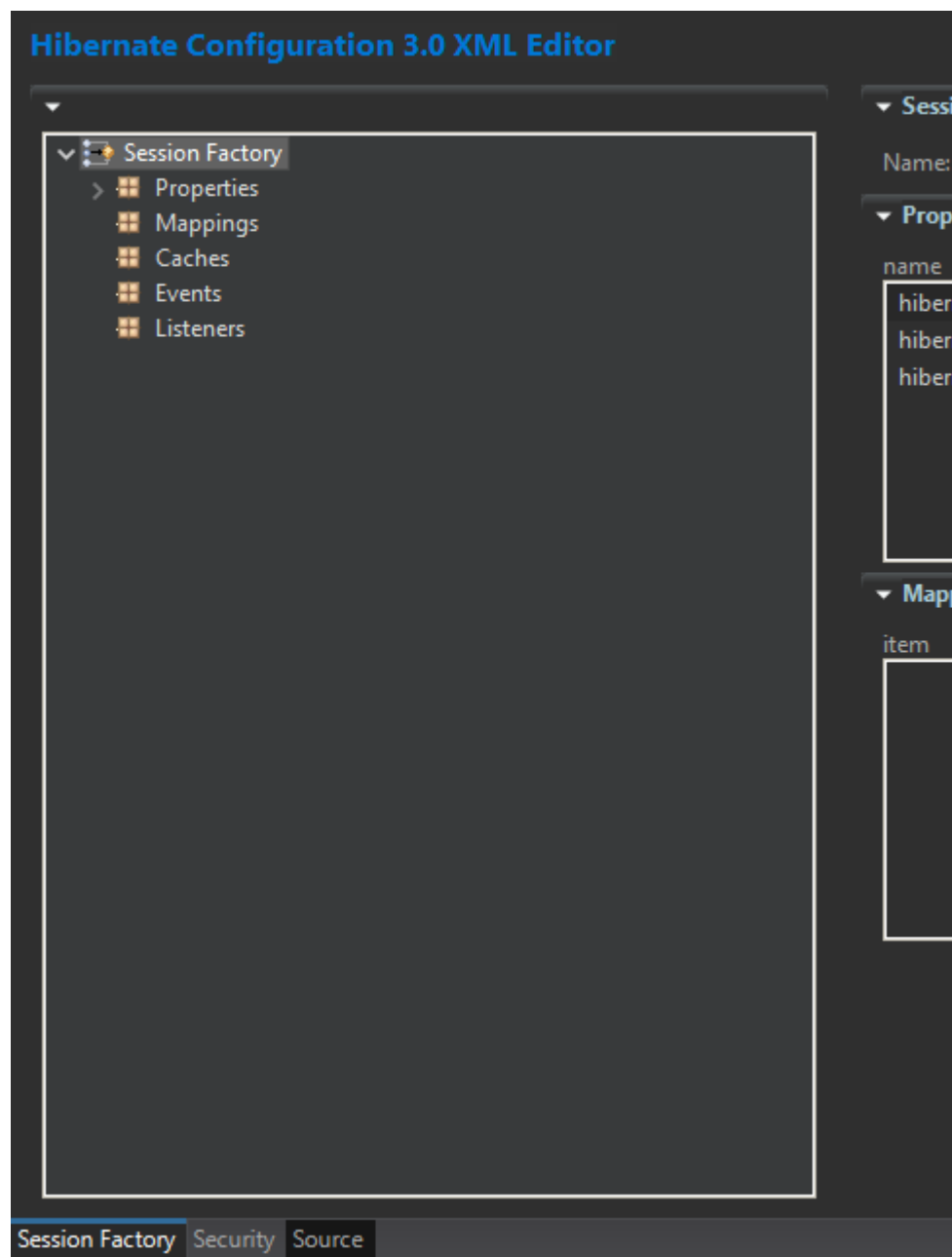
Password:

☐ Create a console configuration

En este punto únicamente tendremos que rellenar los siguientes campos:

- Database dialect : MySQL
- Driver class: com.mysql.jdbc.Driver
- Connection URL: jdbc:mysql://<hostname>/<database>

Y pulsamos 'Finish'.



