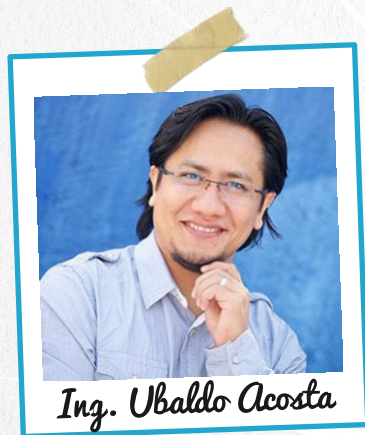


CURSO HIBERNATE Y JPA

HOLA MUNDO CON HIBERNATE/JPA Y NETBEANS



Por el experto: Ing. Ubaldo Acosta



Experiencia y Conocimiento para tu vida

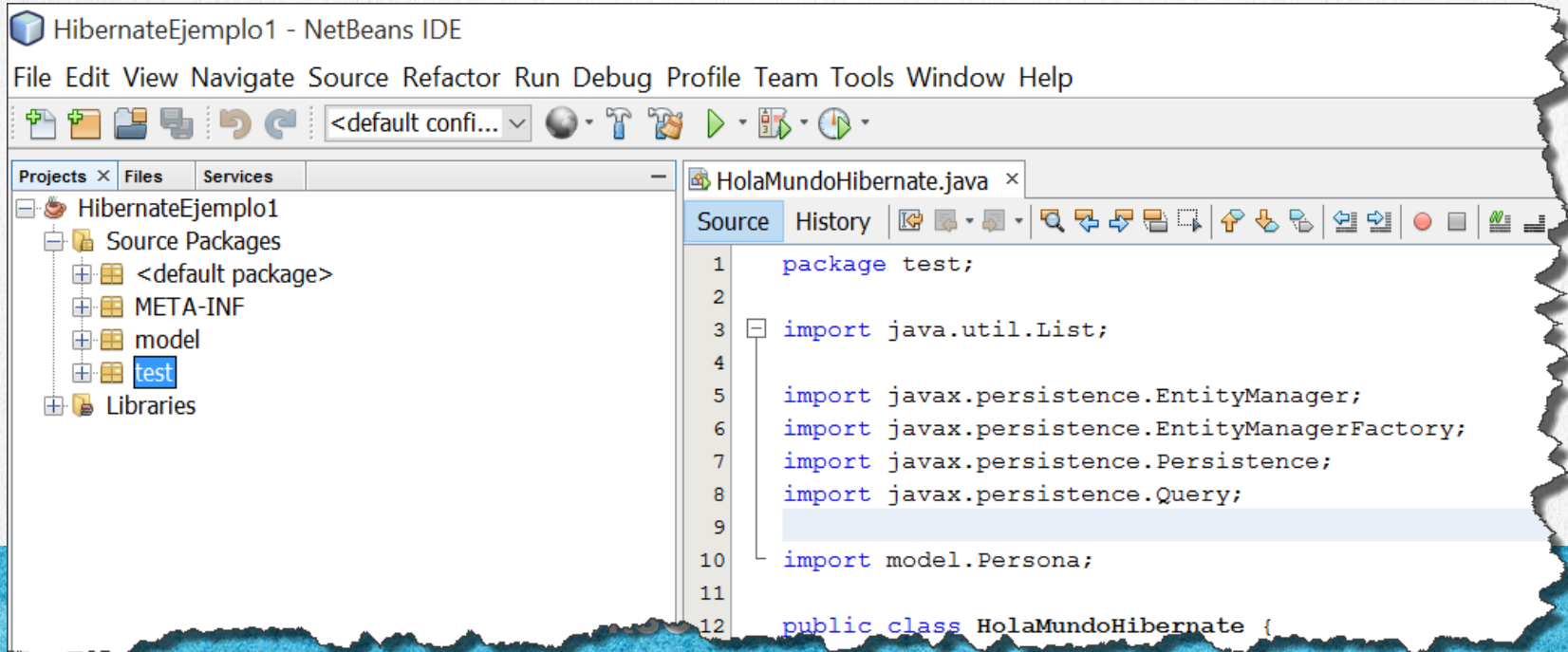


CURSO HIBERNATE Y JPA

www.globalmentoring.com.mx

OBJETIVO DEL EJERCICIO

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



INSTALACIÓN DEL JDK DE JAVA Y NETBEANS

- Si aun no tienen instalado el JDK de Java pueden seguir la siguiente guía:

<http://icursos.net/media/documents/CFJ-B-Ejercicio-01a-Instalacion-jdk-windows.pdf>

<http://icursos.net/media/documents/CFJ-B-Ejercicio-01b-Instalacion-jdk-mac.pdf>

- Si aún no tiene instalado el IDE de Netbeans pueden utilizar esta guía:

<http://icursos.net/media/documents/CFJ-B-Ejercicio-03a-Instalacion-Netbeans-Windows.pdf>

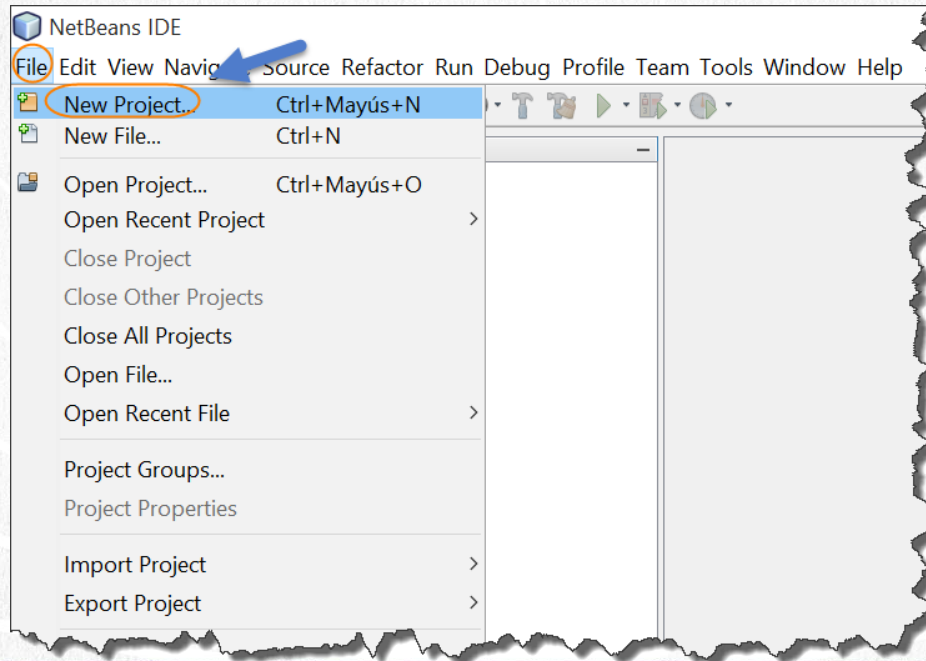
<http://icursos.net/media/documents/CFJ-B-Ejercicio-03b-Instalacion-Netbeans-Mac.pdf>

CURSO HIBERNATE Y JPA

www.globalmentoring.com.mx

1. CREAR PROYECTO

- Creamos un nuevo proyecto Java:

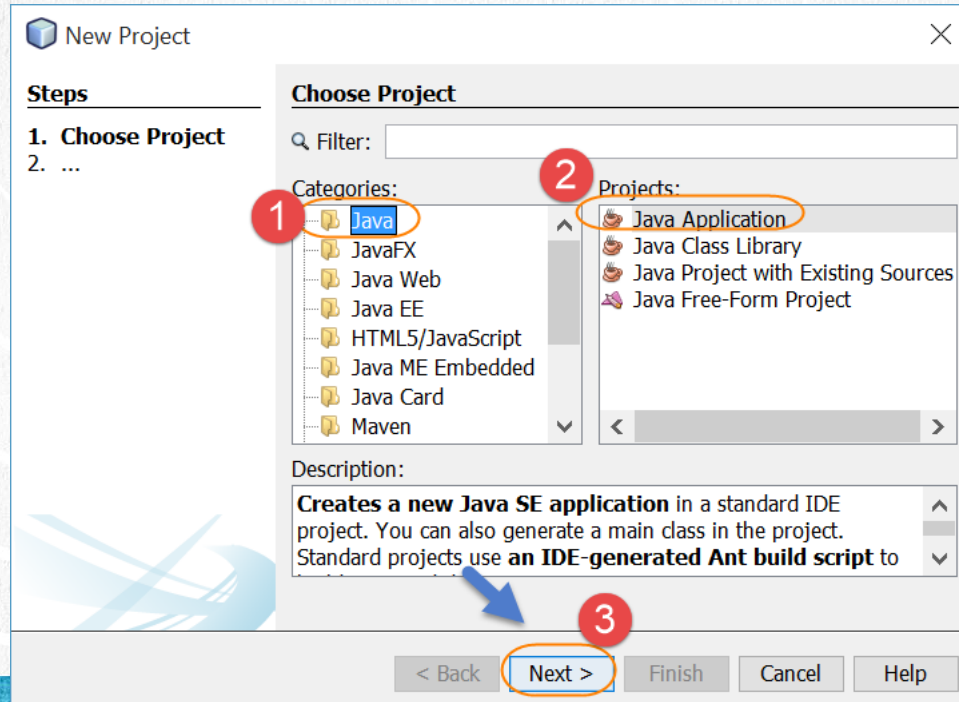


CURSO HIBERNATE Y JPA

www.globalmentoring.com.mx

1. CREAR PROYECTO

- Creamos un nuevo proyecto Java:



