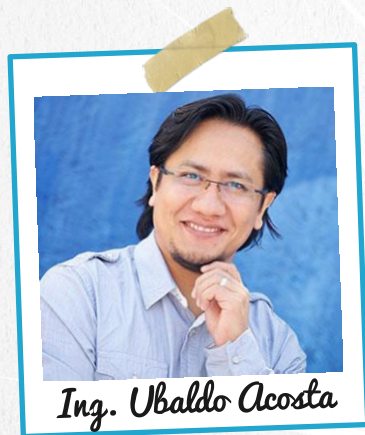


CURSO HIBERNATE Y JPA

HOLA MUNDO CON HIBERNATE/JPA



Por el experto: Ing. Ubaldo Acosta



Experiencia y Conocimiento para tu vida



CURSO HIBERNATE Y JPA

www.globalmentoring.com.mx

OBJETIVO DEL EJERCICIO

Crear una aplicación para hacer un HolaMundo con Hibernate/JPA.

Output - Run (HolaMundoHibernate) x

```
15:42:46 [main] INFO org.hibernate.jpa.internal.util.LogHelper - HHH000204: Processing PersistenceUnitInfo [
    name: HibernateEjemplol
    ...]
15:42:46 [main] INFO org.hibernate.Version - HHH000412: Hibernate Core (5.3.0.CR2)
15:42:46 [main] INFO org.hibernate.cfg.Environment - HHH000206: hibernate.properties not found
15:42:46 [main] INFO org.hibernate.annotations.common.Version - HCANN000001: Hibernate Commons Annotations (5.0.3.Final)
15:42:47 [main] WARN org.hibernate.orm.connections.pooling - HHH10001002: Using Hibernate built-in connection pool (not for production use!)
15:42:47 [main] INFO org.hibernate.orm.connections.pooling - HHH10001005: using driver [com.mysql.jdbc.Driver] at URL [jdbc:mysql://localhost:3306/sga?useSSL=true]
15:42:47 [main] INFO org.hibernate.orm.connections.pooling - HHH10001001: Connection properties: {user=root, password=****}
15:42:47 [main] INFO org.hibernate.orm.connections.pooling - HHH10001003: Autocommit mode: false
15:42:47 [main] INFO org.hibernate.engine.jdbc.connections.internal.DriverManagerConnectionProviderImpl - HHH000115: Hibernate connection pool size: 20 (min=1)
15:42:47 [main] INFO org.hibernate.dialect.Dialect - HHH000400: Using dialect: org.hibernate.dialect.MySQL57Dialect
15:42:48 [main] WARN org.hibernate.type.spi.TypeConfiguration$Scope - HHH000233: Scoping types to session factory org.hibernate.internal.SessionFactoryImpl@19fb8826 after already
15:42:48 [main] INFO org.hibernate.hql.internal.QueryTranslatorFactoryInitiator - HHH000397: Using ASTQueryTranslatorFactory
15:42:49 [main] DEBUG org.hibernate.SQL - select persona0_.id_persona as id_persona0_, persona0_.apellido as apellido2_0_, persona0_.nombre as nombre3_0_ from Persona persona0_
15:42:49 [main] INFO mx.com.gm.test.HolaMundoHibernate - Listado de Personas:
15:42:49 [main] INFO mx.com.gm.test.HolaMundoHibernate - persona:Persona [apellido=Martinez, idPersona=10, nombre=Juan Carlos]
```

CURSO HIBERNATE Y JPA

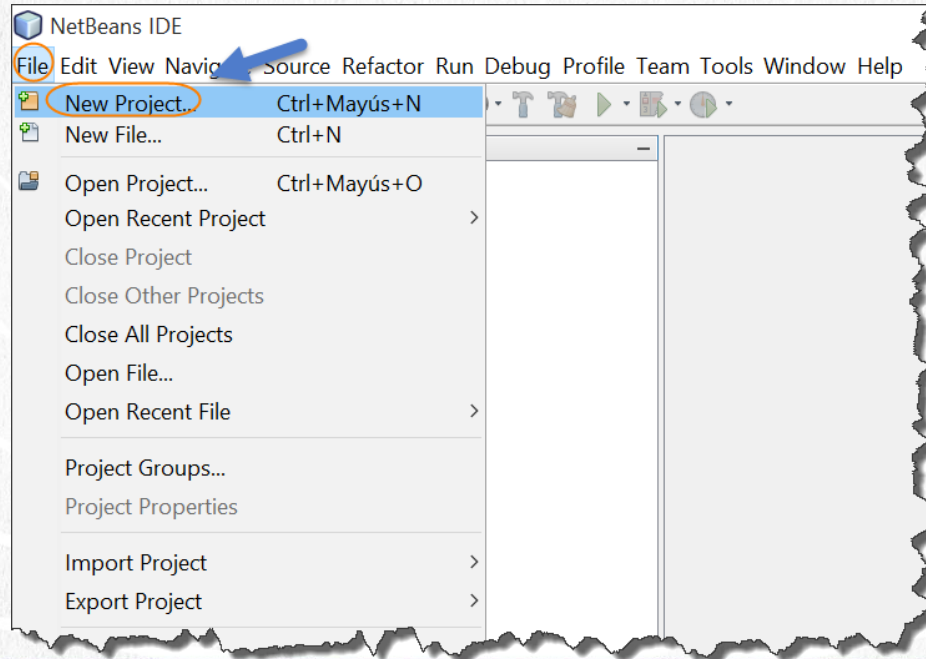
www.globalmentoring.com.mx

1. CREAR PROYECTO

- Vamos a utilizar Maven para crear nuestros proyectos del curso aplicando el framework de Hibernate/JPA.
- Maven es una tecnología que, entre otras cosas, nos permite administrar las librerías de nuestra aplicación Java, ya sea una aplicación estándar, una aplicación Web, aplicación Empresarial o cualquier tipo de aplicación Java que utilice varias librerías.
- En el caso del framework de Hibernate se utilizan muchas librerías, por lo que el uso de Maven en la creación de proyectos con Hibernate es ideal para la administración y despliegue de aplicaciones con este framework. Para más información de maven, pueden consultar el siguiente link:
<https://maven.apache.org/>

1. CREAR PROYECTO

- Creamos un nuevo proyecto Java:

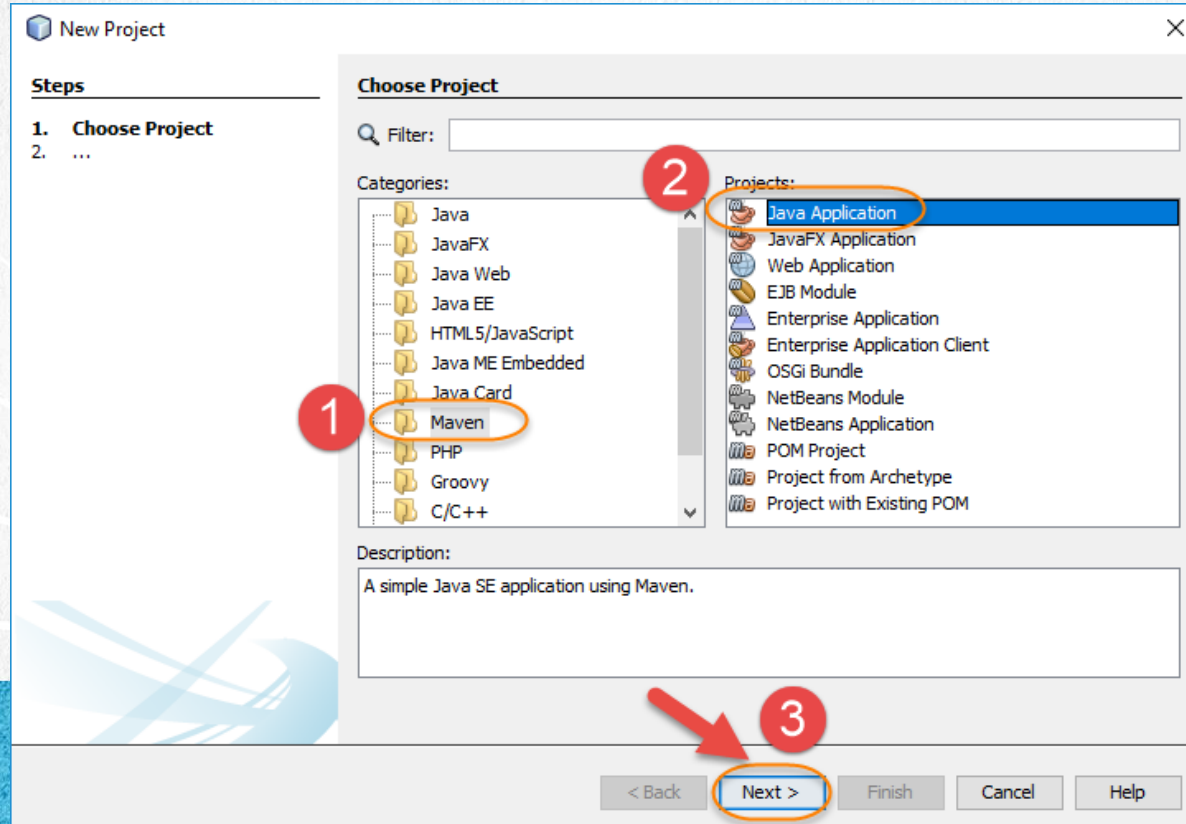


CURSO HIBERNATE Y JPA

www.globalmentoring.com.mx

1. CREAR PROYECTO

- Creamos un nuevo proyecto Java utilizando Maven:



1. CREAR PROYECTO

- Creamos un nuevo proyecto Java. Escribimos los valores según se muestra:

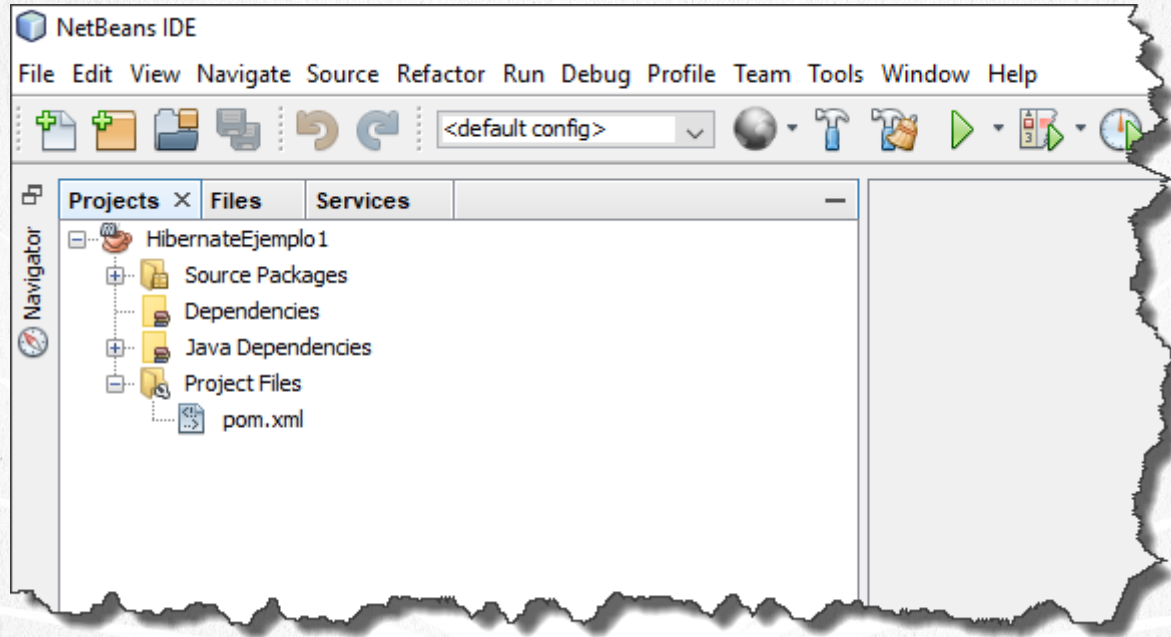
The screenshot shows the 'New Java Application' dialog box with the following fields and values:

- Project Name:** HibernateEjemplo1
- Project Location:** C:\Cursos\Hibernate\Leccion01
- Project Folder:** C:\Cursos\Hibernate\Leccion01\HibernateEjemplo1
- Artifact Id:** HibernateEjemplo1
- Group Id:** mx.com.gm
- Version:** 1.0
- Package:** (Optional)

The 'Finish' button is highlighted with a red arrow and the number 5.

1. CREAR PROYECTO

- Este es el resultado de la creación del proyecto Java con Maven:

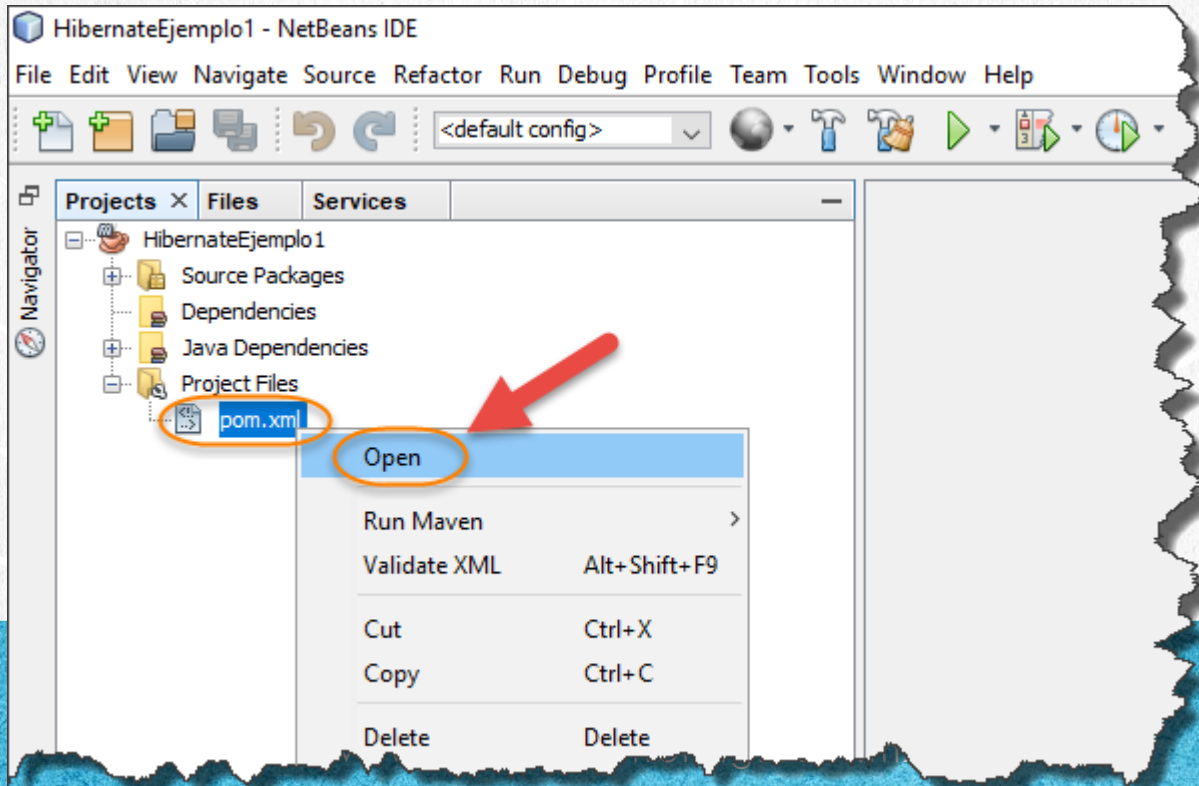


CURSO HIBERNATE Y JPA

www.globalmentoring.com.mx

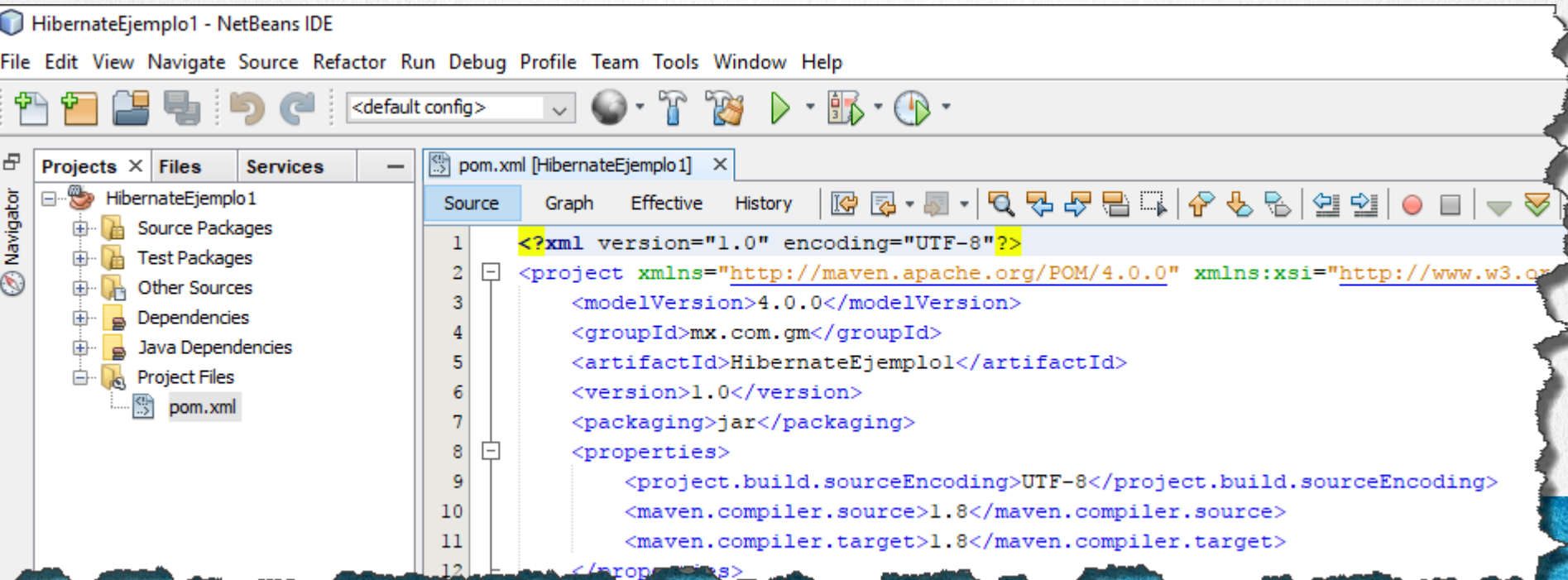
2. ABRIMOS EL ARCHIVO POM.XML DE MAVEN

- El archivo pom.xml de maven administra las librerías Java vamos a utilizar. Abrimos el archivo pom.xml para modificarlo con el código siguiente:



2. ABRIMOS EL ARCHIVO POM.XML DE MAVEN

- Una vez abierto, vamos a modificar la información por completo de este archivo, con la información proporcionada a continuación:



PASO 3. MODIFICAMOS EL CÓDIGO

Archivo pom.xml:

Clic para ver el
archivo

```
<?xml version="1.0" encoding="UTF-8"?>
<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>mx.com.gm</groupId>
  <artifactId>HibernateEjemplol</artifactId>
  <version>1.0</version>
  <packaging>jar</packaging>
  <properties>
    <project.build.sourceEncoding>UTF-8</project.build.sourceEncoding>
    <maven.compiler.source>1.8</maven.compiler.source>
    <maven.compiler.target>1.8</maven.compiler.target>
  </properties>
  <dependencies>
    <dependency>
      <groupId>org.hibernate</groupId>
      <artifactId>hibernate-core</artifactId>
      <version>5.3.0.CR2</version>
    </dependency>
    <dependency>
      <groupId>mysql</groupId>
      <artifactId>mysql-connector-java</artifactId>
      <version>5.1.46</version>
    </dependency>
  </dependencies>
</project>
```


PASO 3. MODIFICAMOS EL CÓDIGO

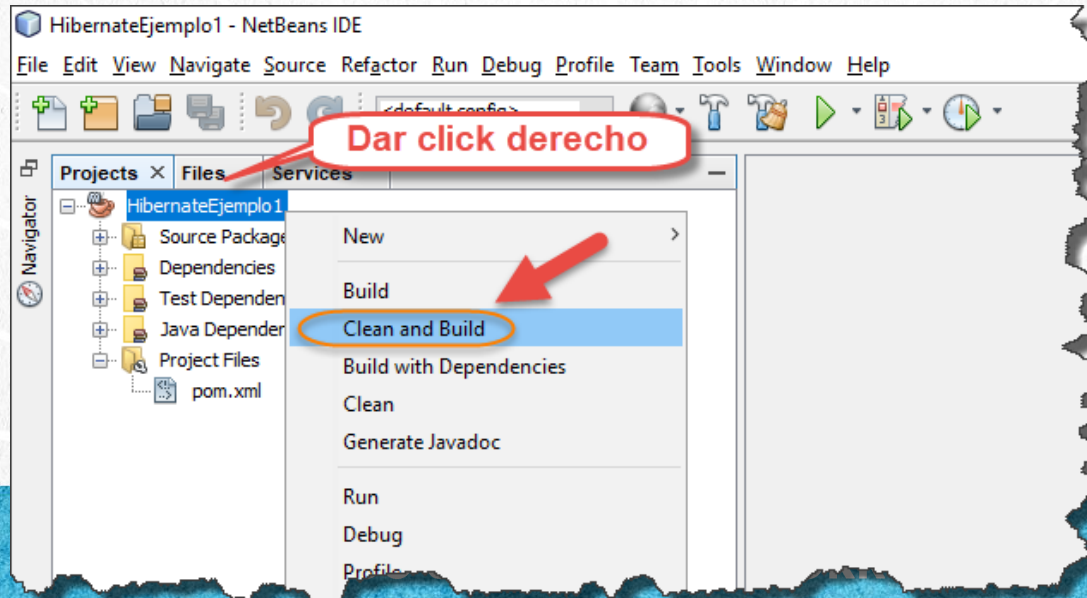
Archivo pom.xml:

Clic para ver el
archivo

```
<dependency>
  <groupId>org.apache.logging.log4j</groupId>
  <artifactId>log4j-api</artifactId>
  <version>2.10.0</version>
</dependency>
<dependency>
  <groupId>org.apache.logging.log4j</groupId>
  <artifactId>log4j-core</artifactId>
  <version>2.10.0</version>
</dependency>
</dependencies>
</project>
```

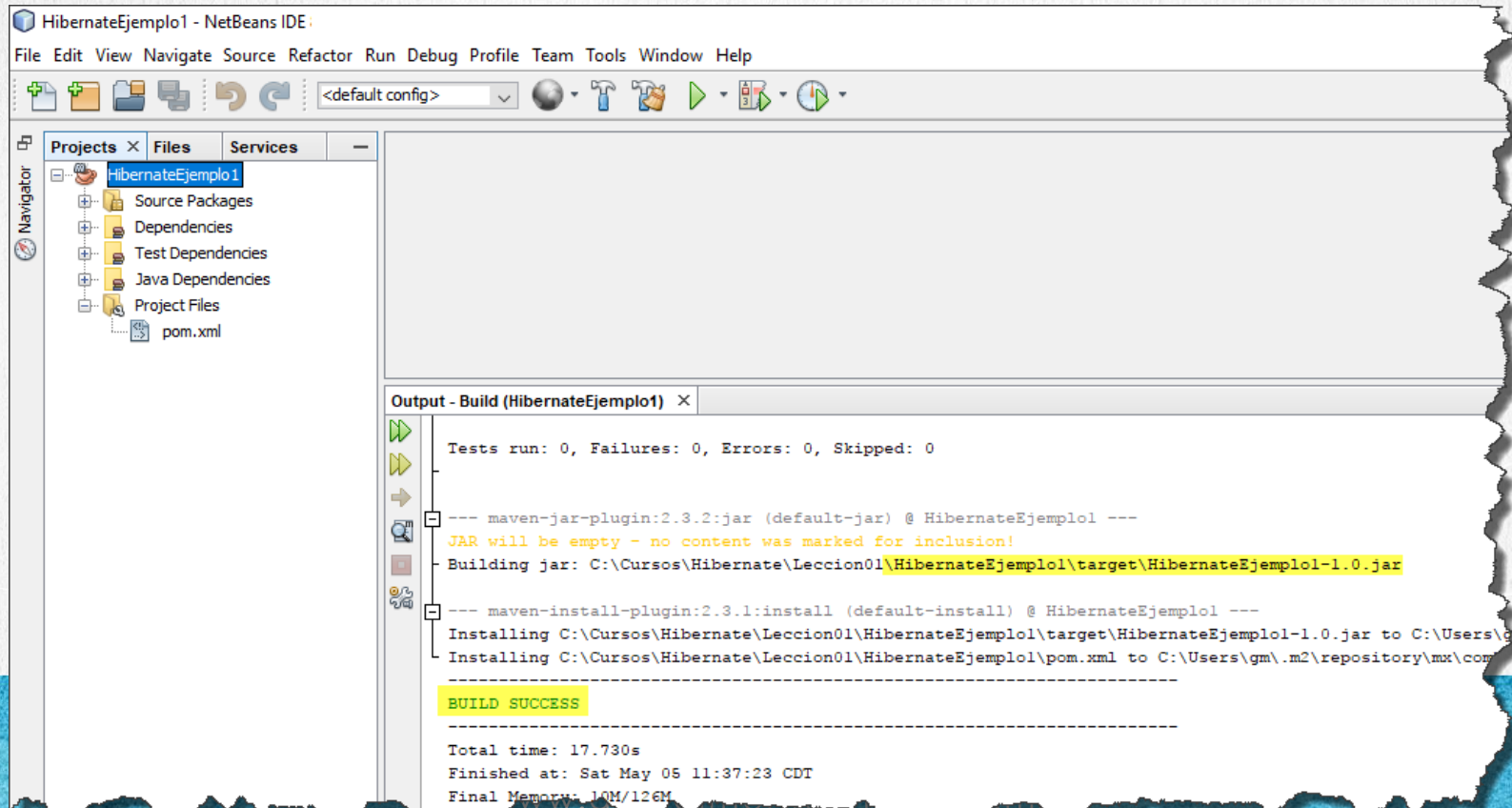
4. HACEMOS CLEAN & BUILD

- Para que se descarguen las librerías, hacemos Clean & Build al proyecto. Si por alguna razón este proceso falla, se debe desactivar cualquier software como antivirus, Windows defender o firewall durante este proceso para que no se impida la descarga de archivos .jar de Java. Una vez terminado se pueden volver a activar estos servicios. Este proceso puede demorar varios minutos dependiendo de su velocidad de internet:



4. HACEMOS CLEAN & BUILD

- Una vez terminado el proceso se debe mostrar una salida similar a la siguiente:



The screenshot shows the NetBeans IDE interface for a project named 'HibernateEjemplo1'. The 'Output - Build (HibernateEjemplo1)' window is open, displaying the following build log:

```
Tests run: 0, Failures: 0, Errors: 0, Skipped: 0

--- maven-jar-plugin:2.3.2:jar (default-jar) @ HibernateEjemplo1 ---
JAR will be empty - no content was marked for inclusion!
Building jar: C:\Cursos\Hibernate\Leccion01\HibernateEjemplo1\target\HibernateEjemplo1-1.0.jar

--- maven-install-plugin:2.3.1:install (default-install) @ HibernateEjemplo1 ---
Installing C:\Cursos\Hibernate\Leccion01\HibernateEjemplo1\target\HibernateEjemplo1-1.0.jar to C:\Users\c...
Installing C:\Cursos\Hibernate\Leccion01\HibernateEjemplo1\pom.xml to C:\Users\gm\.m2\repository\mx\com\...

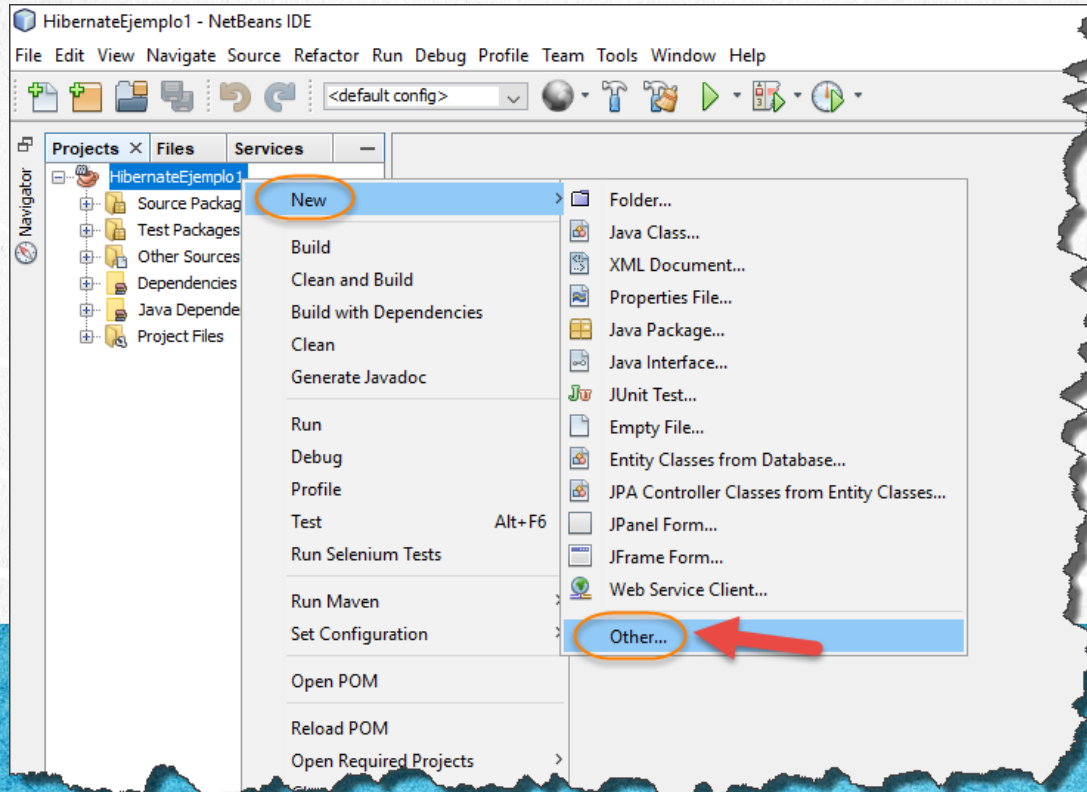
BUILD SUCCESS

-----

Total time: 17.730s
Finished at: Sat May 05 11:37:23 CDT
Final Memory: 10M/126M
```

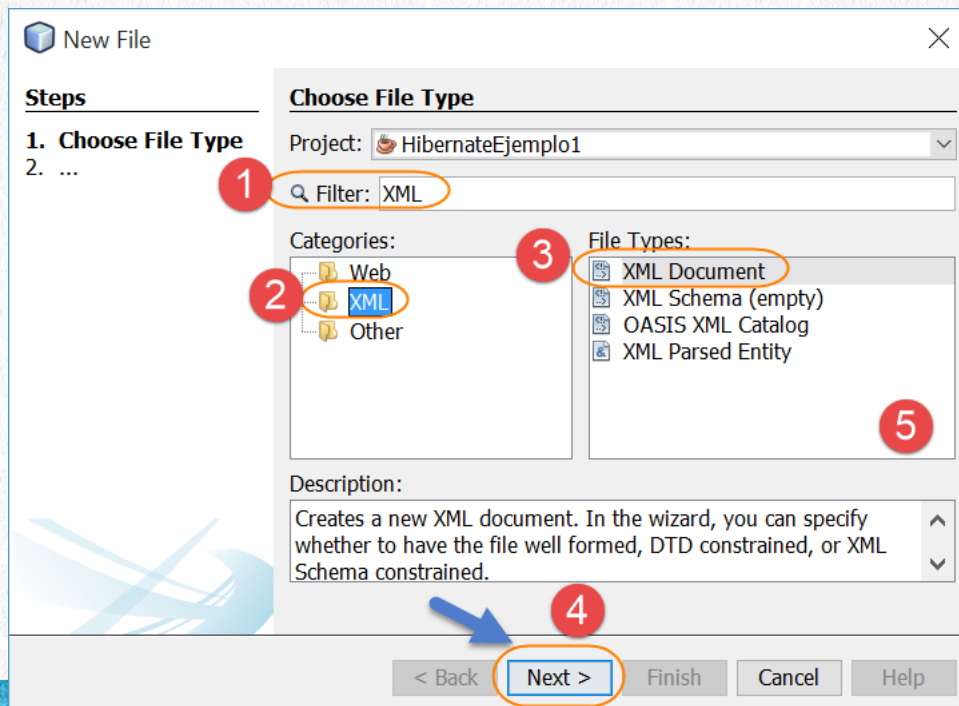
5. CREAR UN ARCHIVO XML

- Creamos el archivo persistence.xml. El archivo persistence.xml tiene la configuración de conexión a la base de datos, en este caso mysql, entre otros valores, como las clases de entidad que vamos a utilizar en el proyecto, y el proveedor del Entity Manager:



5. CREAR UN ARCHIVO XML

- Filtramos por creación de archivos xml:



CURSO HIBERNATE Y JPA

www.globalmentoring.com.mx

5. CREAR UN ARCHIVO XML

- Creamos el archivo persistence.xml. Lo depositamos en la ruta indicada:

New XML Document

Steps

1. Choose File Type
2. **Name and Location**
3. Select Document Type
4. ...

Name and Location

File Name:

Project:

Folder:

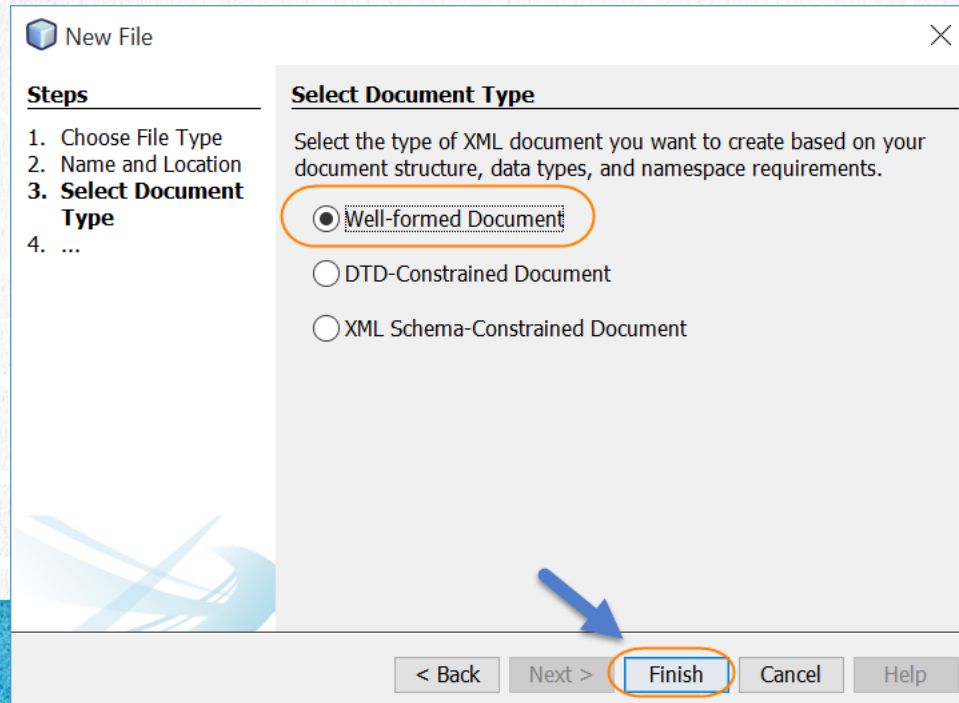
Created File:

CURSO HIBERNATE Y JPA

www.globalmentoring.com.mx

5. CREAR UN ARCHIVO XML

- En este paso seleccionamos cualquier opción, no es importante ya que vamos a sobrescribir el archivo:



PASO 6. MODIFICAMOS EL CÓDIGO

Archivo persistence.xml:

Clic para ver el
archivo

```
<?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="HibernateEjemplo1" transaction-type="RESOURCE_LOCAL">
    <provider>org.hibernate.jpa.HibernatePersistenceProvider</provider>
    <class>mx.com.gm.domain.Persona</class>
    <properties>
      <property name="javax.persistence.jdbc.url" value="jdbc:mysql://localhost:3306/sga?useSSL=true"/>
      <property name="javax.persistence.jdbc.user" value="root"/>
      <property name="javax.persistence.jdbc.password" value="admin"/>
      <property name="javax.persistence.jdbc.driver" value="com.mysql.jdbc.Driver"/>
      <property name="hibernate.show_sql" value="true" />
    </properties>
  </persistence-unit>
</persistence>
```

CURSO DE JAVA CON JDBC

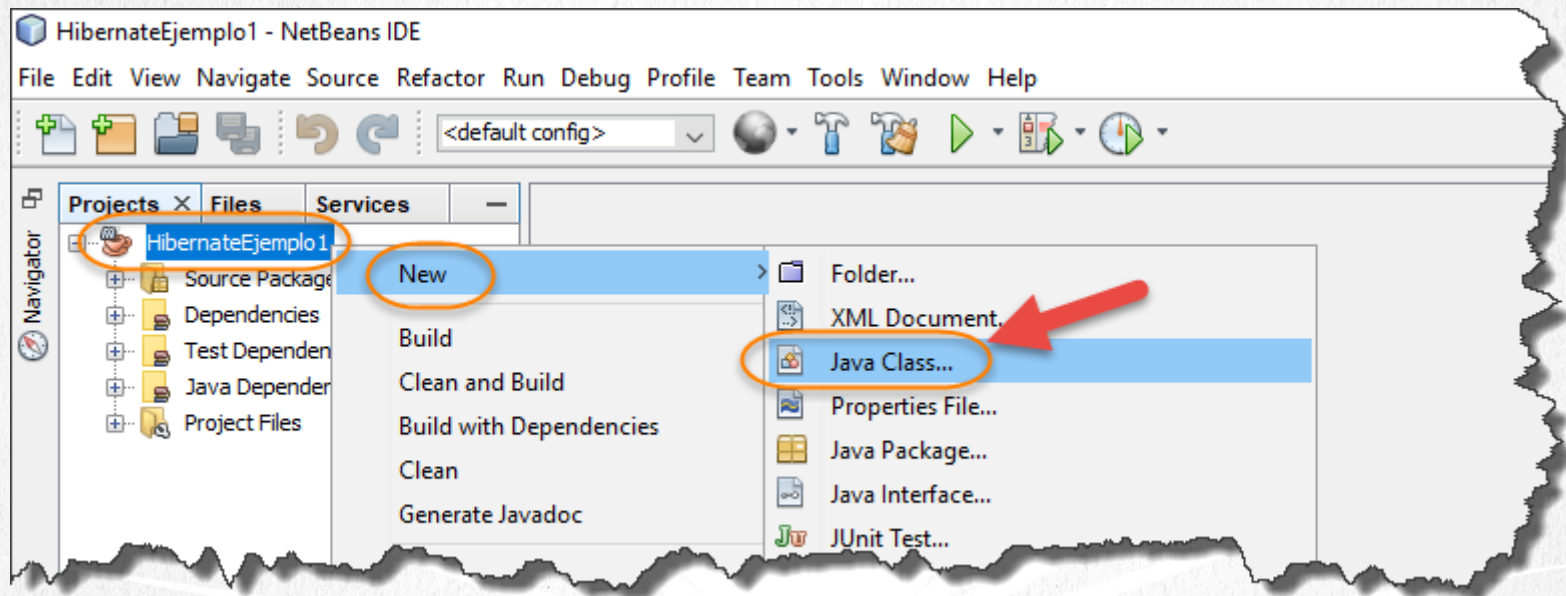
www.globalmentoring.com.mx

7. CREAR LA CLASE PERSONA

- Creamos la clase Persona.java.
- Esta es una clase de Entidad, por lo que mapea con la tabla de persona en la base de datos de “sga” en MySQL.
- Posteriormente veremos a detalle cómo crear las clases de Entidad, en este ejemplo sólo es para poder realizar la primer prueba con Hibernate.
- Veamos cómo queda nuestra clase de Entidad llamada Persona.

7. CREAR LA CLASE PERSONA

- Creamos la clase Persona.java:

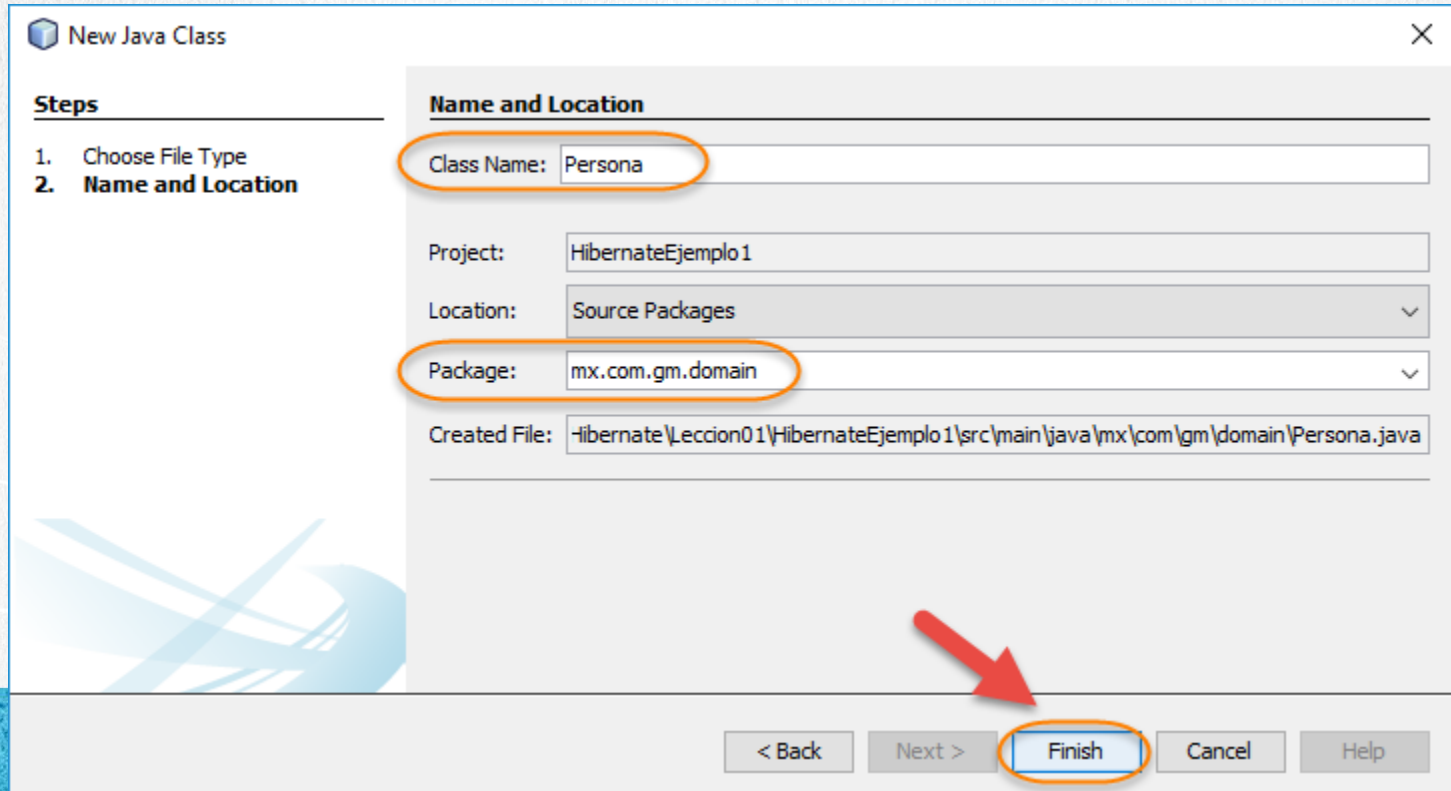


CURSO HIBERNATE Y JPA

www.globalmentoring.com.mx

7. CREAR LA CLASE PERSONA

- Creamos la clase Persona.java:



New Java Class

Steps

1. Choose File Type
2. **Name and Location**

Name and Location

Class Name:

Project:

Location:

Package:

Created File:

PASO 8. MODIFICAMOS EL CÓDIGO

Archivo Persona.java:

[Click para descargar archivo](#)

```
package mx.com.gm.domain;

import java.io.Serializable;
import javax.persistence.*;

@Entity
public class Persona implements Serializable {

    private static final long serialVersionUID = 1L;

    @Column(name = "id_persona")
    @Id
    private int idPersona;

    private String apellido;

    private String nombre;

    public Persona() {
    }

    public String getApellido() {
        return this.apellido;
    }

    public void setApellido(String apellido) {
        this.apellido = apellido;
    }
}
```

PASO 8. MODIFICAMOS EL CÓDIGO

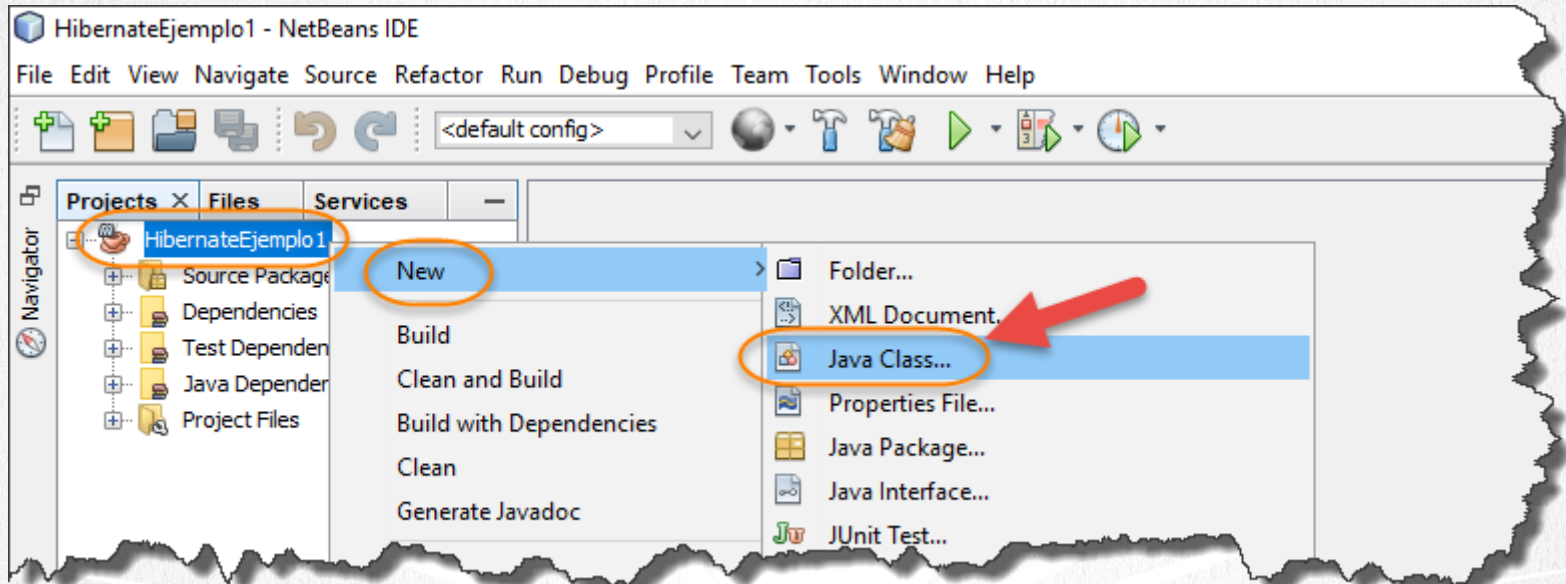
Archivo Persona.java:

Clic para ver el archivo

```
public int getIdPersona() {  
    return this.idPersona;  
}  
  
public void setIdPersona(int idPersona) {  
    this.idPersona = idPersona;  
}  
  
public String getNombre() {  
    return this.nombre;  
}  
  
public void setNombre(String nombre) {  
    this.nombre = nombre;  
}  
  
@Override  
public String toString() {  
    return "Persona [apellido=" + apellido + ", idPersona=" + idPersona + ", nombre=" + nombre + "];"  
}  
}
```


9. CREAR UNA CLASE JAVA

- Creamos la clase HolaMundoHibernate.java. Esta será nuestra clase de prueba:



9. CREAR UNA CLASE JAVA

- Creamos la clase HolaMundoHibernate.java:

New Java Class

Steps

1. Choose File Type
2. **Name and Location**

Name and Location

Class Name:

Project:

Location:

Package:

Created File:

< Back Next > **Finish** Cancel Help

PASO 10. MODIFICAMOS EL CÓDIGO

Archivo HolaMundoHibernate.java:

Clic para ver el
archivo

```
package test;

import java.util.List;

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

import model.Persona;

public class HolaMundoHibernate {

    public static void main(String[] args) {
        String hql = "SELECT p FROM Persona p";
        EntityManagerFactory fabrica = Persistence.createEntityManagerFactory("HibernateEjemplo1");
        EntityManager entityManager = fabrica.createEntityManager();
        Query query = entityManager.createQuery(hql);
        List<Persona> list = query.getResultList();
        for (Persona p : list) {
            System.out.println("persona:" + p);
        }
    }
}
```


11. CREAR UN ARCHIVO XML

Creamos un archivo log4j2.xml. El API de log4j nos permite manejar el log o bitácora de una aplicación Java de manera más simple.

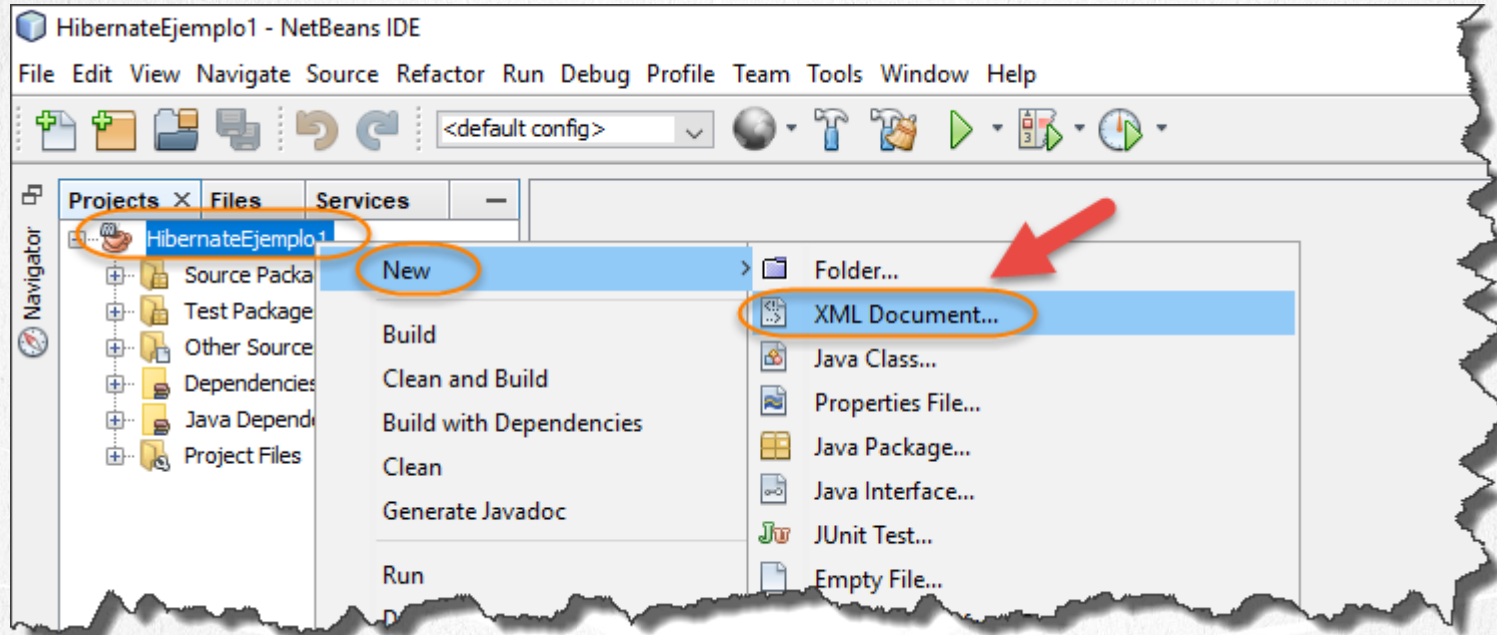
Para poder utilizar este API solo se requiere agregar las librerías de log4j las cuales ya han sido agregadas al archivo pom.xml de maven, y el archivo log4j2.xml en algún lugar que reconozca el classpath, por ejemplo en la carpeta de resources del proyecto.