Acto seguido entraremos en la configuración de Hibernate. Iremos a la parte inferior y pulsaremos sobre la pestaña 'Source'.

```

<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE hibernate-configuration PUBLIC
    "-//Hibernate/Hibernate Configuration DTD 3.0//EN"
    "http://www.hibernate.org/dtd/hibernate-configuration-3.0.dtd">
<hibernate-configuration>
  <session-factory>
    <property name="hibernate.connection.driver_class">com.mysql.jdbc.Driver</property>
    <property name="hibernate.connection.url">jdbc:mysql://&lt;hostname&gt;/&lt;database&gt;</property>
    <property name="hibernate.dialect">org.hibernate.dialect.MySQL55Dialect</property>
    <property name="hibernate.show_sql">true</property>
    <property name="hibernate.format_sql">true</property>
    <property name="hibernate.hbm2ddl.auto">create</property>
  </session-factory>
</hibernate-configuration>

```

Dentro de la etiqueta <session-factory> vamos a pegar las siguientes líneas:

```

<property name="hibernate.dialect">org.hibernate.dialect.MySQL55Dialect</property>
<property name="hibernate.show_sql">true</property>
<property name="hibernate.format_sql">true</property>
<property name="hibernate.hbm2ddl.auto">create</property>

```

En caso de que la primera propiedad (org.hibernate.dialect.MySQL55Dialect) esté especificada, habrá que sobrescribirla.

4- Creación de clases para convertirlas en tablas.

```
package hibernate.conexion;

import javax.persistence.Column;
import javax.persistence.Entity;
import javax.persistence.Id;

@Entity
public class User {

    @Id
    private int id;

    @Column
    private String userName;

    @Column
    private String userMessage;

    public User() {
        super();
    }

    public int getId() {
        return id;
    }

    public void setId(int id) {
        this.id = id;
    }

    public String getUserName() {
        return userName;
    }

    public void setUserName(String userName) {
        this.userName = userName;
    }

    public String getUserMessage() {
        return userMessage;
    }

    public void setUserMessage(String userMessage) {
        this.userMessage = userMessage;
    }

}
```

Creamos una clase que convertiremos en una tabla. Ponemos las anotaciones '@Entity, @Id y @Column' para indicar cuales son las columnas y las claves.


```
package hibernate.conexion;

import org.hibernate.cfg.Configuration;

public class HibernateUtil {

    private static final SessionFactory sessionFactory;

    static {
        try {

            sessionFactory = new Configuration().configure().buildSessionFactory();

        } catch (Throwable ex) {

            System.err.println("Initial SessionFactory creation failed." + ex);
            throw new ExceptionInInitializerError(ex);

        }
    }

    public static SessionFactory getSessionFactory() {
        return sessionFactory;
    }
}
```

Creamos una clase para la conexión con la base de datos.

```

package hibernate.conexion;

import org.hibernate.Session;

public class ORM_conexion {

    public static void main(String[] args) {

        try (Session s = HibernateUtil.getSessionFactory().openSession()) {

            Transaction t = null;

            try {

                t = s.beginTransaction();

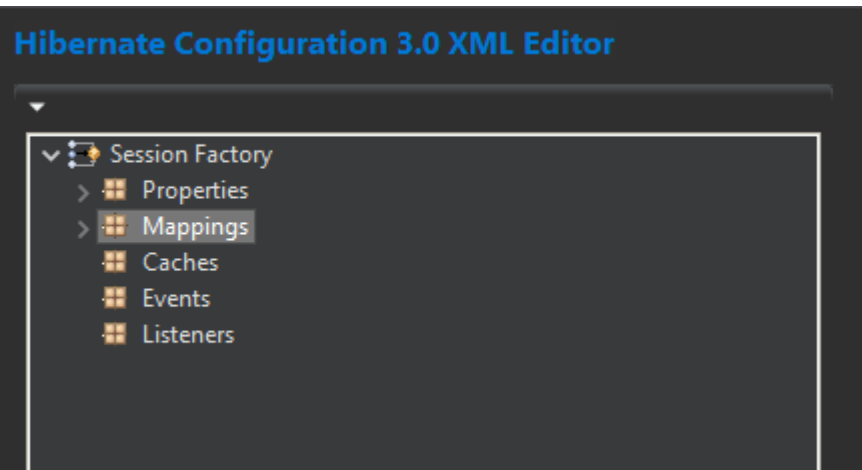
                User user1 = new User();
                user1.setId(1);
                user1.setUserName("Paco");
                user1.setUserMessage("Hello world from Pepe");
                s.save(user1);

                t.commit();

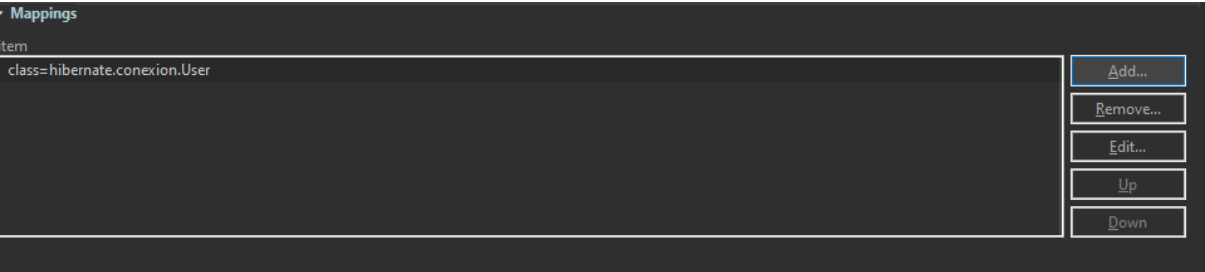
            } catch (Exception e) {
                e.printStackTrace(System.err);
                if (t != null) {
                    t.rollback();
                }
            }
        }
    }
}