CURSO HIBERNATE Y JPA

www.globalmentoring.com.mx

1. CREAR PROYECTO

- Creamos un nuevo proyecto Java:

New Java Application

Steps

1. Choose Project
2. **Name and Location**

Name and Location

Project Name:

Project Location:

Project Folder:

☐ Use Dedicated Folder for Storing Libraries

Libraries Folder:

Different users and projects can share the same compilation libraries (see Help for details).

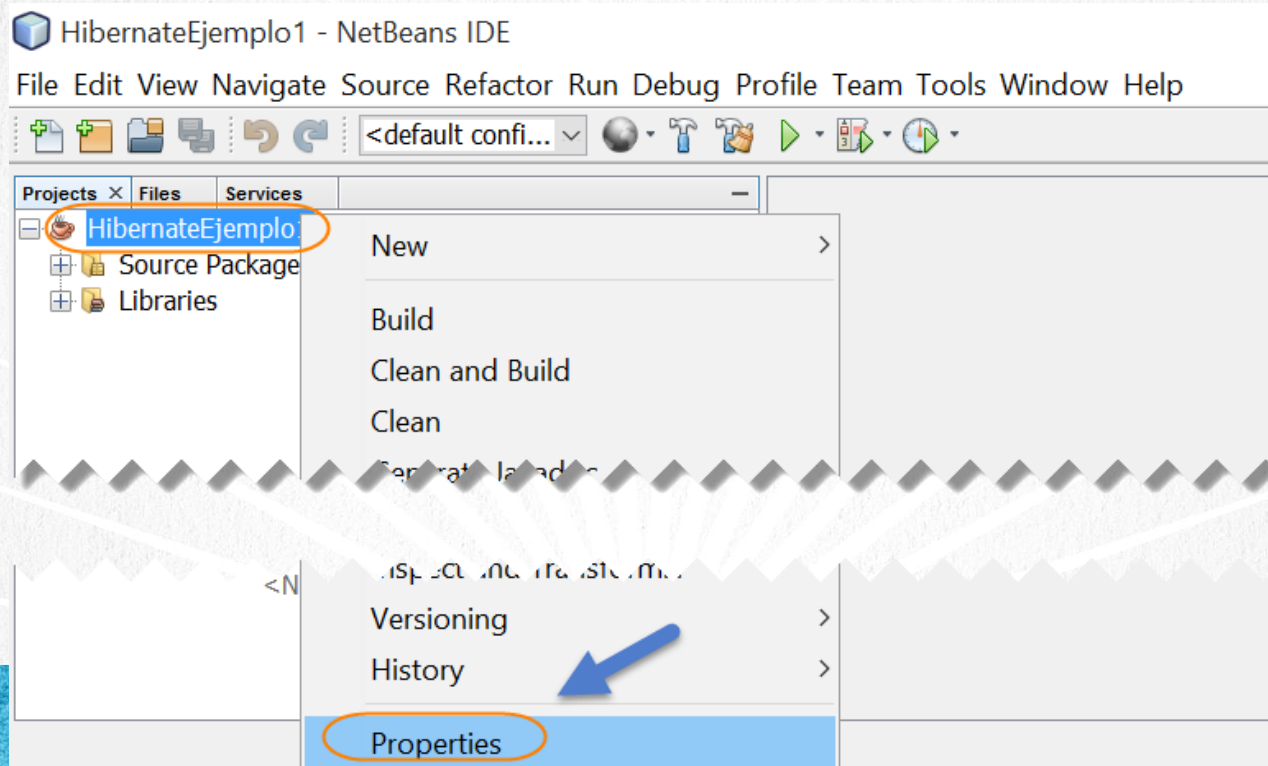
☐ Create Main Class

CURSO HIBERNATE Y JPA

www.globalmentoring.com.mx

1. CREAR PROYECTO

- Creamos un nuevo proyecto Java:



2. DESCARGA LIBRERÍAS HIBERNATE/JPA

Descargamos las librerías necesarias para trabajar con Hibernate/JPA:

<http://icursos.net/cursos/Hibernate/libs/hibernate-libs-1.zip>

Descomprimir el archivo y guardar todos los archivos .jars en alguna ruta de nuestro disco duro, por ejemplo:

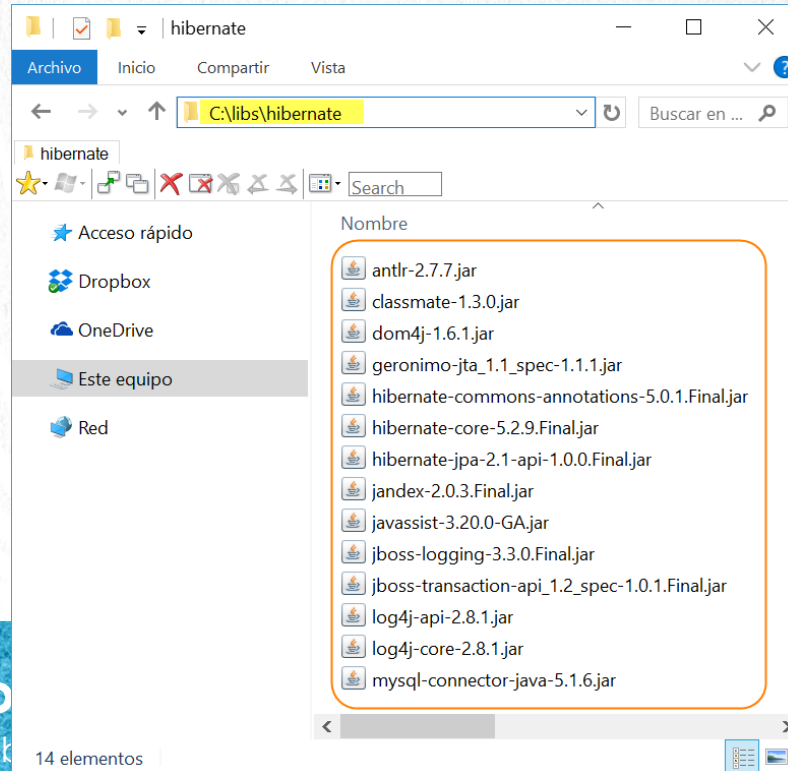
C:\libs\hibernate

NOTA 1: Si ya tienen descargadas las librerías omitir este paso 2.

NOTA 2: La versión de las librerías puede ser más reciente, así que se puede usar la última versión proporcionada, se estará actualizando este archivo de librerías, para que sigan siendo compatibles entre ellas y usen la versión más reciente.

