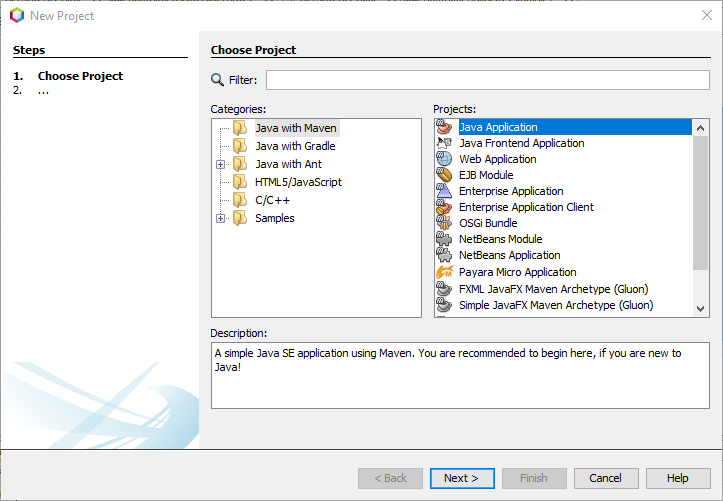
**PROYECTO BÁSICO PARA LA UD2 ORM CON HIBERNATE**

En este documento se explica como configurar un proyecto mínimo para empezar a trabajar con la unidad 3, añadiendo todas las dependencias necesarias para su buen funcionamiento.

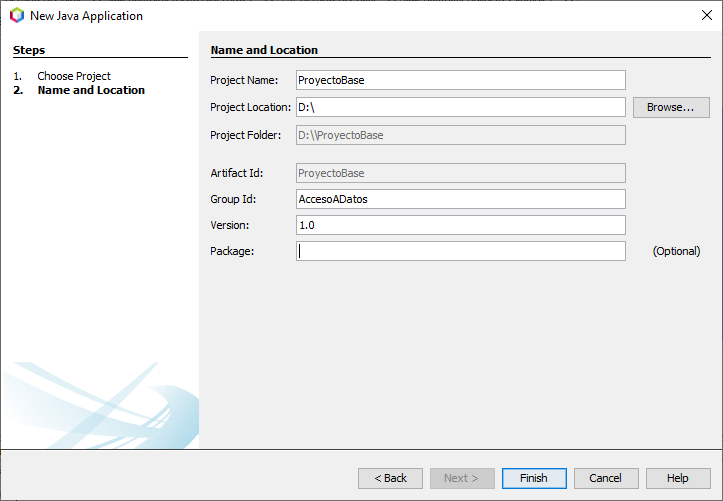
Los pasos a realizar son los siguientes:

1) Creación del proyecto

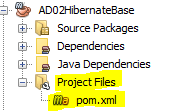
Pulsamos en File-> New Project y seleccionamos un proyecto con el gestor de proyectos Maven.



Indicamos el nombre del proyecto, la ubicación, el identificador de grupo y la versión del proyecto (estos dos últimos no tienen demasiada relevancia)



2) Añadir las dependencias necesarias en el archivo pom.xml. Dicho archivo se encuentra en la carpeta del proyecto con nombre "Project Files"



En dicho archivo añadimos las siguientes dependencias (a continuación de la etiqueta </version>:

<dependencies>

<dependency>

<groupId>org.hibernate</groupId>

<artifactId>hibernate-core-jakarta</artifactId>

<version>5.6.12.Final</version>

<type>pom.sha512</type>

</dependency>

<dependency>

<groupId>org.hibernate</groupId>

<artifactId>hibernate-testing-jakarta</artifactId>

<version>5.6.12.Final</version>

</dependency>

<dependency>

<groupId>mysql</groupId>

<artifactId>mysql-connector-java</artifactId>

<version>8.0.30</version>

</dependency>

<dependency>

<groupId>org.slf4j</groupId>

<artifactId>slf4j-log4j12</artifactId>

<version>1.7.25</version>

</dependency>

</dependencies>

3) Creamos el fichero de configuración persistence.xml

Para ello, pulsamos con el botón derecho en el proyecto -> New-> Other->Categoría XML-> XML Document

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

Indicamos el nombre del fichero poniendo *persistence.xml* e indicamos en *Folder* la ruta *src\main\resources\META-INF* (ojo indicarlo correctamente, ya que Maven busca el fichero de configuración en este lugar). En resto de opciones no las modificamos, avanzamos y pulsamos *Finish*

Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

Descripción generada automáticamente

Una vez creado, limpiamos su contenido y añadimos las siguientes propiedades dentro de él.

<persistence 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"

version="2.1">

<persistence-unit name="templatePU" transaction-type="RESOURCE\_LOCAL">

<exclude-unlisted-classes>false</exclude-unlisted-classes>

<properties>

<property name="jakarta.persistence.jdbc.driver" value="com.mysql.cj.jdbc.Driver"/>

<property name="jakarta.persistence.jdbc.url" value="jdbc:mysql://192.168.1.83:3306/mysql?zeroDateTimeBehavior=CONVERT\_TO\_NULL"/>

<property name="jakarta.persistence.jdbc.user" value="usuario"/>

<property name="jakarta.persistence.jdbc.password" value="adad"/>

<property name="hibernate.dialect" value="org.hibernate.dialect.MySQLDialect"/>

<property name="hibernate.show\_sql" value="true"/>

</properties>

</persistence-unit>

</persistence>

Evidentemente, habrá que adaptar la URL, usuario, contraseña, driver …

3) Testeo del proyecto

Para probar que todo se ha configurado de forma adecuada, puedes crearte una clase con el código que a continuación se detalla (se ha creado un paquete llamado test previamente).

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

El código que aparece a continuación crea los objetos necesarios: factoria, manager … una transacción y cierra todo ello.

package test;

import jakarta.persistence.EntityManager;

import jakarta.persistence.EntityManagerFactory;

import jakarta.persistence.Persistence;

public class Run {

public static void main(String[] args) {

EntityManagerFactory entityManagerFactory = Persistence.createEntityManagerFactory("templatePU");

EntityManager entityManager = entityManagerFactory.createEntityManager();

entityManager.getTransaction().begin();

System.out.println("TESTEO DEL PROYECTO BASE CORRECTO!!");

entityManager.getTransaction().commit();

entityManager.close();

entityManagerFactory.close();

}

}

CREACIÓN DE UN NUEVO PROYECTO