```

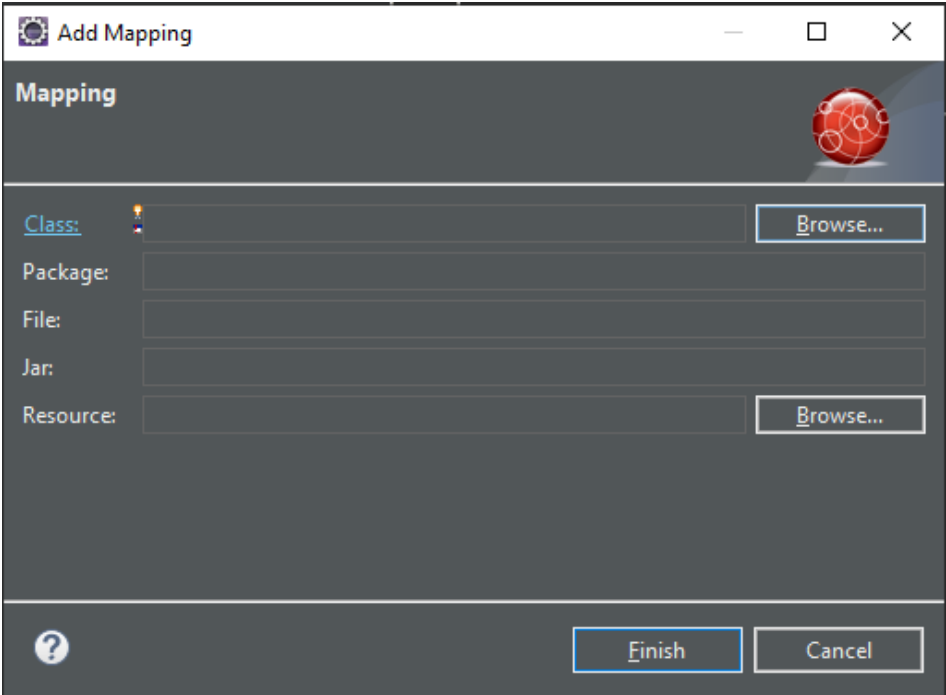
Creamos la clase principal donde introduciremos los datos en la tabla.



En la configuración de hibernate vamos a la pestaña 'Mappings'.



Vamos a la pestaña 'Add...'.



Añadimos la clase que se convertirá en tabla.

```

1 <?xml version="1.0" encoding="UTF-8"?>
2 <!DOCTYPE hibernate-configuration PUBLIC "-//Hibernate/Hibernate Configuration DTD 3.0//EN"
3 "http://www.hibernate.org/dtd/hibernate-configuration-3.0.dtd">
4 <hibernate-configuration>
5 <session-factory>
6 <property name="hibernate.connection.driver_class">com.mysql.jdbc.Driver</property>
7 <property name="hibernate.connection.url">jdbc:mysql://localhost:3306/hibernate</property>
8 <property name="hibernate.dialect">org.hibernate.dialect.MySQL55Dialect</property>
9 <property name="hibernate.show_sql">true</property>
10 <property name="hibernate.format_sql">true</property>
11 <property name="hibernate.hbm2ddl.auto">create</property>
12 <property name="hibernate.connection.username">root</property>
13 <property name="hibernate.connection.password">root</property>
14 <property name="hibernate.generate_statistics">false</property>
15 <mapping class="hibernate.conexion.User"/>
16 </session-factory>
17 </hibernate-configuration>
18

```

Volvemos a la pestaña 'Source' de hibernate y vemos que aparece el atributo '`name=""`' en la etiqueta de 'session-factory'. Procedemos a borrarlo.