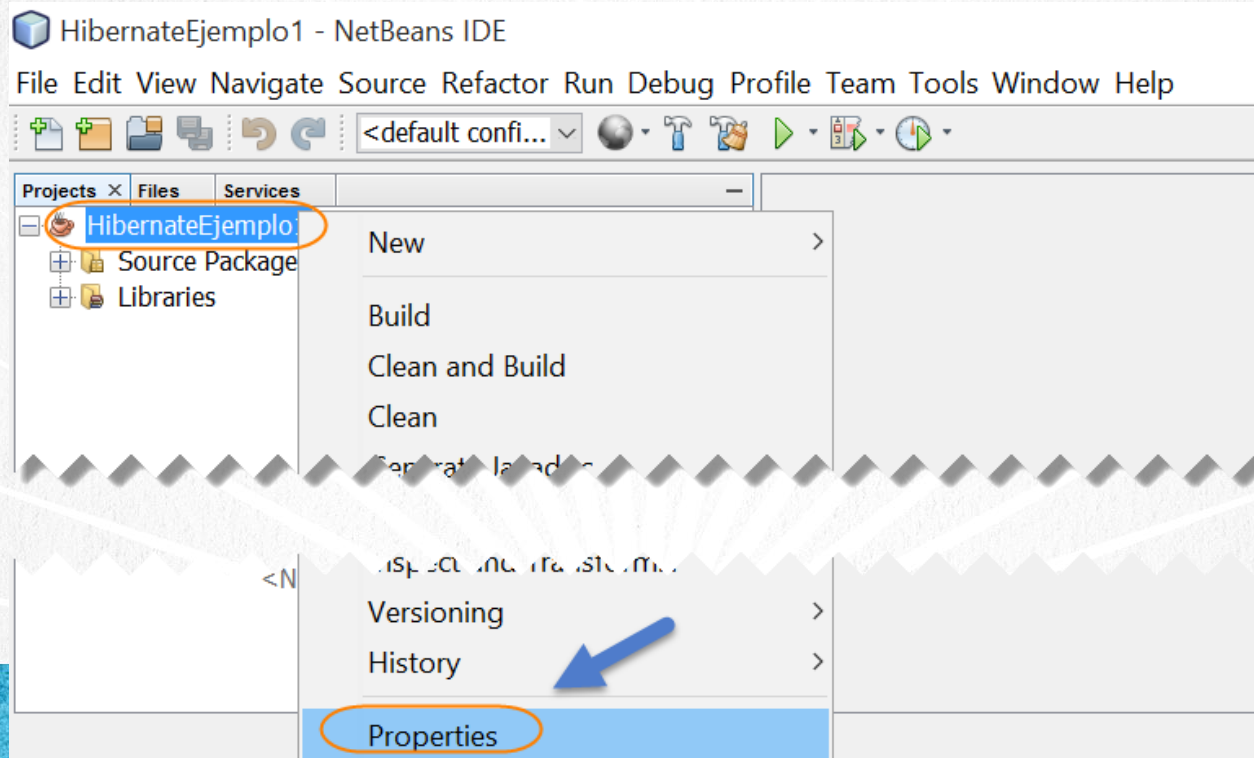
2. DESCARGA LIBRERÍAS HIBERNATE/JPA

Así quedan las librerías descomprimidas. La versión de las librerías puede ser más reciente según las librerías descargadas, pero se puede usar la versión más reciente :



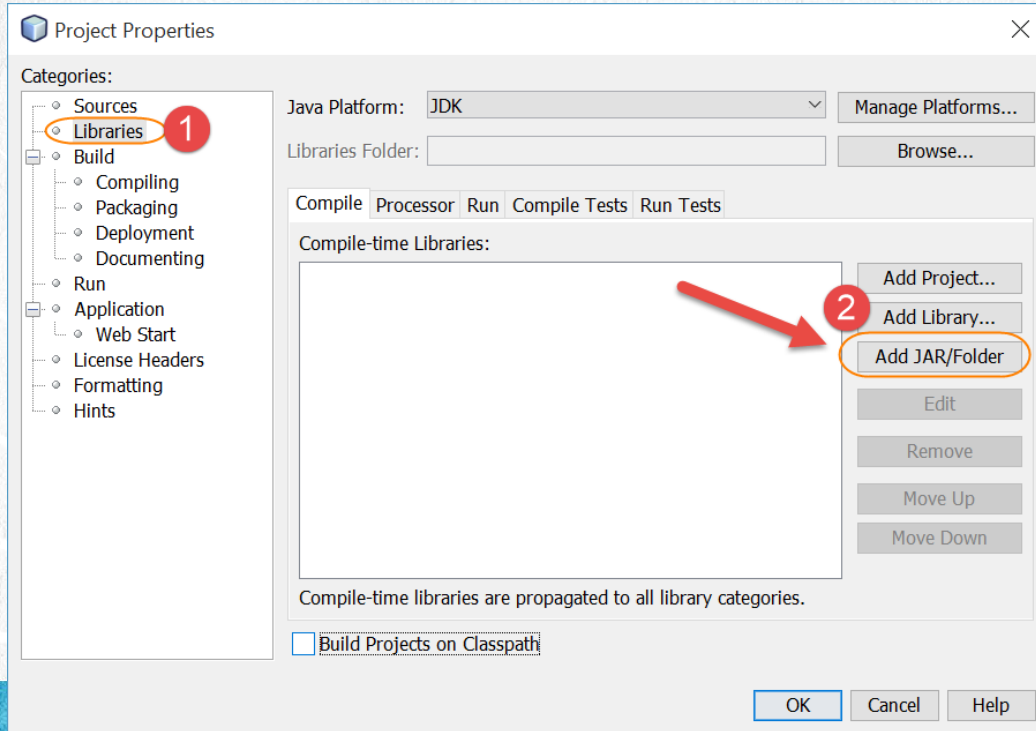
3. AGREGAR LAS LIBRERÍAS

- Agregamos las librerías necesarias para trabajar con Hibernate/JPA:



3. AGREGAR LAS LIBRERÍAS

- Agregamos las librerías necesarias para trabajar con Hibernate/JPA:

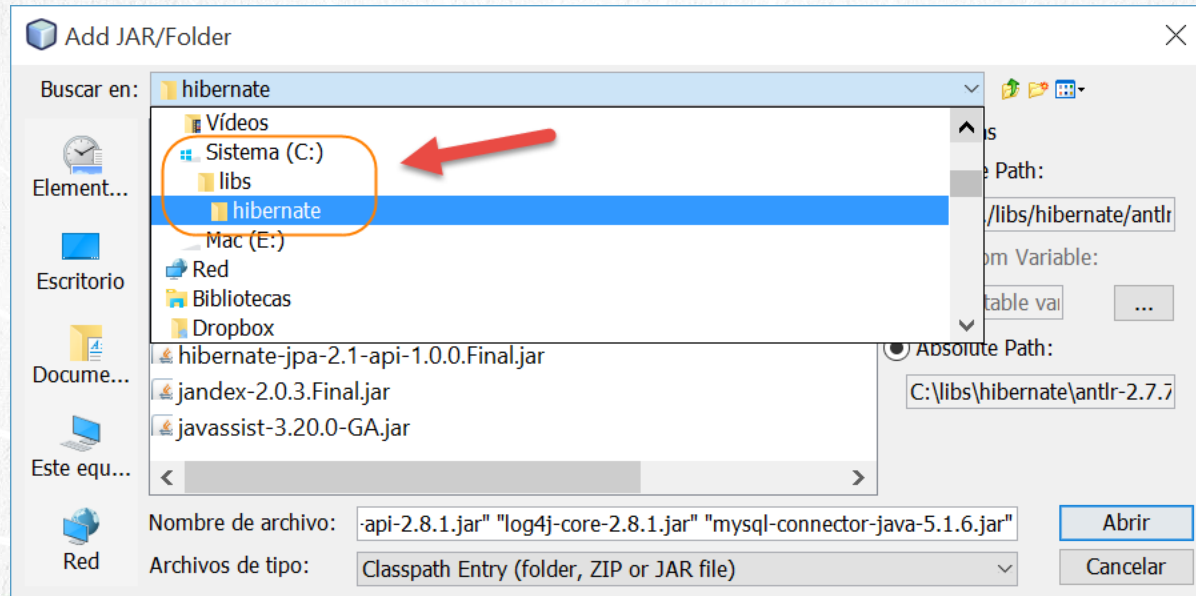


CURSO HIBERNATE Y JPA

www.globalmentoring.com.mx

3. AGREGAR LAS LIBRERÍAS

- Seleccionamos la ruta donde se agregaron las librerías. Ej. C:\libs\hibernate:



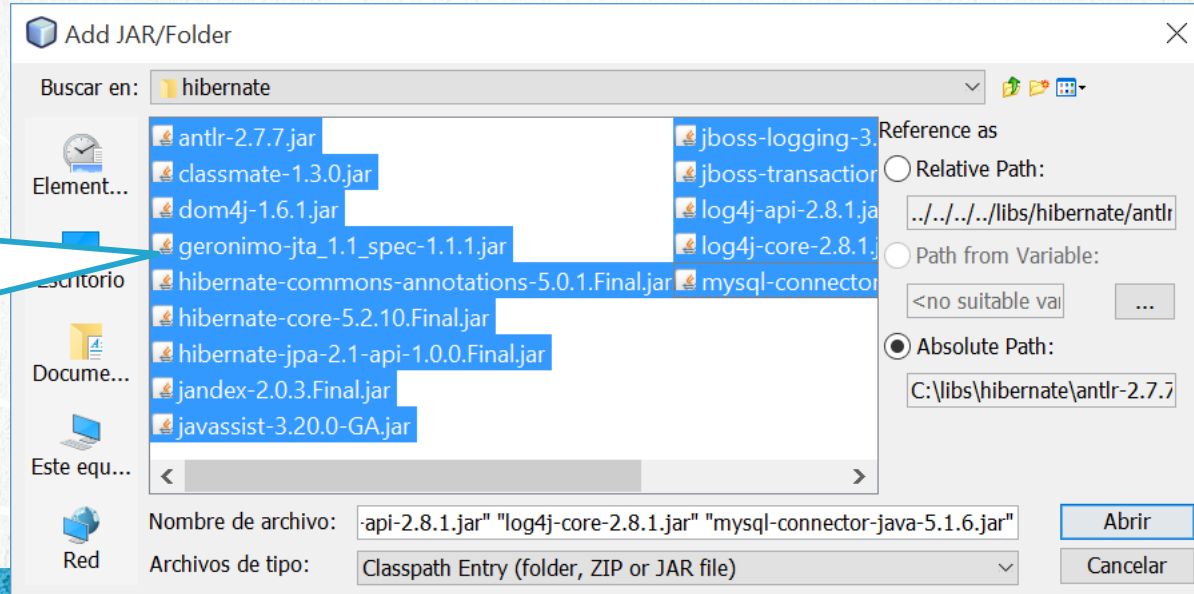
CURSO HIBERNATE Y JPA

www.globalmentoring.com.mx

3. AGREGAR LAS LIBRERÍAS

- Agregamos las librerías necesarias para trabajar con Hibernate/JPA. La versión de las librerías puede ser más reciente según las librerías descargadas:

Seleccionar
todas las
librerías

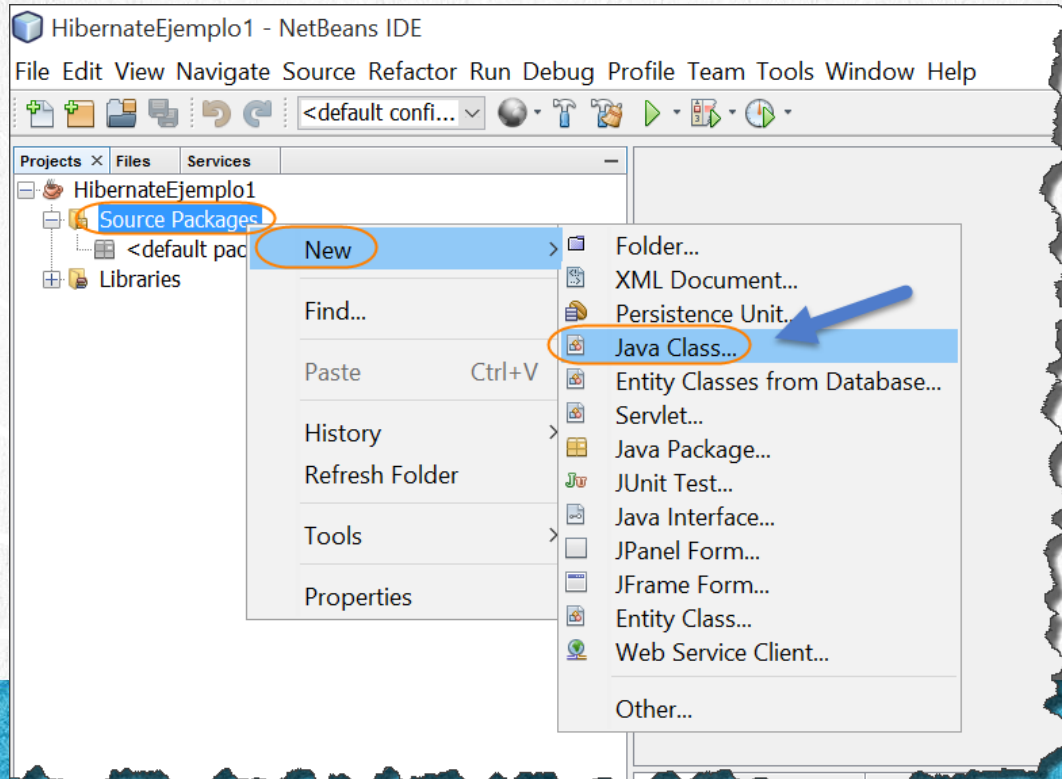


CURSO HIBERNATE Y JPA

www.globalmentoring.com.mx

4. CREAR LA CLASE PERSONA

- Creamos la clase Persona.java:



4. 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:

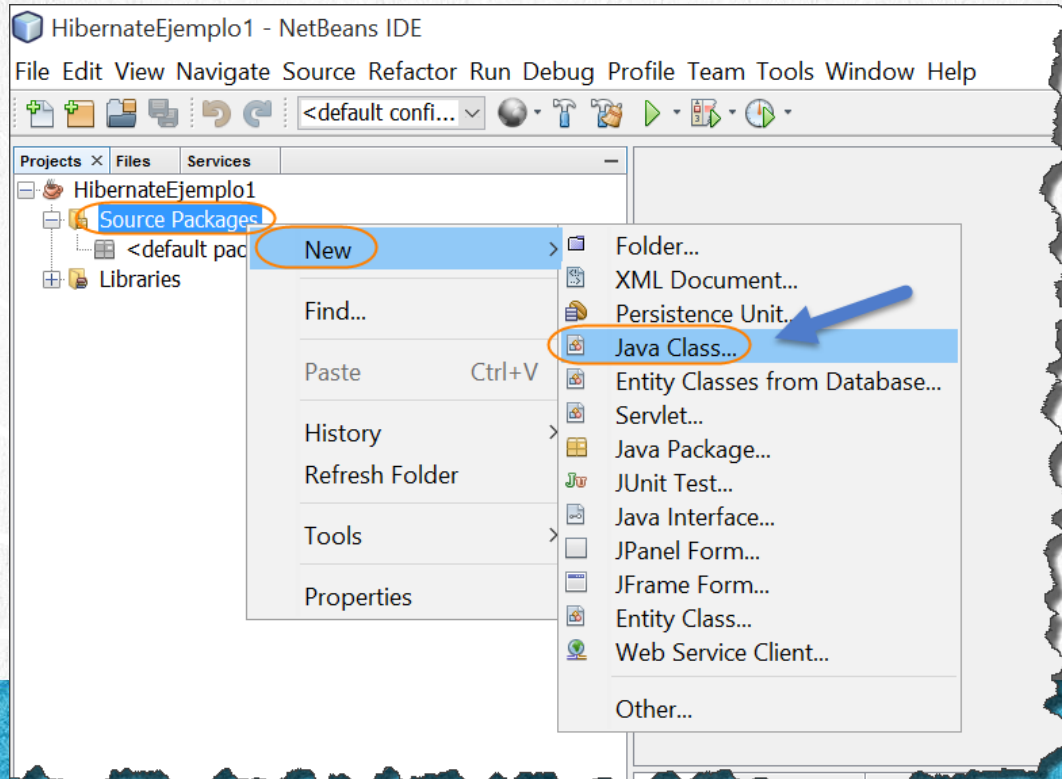
< Back Next > **Finish** Cancel Help

CURSO HIBERNATE Y JPA

www.globalmentoring.com.mx

5. CREAR UNA CLASE JAVA

- Creamos la clase HolaMundoHibernate.java:



5. 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:

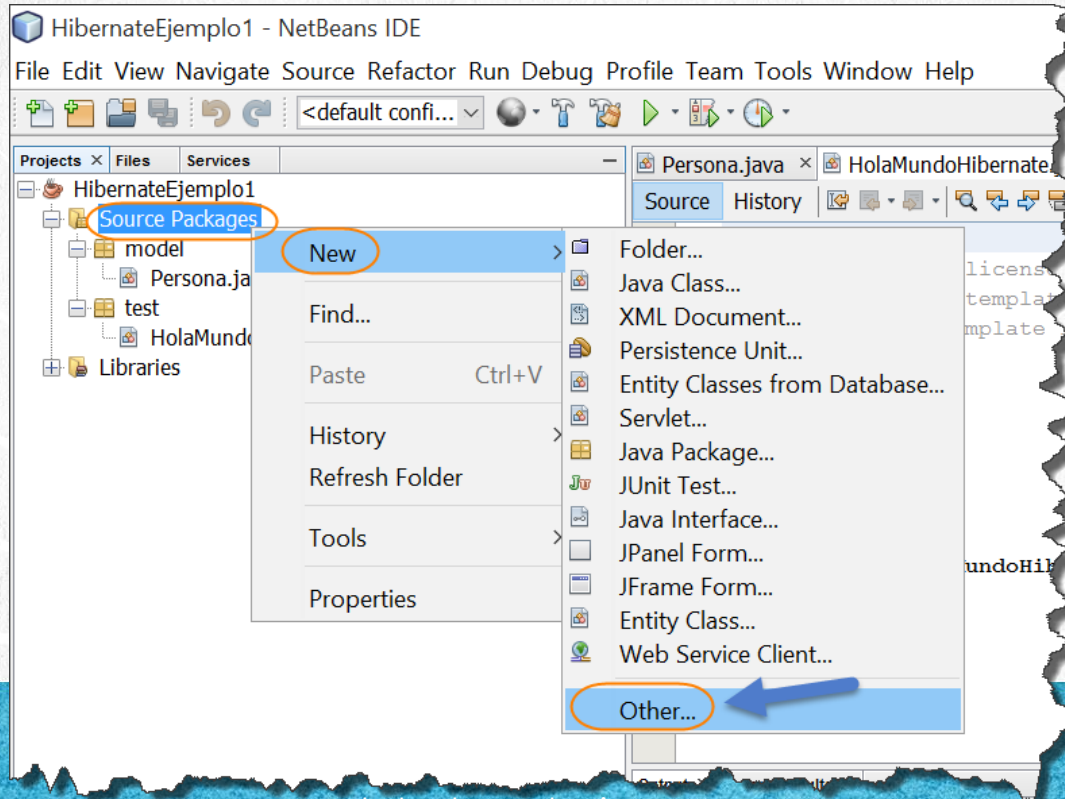
Created File:

CURSO HIBERNATE Y JPA

www.globalmentoring.com.mx

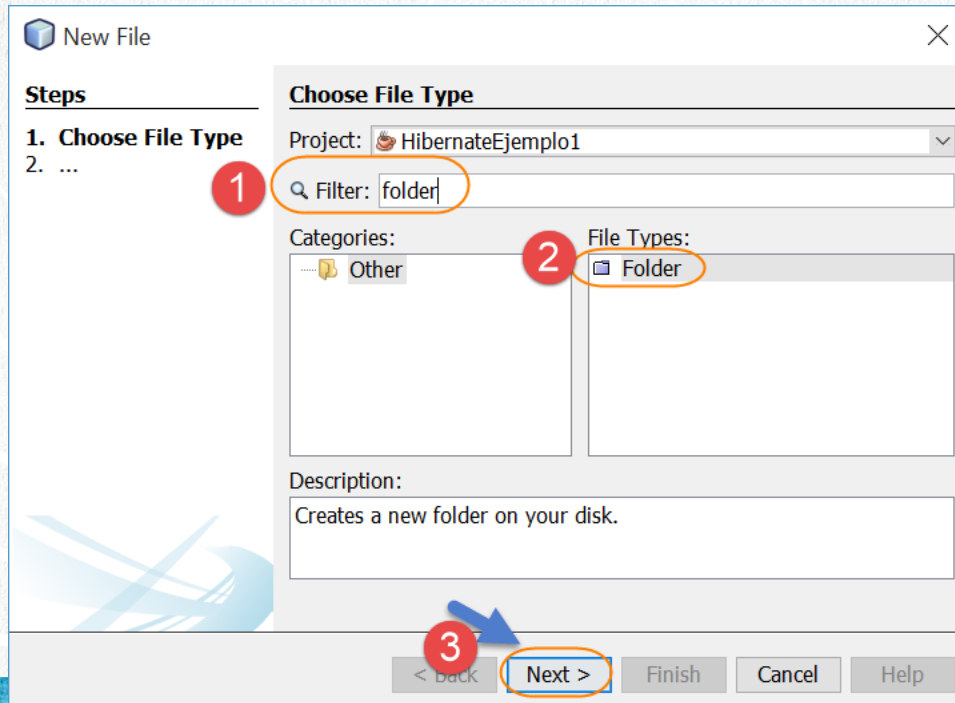
6. CREAR UNA CARPETA

- Creamos la carpeta META-INF:



6. CREAR UNA CARPETA

- Creamos la carpeta META-INF:

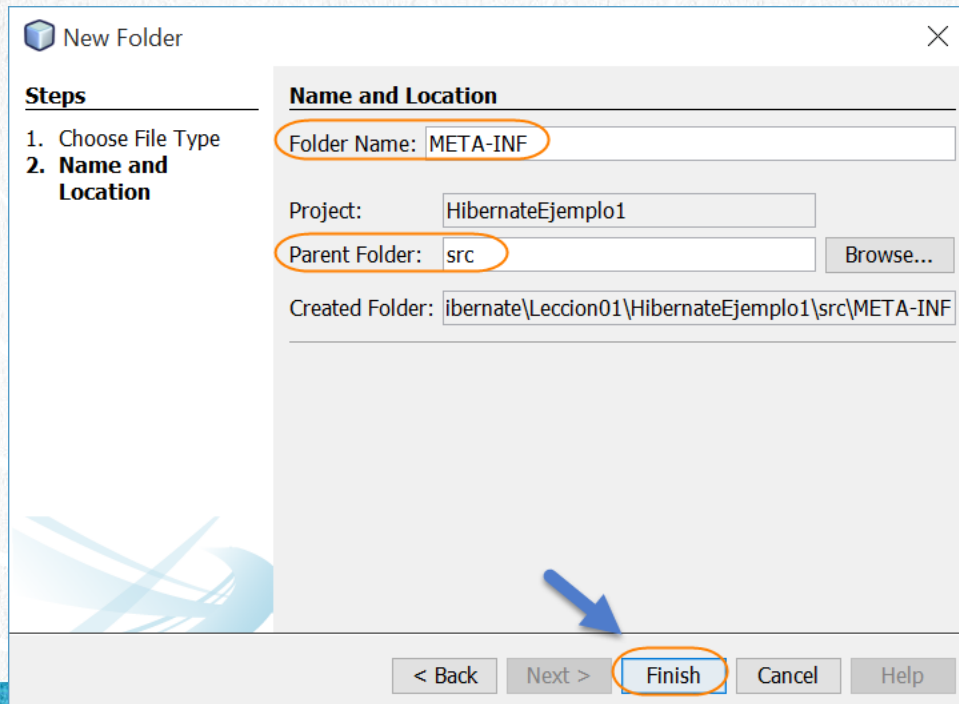


CURSO HIBERNATE Y JPA

www.globalmentoring.com.mx

6. CREAR UNA CARPETA

- Creamos la carpeta META-INF:

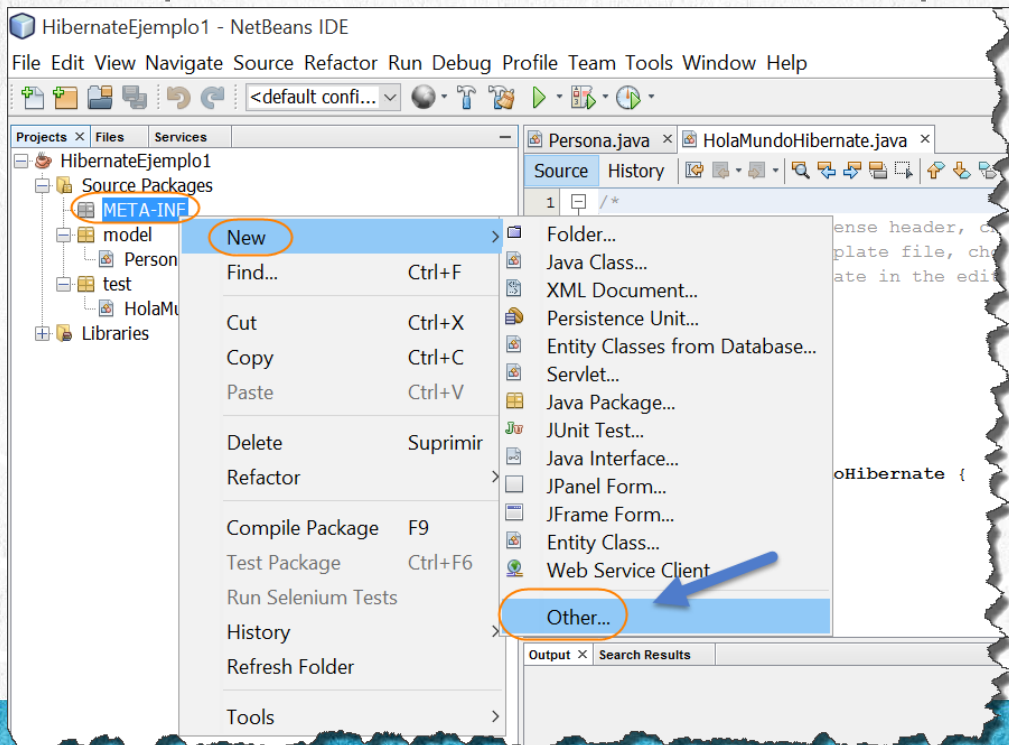


CURSO HIBERNATE Y JPA

www.globalmentoring.com.mx

7. CREAR UN ARCHIVO XML

- Creamos el archivo persistence.xml dentro de la carpeta META-INF:

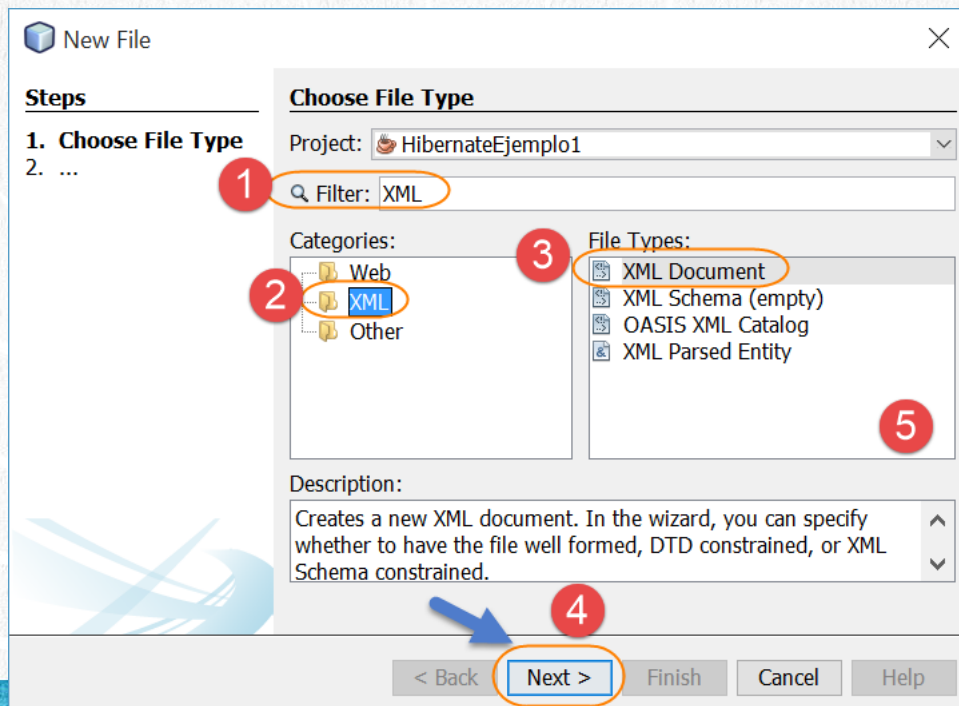


CURSO HIBERNATE Y JPA

www.globalmentoring.com.mx

7. CREAR UN ARCHIVO XML

- Filtramos por creación de archivos xml:

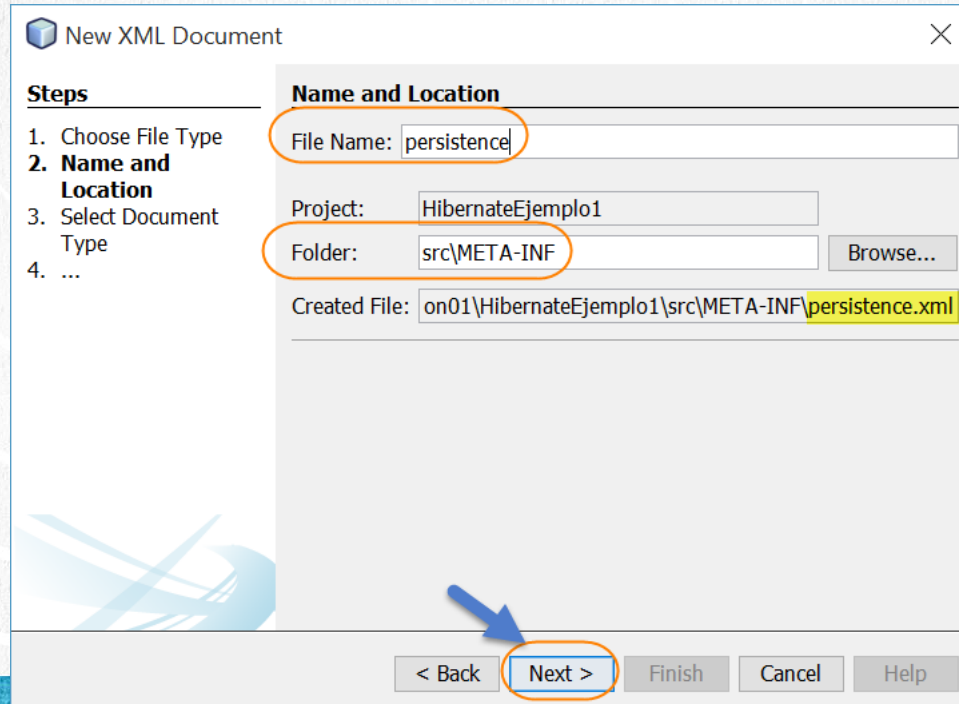


CURSO HIBERNATE Y JPA

www.globalmentoring.com.mx

7. CREAR UN ARCHIVO XML

- Creamos el archivo persistence.xml:



New XML Document

Steps

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

Name and Location

File Name: persistence

Project: HibernateEjemplo1

Folder: src\META-INF Browse...