Con esto estaremos listos para especificar que información queremos que se envíe a la consola u otros lugares, como un archivo. Para más información de esta API consultar:

<https://logging.apache.org/log4j/2.x/>

11. CREAR UN ARCHIVO XML

- Creamos el archivo log4j2.xml:



11. CREAR UN ARCHIVO XML

- Creamos el archivo log4j2.xml:

New XML Document

Steps

1. Choose File Type
- 2. Name and Location**
3. Select Document Type
4. ...

Name and Location

File Name:

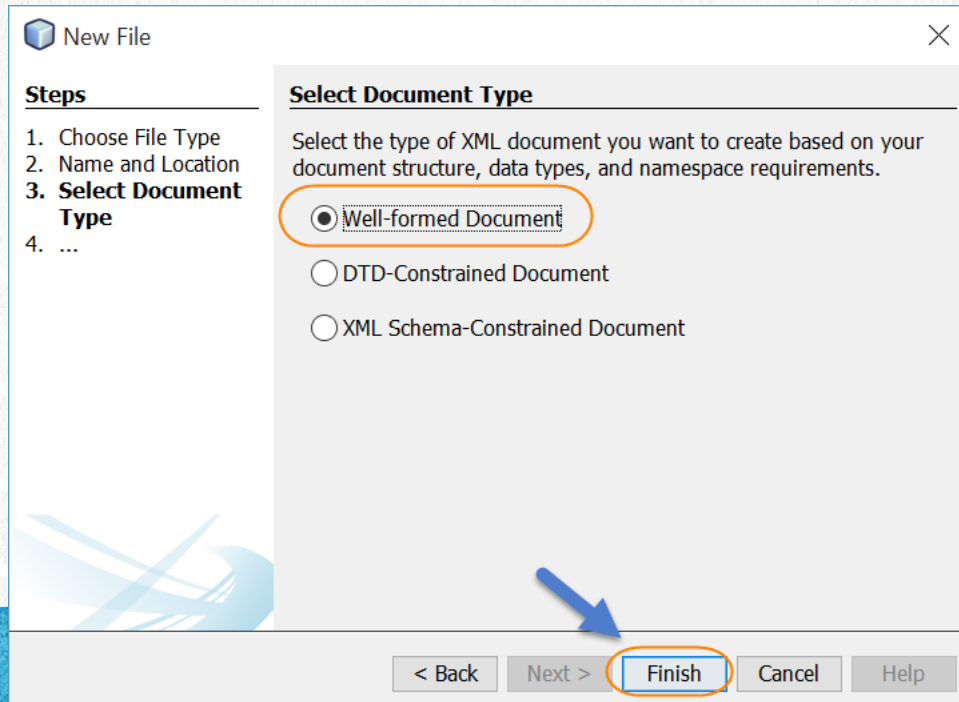
Project:

Folder:

Created File:

11. CREAR UN ARCHIVO XML

- Creamos el archivo log4j2.xml. En este paso seleccionamos cualquier opción, no es importante ya que vamos a sobrescribir el archivo:



PASO 12. MODIFICAMOS EL CÓDIGO

Archivo log4j2.xml:

Clic para ver el
archivo

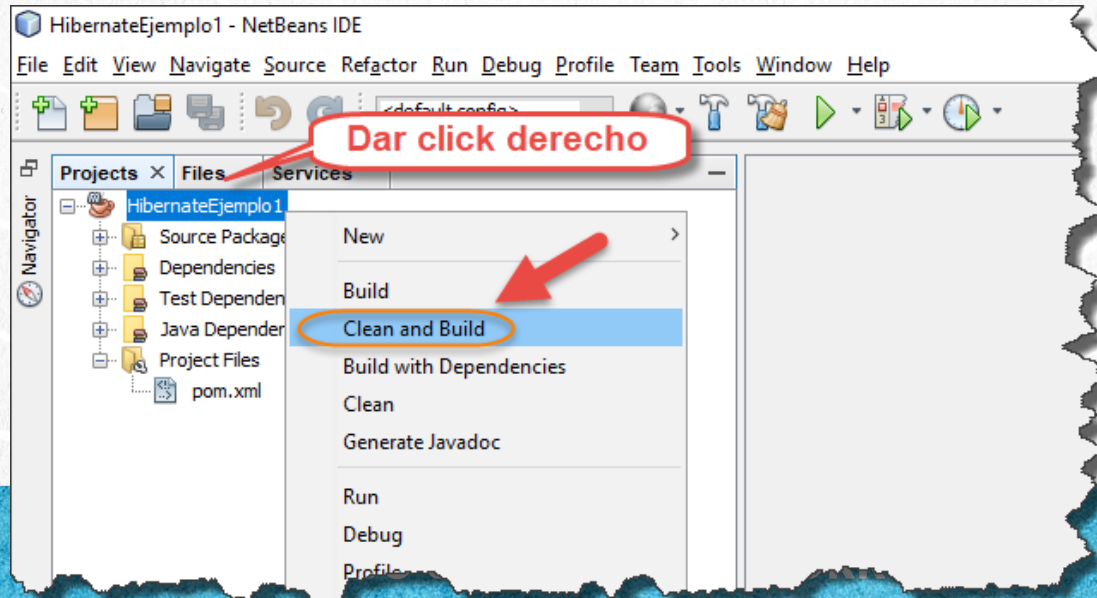
```
<?xml version="1.0" encoding="UTF-8"?>
<Configuration status="INFO">
  <Appenders>
    <Console name="Console" target="SYSTEM_OUT">
      <PatternLayout pattern="%d{HH:mm:ss} [%t] %-5level %logger{36} - %msg%n" />
    </Console>
  </Appenders>
  <Loggers>
    <Logger name="org.hibernate.SQL" level="debug" additivity="false">
      <AppenderRef ref="Console"/>
    </Logger>
    <logger name="org.hibernate.type.descriptor.sql.BasicBinder" level="trace" additivity="false">
      <AppenderRef ref="Console"/>
    </logger>
    <Root level="info">
      <AppenderRef ref="Console" />
    </Root>
  </Loggers>
</Configuration>
```

CURSO DE JAVA CON JDBC

www.globalmentoring.com.mx

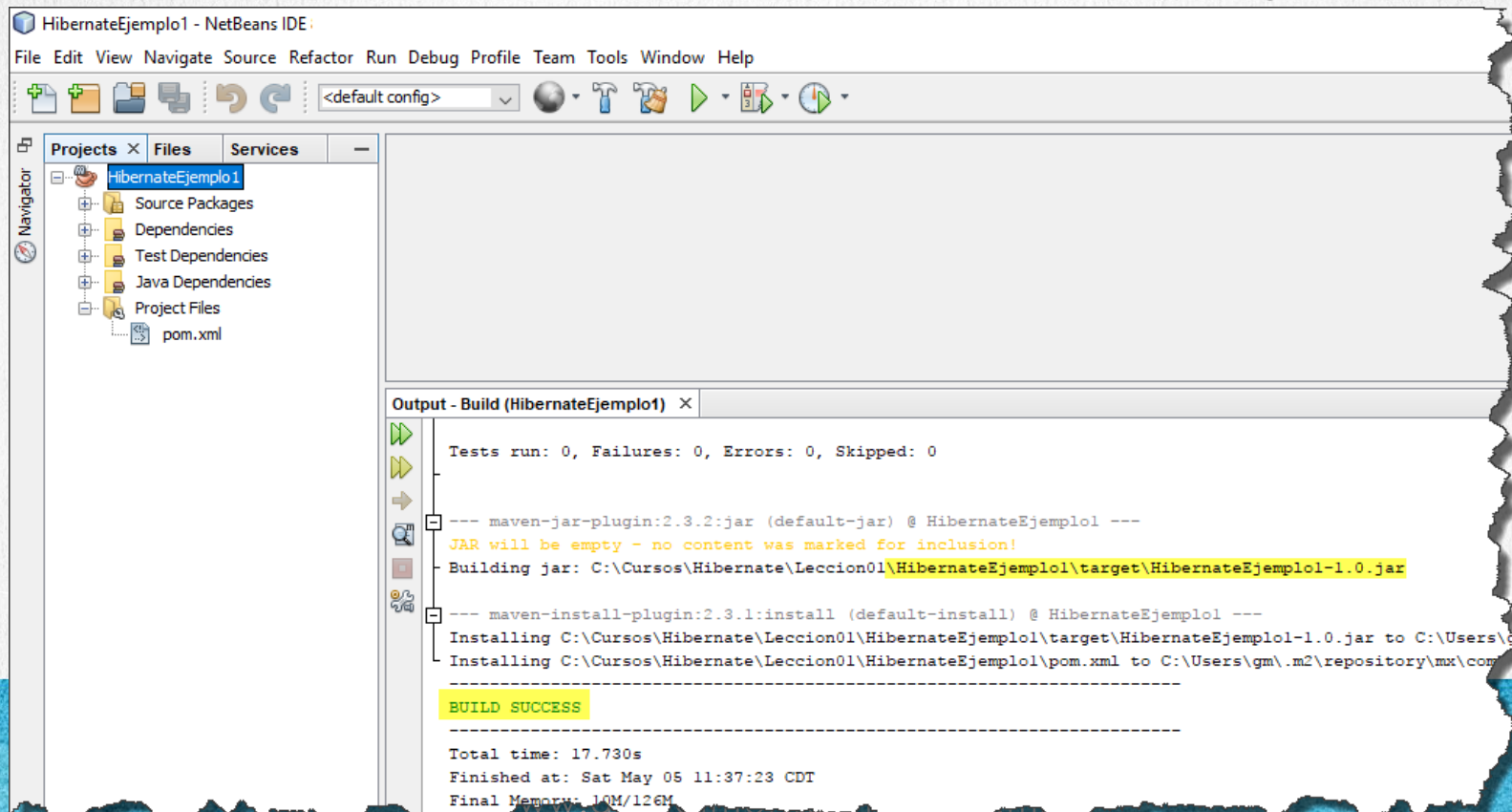
13. HACEMOS CLEAN & BUILD

- Para que se descarguen las librerías, hacemos Clean & Build al proyecto. Si por alguna razón este proceso falla, se debe desactivar cualquier software como antivirus, Windows defender o firewall durante este proceso para que no se impida la descarga de archivos .jar de Java. Una vez terminado se pueden volver a activar estos servicios. Este proceso puede demorar varios minutos dependiendo de su velocidad de internet:



13. HACEMOS CLEAN & BUILD

- Una vez terminado el proceso se debe mostrar una salida similar a la siguiente:



The screenshot shows the NetBeans IDE interface for a project named 'HibernateEjemplo1'. The 'Output - Build (HibernateEjemplo1)' window is open, displaying the following build log:

```
Tests run: 0, Failures: 0, Errors: 0, Skipped: 0

--- maven-jar-plugin:2.3.2:jar (default-jar) @ HibernateEjemplo1 ---
JAR will be empty - no content was marked for inclusion!
Building jar: C:\Cursos\Hibernate\Leccion01\HibernateEjemplo1\target\HibernateEjemplo1-1.0.jar

--- maven-install-plugin:2.3.1:install (default-install) @ HibernateEjemplo1 ---
Installing C:\Cursos\Hibernate\Leccion01\HibernateEjemplo1\target\HibernateEjemplo1-1.0.jar to C:\Users\c...
Installing C:\Cursos\Hibernate\Leccion01\HibernateEjemplo1\pom.xml to C:\Users\gm\.m2\repository\mx\com\...

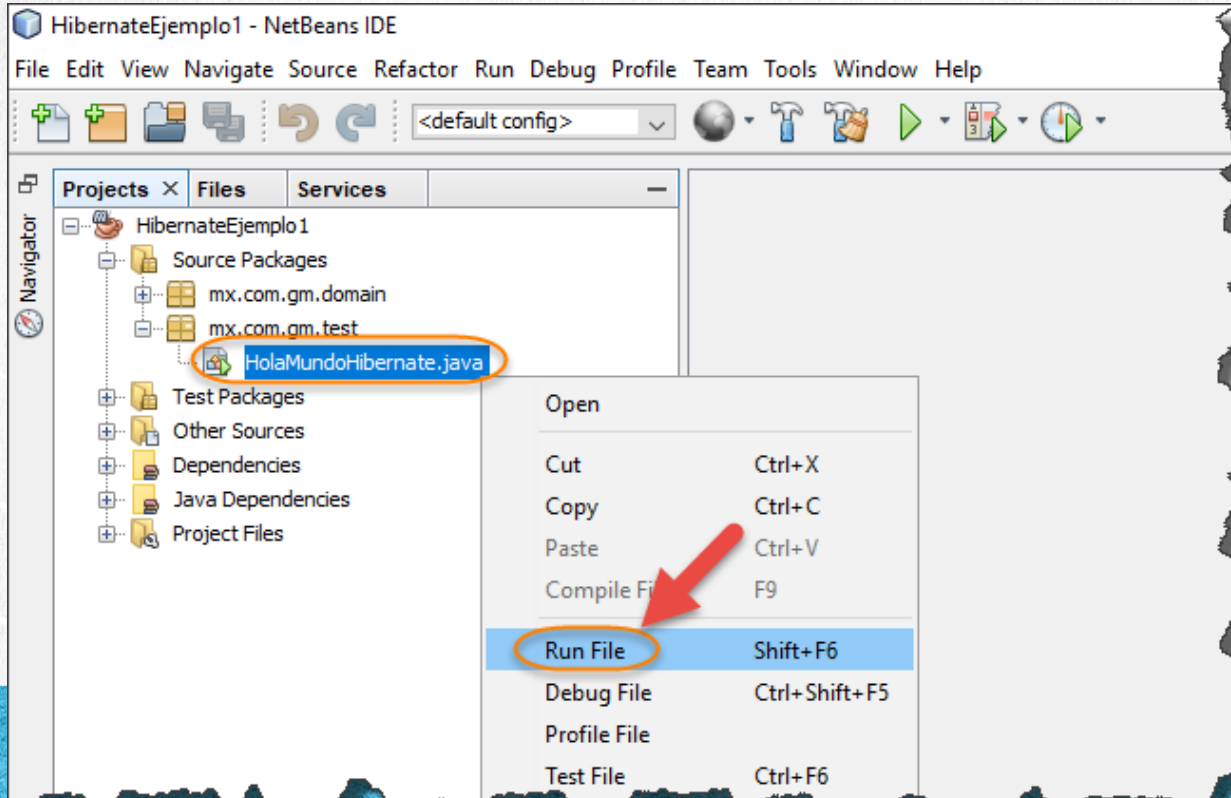
BUILD SUCCESS

-----

Total time: 17.730s
Finished at: Sat May 05 11:37:23 CDT
Final Memory: 10M/126M
```

14. EJECUTAMOS LA APLICACIÓN

- Ejecutamos la aplicación desde la clase HolaMundoHibernate.java:



14. EJECUTAMOS LA APLICACIÓN

- Vemos que se listan las personas almacenadas en la tabla de Personas a través del query de Hibernate/JPA que hemos ejecutado:

```
Output - Run (HolaMundoHibernate) x
-----
15:42:46 [main] INFO org.hibernate.jpa.internal.util.LogHelper - HHH000204: Processing PersistenceUnitInfo [
    name: HibernateEjemplol
    ...]
15:42:46 [main] INFO org.hibernate.Version - HHH000412: Hibernate Core {5.3.0.CR2}
15:42:46 [main] INFO org.hibernate.cfg.Environment - HHH000206: hibernate.properties not found
15:42:46 [main] INFO org.hibernate.annotations.common.Version - HCANN000001: Hibernate Commons Annotations {5.0.3.Final}
15:42:47 [main] WARN org.hibernate.orm.connections.pooling - HHH10001002: Using Hibernate built-in connection pool (not for production use!)
15:42:47 [main] INFO org.hibernate.orm.connections.pooling - HHH10001005: using driver [com.mysql.jdbc.Driver] at URL [jdbc:mysql://localhost:3306/sga?useSSL=true]
15:42:47 [main] INFO org.hibernate.orm.connections.pooling - HHH10001001: Connection properties: {user=root, password=****}
15:42:47 [main] INFO org.hibernate.orm.connections.pooling - HHH10001003: Autocommit mode: false
15:42:47 [main] INFO org.hibernate.engine.jdbc.connections.internal.DriverManagerConnectionProviderImpl - HHH000115: Hibernate connection pool size: 20 (min=1)
15:42:47 [main] INFO org.hibernate.dialect.Dialect - HHH000400: Using dialect: org.hibernate.dialect.MySQL57Dialect
15:42:48 [main] WARN org.hibernate.type.spi.TypeConfiguration$Scope - HHH000233: Scoping types to session factory org.hibernate.internal.SessionFactoryImpl@19fb8826 after already
15:42:48 [main] INFO org.hibernate.hql.internal.QueryTranslatorFactoryInitiator - HHH000397: Using ASTQueryTranslatorFactory
15:42:49 [main] DEBUG org.hibernate.SQL - select persona0_.id_persona as id_persona0_, persona0_.apellido as apellido2_0_, persona0_.nombre as nombre3_0_ from Persona persona0_
15:42:49 [main] INFO mx.com.gm.test.HolaMundoHibernate - Listado de Personas:
15:42:49 [main] INFO mx.com.gm.test.HolaMundoHibernate - persona:Persona [apellido=Martinez, idPersona=10, nombre=Juan Carlos]
```


CONCLUSIÓN DEL EJERCICIO

Con este ejercicio hemos creado nuestro primer proyecto con Hibernate Framework.

Utilizar tanto Eclipse como Netbeans, o cualquier otro IDE es prácticamente lo mismo, ya que no estamos utilizando ningún Wizard de los IDEs, los cuales se pueden utilizar, pero en nuestro caso enseñaremos los conceptos desde cero, por lo tanto no será necesario utilizar los Wizards, mas que para automatizar ciertas tareas lo mínimo necesario.

Sin embargo, ya teniendo claro los conceptos aplicados y aprendidos en el curso, rápidamente podrán aplicar los Wizards y ventajas que su IDE de preferencia les ofrezca, sin embargo la idea del curso no es aprender a utilizar un IDE, sino poder aplicar estos conceptos a cualquier IDE, sea Eclipse, Netbeans, IntelliJ Idea o cualquier otro IDE.

CURSO HIBERNATE Y JPA

www.globalmentoring.com.mx

CURSO ONLINE

HIBERNATE & JPA

Por: Ing. Ubaldo Acosta



Experiencia y Conocimiento para tu vida



CURSO HIBERNATE Y JPA
www.globalmentoring.com.mx