En este momento ya tenemos configurado nuestro hibernate y ya podemos funcionar con él.

```

dic 01, 2021 8:52:58 A. M. org.hibernate.Version logVersion
INFO: HHH0000412: Hibernate ORM core version 5.6.1.Final
dic 01, 2021 8:52:59 A. M. org.hibernate.annotations.common.reflection.java.JavaReflectionManager <clinit>
INFO: HCN00000001: Hibernate Commons Annotations (5.1.2.Final)
dic 01, 2021 8:52:59 A. M. org.hibernate.engine.jdbc.connections.internal.DriverManagerConnectionProviderImpl configure
WARN: HHH10001002: Using Hibernate built-in connection pool (not for production use!)
Loading class 'com.mysql.jdbc.Driver'. This is deprecated. The new driver class is 'com.mysql.cj.jdbc.Driver'. The driver is automatically registered via the SPI and manual loading of the driver
class is generally unnecessary.
dic 01, 2021 8:52:59 A. M. org.hibernate.engine.jdbc.connections.internal.DriverManagerConnectionProviderImpl buildCreator
INFO: HHH10001005: using driver [com.mysql.jdbc.Driver] at URL [jdbc:mysql://localhost:3306/hibernate]
dic 01, 2021 8:52:59 A. M. org.hibernate.engine.jdbc.connections.internal.DriverManagerConnectionProviderImpl buildCreator
INFO: HHH10001001: Connection properties: {password=****, user=root}
dic 01, 2021 8:52:59 A. M. org.hibernate.engine.jdbc.connections.internal.DriverManagerConnectionProviderImpl buildCreator
INFO: HHH10001003: Autocommit mode: false
dic 01, 2021 8:52:59 A. M. org.hibernate.engine.jdbc.connections.internal.DriverManagerConnectionProviderImpl$PooledConnections <init>
INFO: HHH000115: Hibernate connection pool size: 20 (min=1)
dic 01, 2021 8:52:59 A. M. org.hibernate.dialect.Dialect <init>
INFO: HHH000400: Using dialect: org.hibernate.dialect.MySQL55Dialect
Hibernate:
    drop table if exists User
dic 01, 2021 8:52:59 A. M. org.hibernate.resource.transaction.backend.jdbc.internal.DdlTransactionIsolatorNonJtaImpl getIsolatedConnection
INFO: HHH10001501: Connection obtained from JdbcConnectionAccess [org.hibernate.engine.jdbc.env.internal.JdbcEnvironmentInitiator$ConnectionProviderJdbcConnectionAccess@6f76c2cc] for (non-JTA) DDL
execution was not in auto-commit mode; the Connection 'local transaction' will be committed and the Connection will be set into auto-commit mode.
Hibernate:
    create table User (
      id integer not null,
      userMessage varchar(255),
      userName varchar(255),
      primary key (id)
    ) engine=InnoDB
dic 01, 2021 8:52:59 A. M. org.hibernate.resource.transaction.backend.jdbc.internal.DdlTransactionIsolatorNonJtaImpl getIsolatedConnection
INFO: HHH10001501: Connection obtained from JdbcConnectionAccess [org.hibernate.engine.jdbc.env.internal.JdbcEnvironmentInitiator$ConnectionProviderJdbcConnectionAccess@1b7332a7] for (non-JTA) DDL
execution was not in auto-commit mode; the Connection 'local transaction' will be committed and the Connection will be set into auto-commit mode.
dic 01, 2021 8:52:59 A. M. org.hibernate.engine.transaction.jta.platform.internal.JtaPlatformInitiator initiateService
INFO: HHH000490: Using JtaPlatform implementation: [org.hibernate.engine.transaction.jta.platform.internal.NoJtaPlatform]
Hibernate:
    insert
    into
      User
      (userMessage, userName, id)
    values
      (?, ?, ?)

```

(Imagen de la salida)