Created File: on01\HibernateEjemplo1\src\META-INF\persistence.xml

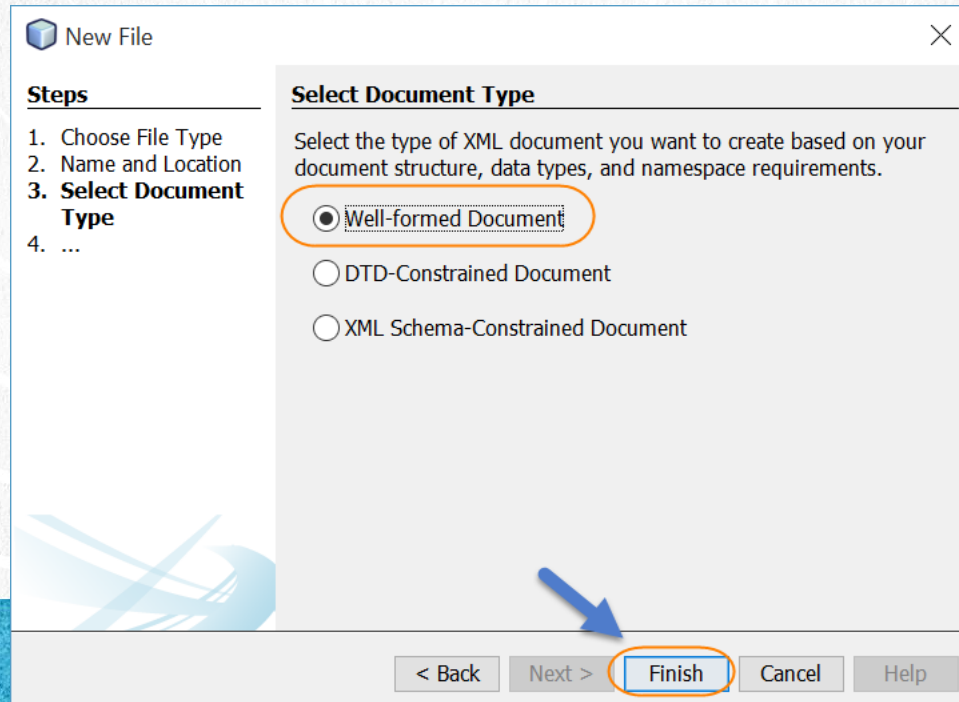
< Back **Next >** Finish Cancel Help

CURSO HIBERNATE Y JPA

www.globalmentoring.com.mx

7. CREAR UN ARCHIVO XML

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



8. CREAMOS 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.

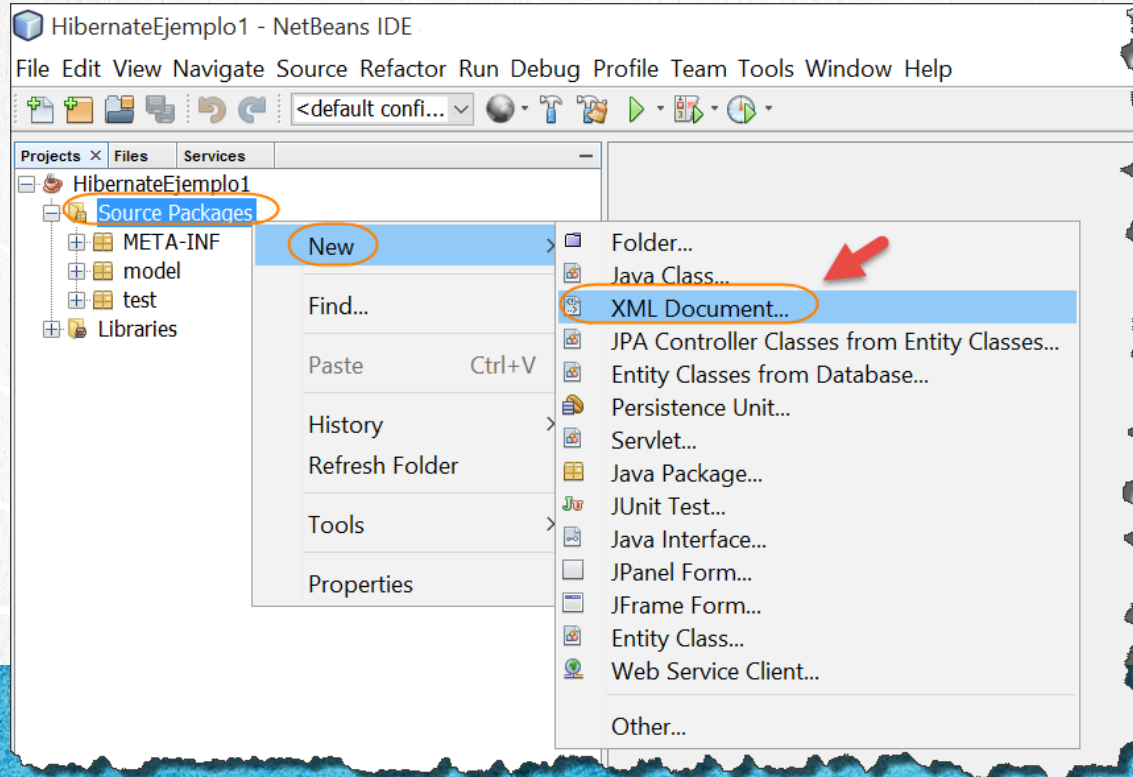
Para poder utilizar este API solo se requiere agregar las librerías de log4j las cuales ya han sido agregadas desde que se agregaron las librerías, y el archivo log4j2.xml en algún lugar que reconozca el classpath, por ejemplo en la carpeta src del proyecto.

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/>

8. CREAR UN ARCHIVO XML

- Creamos el archivo log4j2.xml dentro de la carpeta src:



8. 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: log4j2

Project: HibernateEjemplo1

Folder: src Browse...

Created File: \\Hibernate\\Leccion01\\HibernateEjemplo1\\src\\log4j2.xml

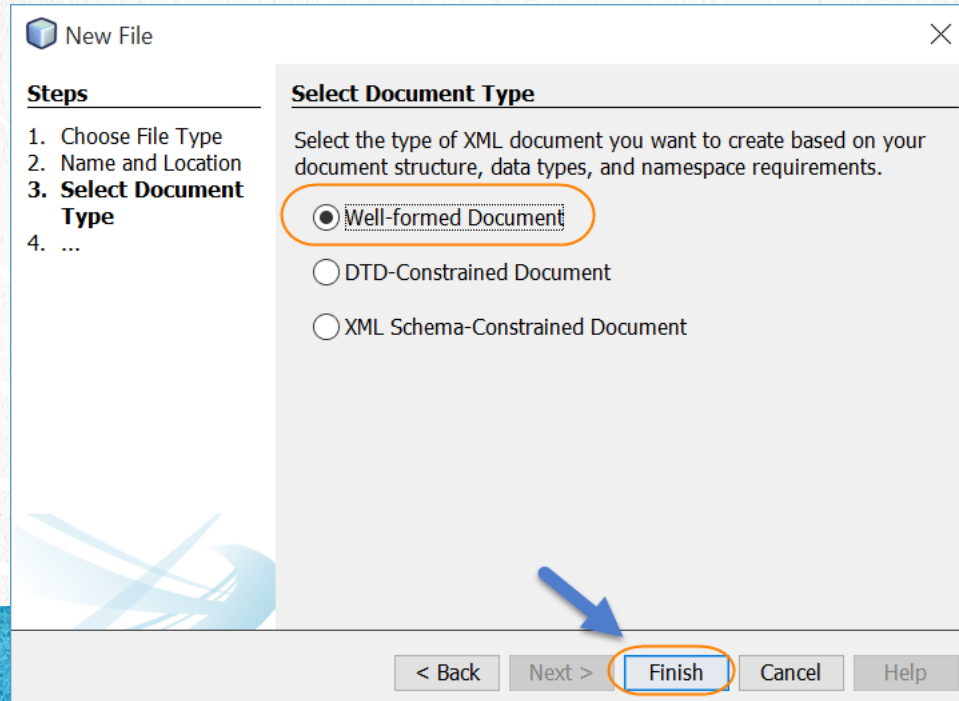
< Back **Next >** Finish Cancel Help

CURSO HIBERNATE Y JPA

www.globalmentoring.com.mx

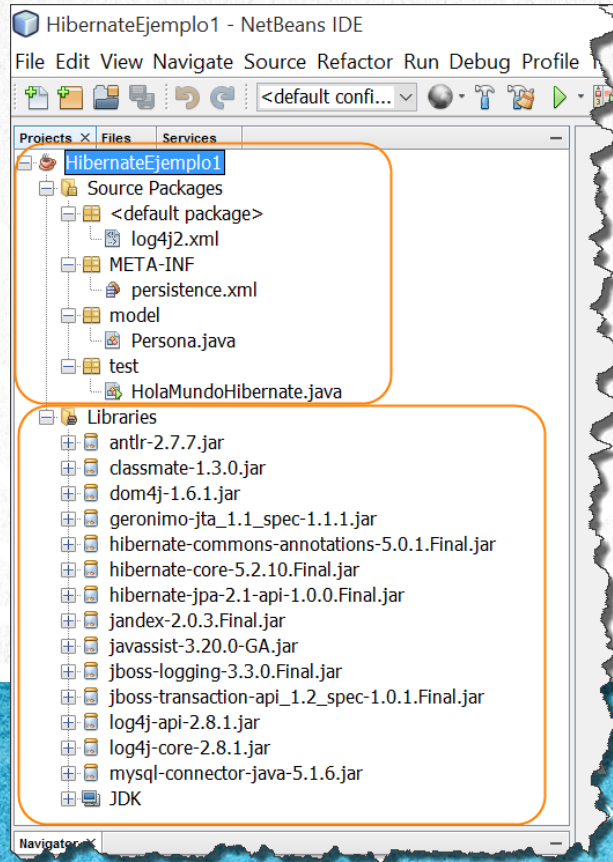
8. CREAR UN ARCHIVO XML

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



9. ESTRUCTURA DEL PROYECTO

- La estructura del proyecto queda como sigue:



PASO 10. MODIFICAMOS EL CÓDIGO

Archivo Persona.java:

[Click para descargar archivo](#)

```
package model;

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 10. 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 + "];"  
}  
}
```

PASO 11. 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);
        }
    }
}
```


PASO 12. 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>model.Persona</class>
    <properties>
      <property name="javax.persistence.jdbc.url" value="jdbc:mysql://localhost:3306/sga"/>
      <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"/>
    </properties>
  </persistence-unit>
</persistence>
```

PASO 13. MODIFICAMOS EL CÓDIGO

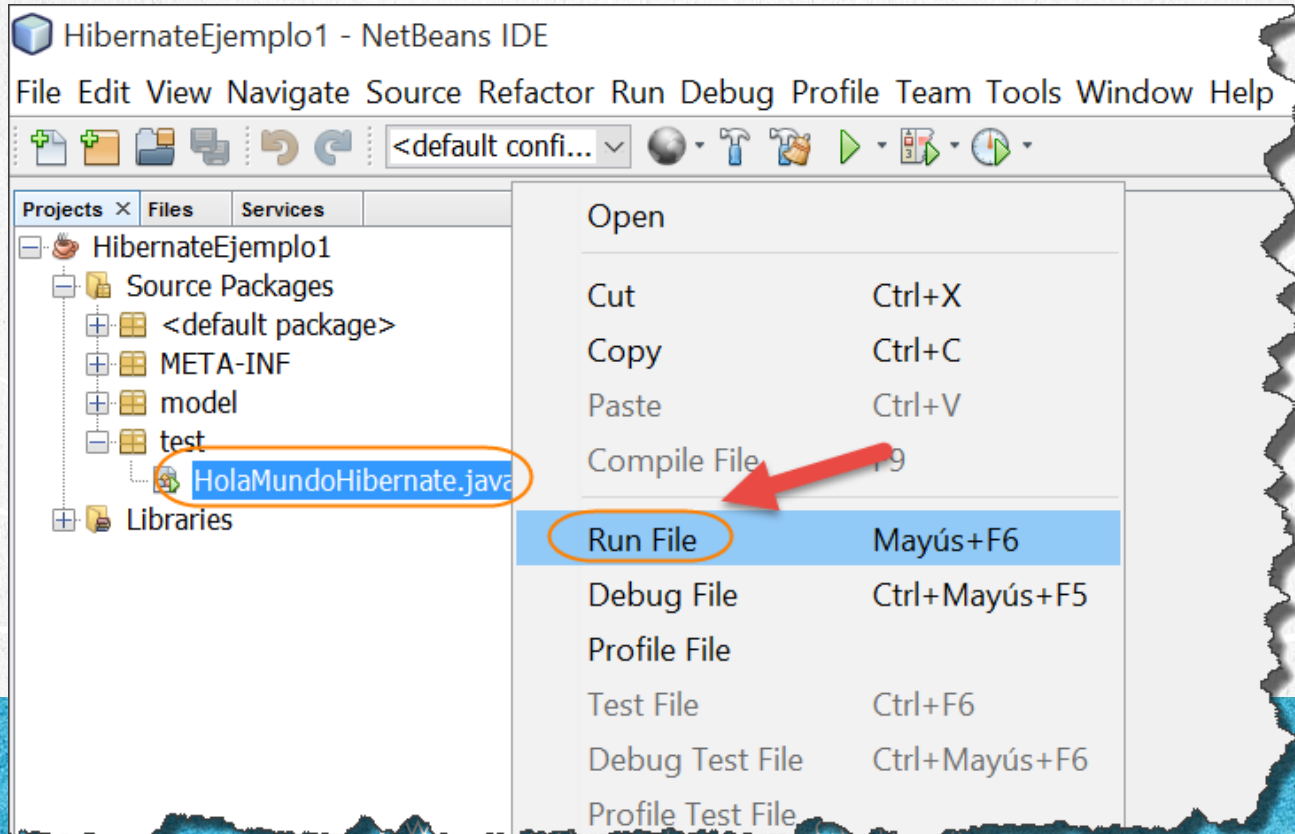
Archivo log4j2.xml:

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>
```

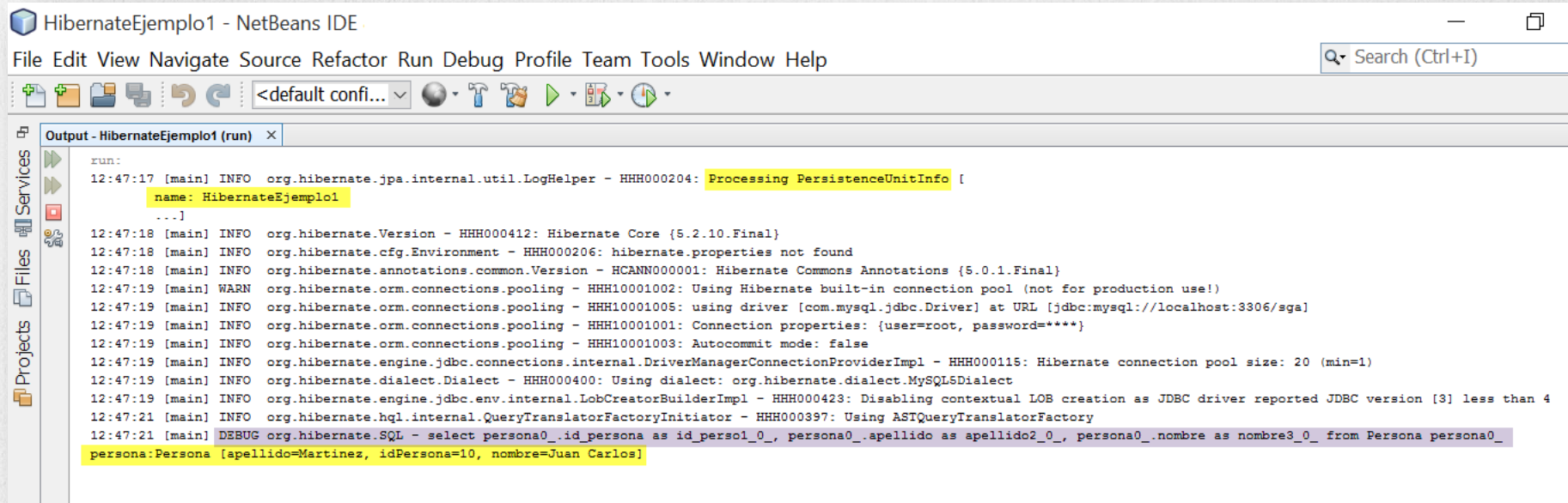

14. EJECUTAMOS LA APLICACIÓN

- Ejecutamos la aplicación desde la clase HolaMundoHibernate.java:



12. 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:



```
run:
12:47:17 [main] INFO org.hibernate.jpa.internal.util.LogHelper - HHH000204: Processing PersistenceUnitInfo [
    name: HibernateEjemplo1
    ...]
12:47:18 [main] INFO org.hibernate.Version - HHH000412: Hibernate Core {5.2.10.Final}
12:47:18 [main] INFO org.hibernate.cfg.Environment - HHH000206: hibernate.properties not found
12:47:18 [main] INFO org.hibernate.annotations.common.Version - HCANN000001: Hibernate Commons Annotations {5.0.1.Final}
12:47:19 [main] WARN org.hibernate.orm.connections.pooling - HHH10001002: Using Hibernate built-in connection pool (not for production use!)
12:47:19 [main] INFO org.hibernate.orm.connections.pooling - HHH10001005: using driver [com.mysql.jdbc.Driver] at URL [jdbc:mysql://localhost:3306/sga]
12:47:19 [main] INFO org.hibernate.orm.connections.pooling - HHH10001001: Connection properties: {user=root, password=****}
12:47:19 [main] INFO org.hibernate.orm.connections.pooling - HHH10001003: Autocommit mode: false
12:47:19 [main] INFO org.hibernate.engine.jdbc.connections.internal.DriverManagerConnectionProviderImpl - HHH000115: Hibernate connection pool size: 20 (min=1)
12:47:19 [main] INFO org.hibernate.dialect.Dialect - HHH000400: Using dialect: org.hibernate.dialect.MySQL5Dialect
12:47:19 [main] INFO org.hibernate.engine.jdbc.env.internal.LobCreatorBuilderImpl - HHH000423: Disabling contextual LOB creation as JDBC driver reported JDBC version [3] less than 4
12:47:21 [main] INFO org.hibernate.hql.internal.QueryTranslatorFactoryInitiator - HHH000397: Using ASTQueryTranslatorFactory
12:47:21 [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_
    persona:Persona [apellido=Martinez, idPersona=10, nombre=Juan Carlos]
```


CONCLUSIÓN DEL EJERCICIO

Vimos que utilizar tanto Eclipse como Netbeans 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, o cualquier otro.



Experiencia y Conocimiento para tu vida

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