

CURSO DE HIBERNATE Y JPA

EJERCICIO

LABORATORIO PERSISTENCIA DE ALUMNOS WEB



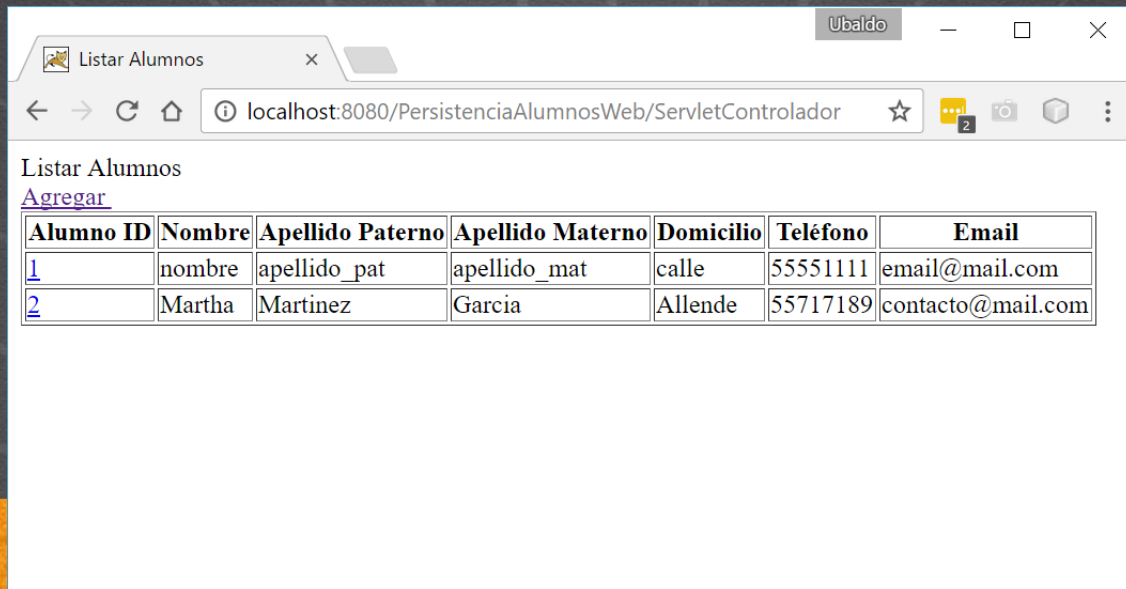
Experiencia y Conocimiento para tu vida

CURSO DE HIBERNATE Y JPA

www.globalmentoring.com.mx

OBJETIVO DEL EJERCICIO

Crear una aplicación Web proyecto para poner el ciclo de vida de los objetos de entidad con Hibernate y JPA. Al finalizar deberemos observar lo siguiente:



The screenshot shows a web browser window with the title 'Ubaldo'. The address bar displays 'localhost:8080/PersistenciaAlumnosWeb/ServletControlador'. The page content includes a link 'Agregar' and a table with student information.

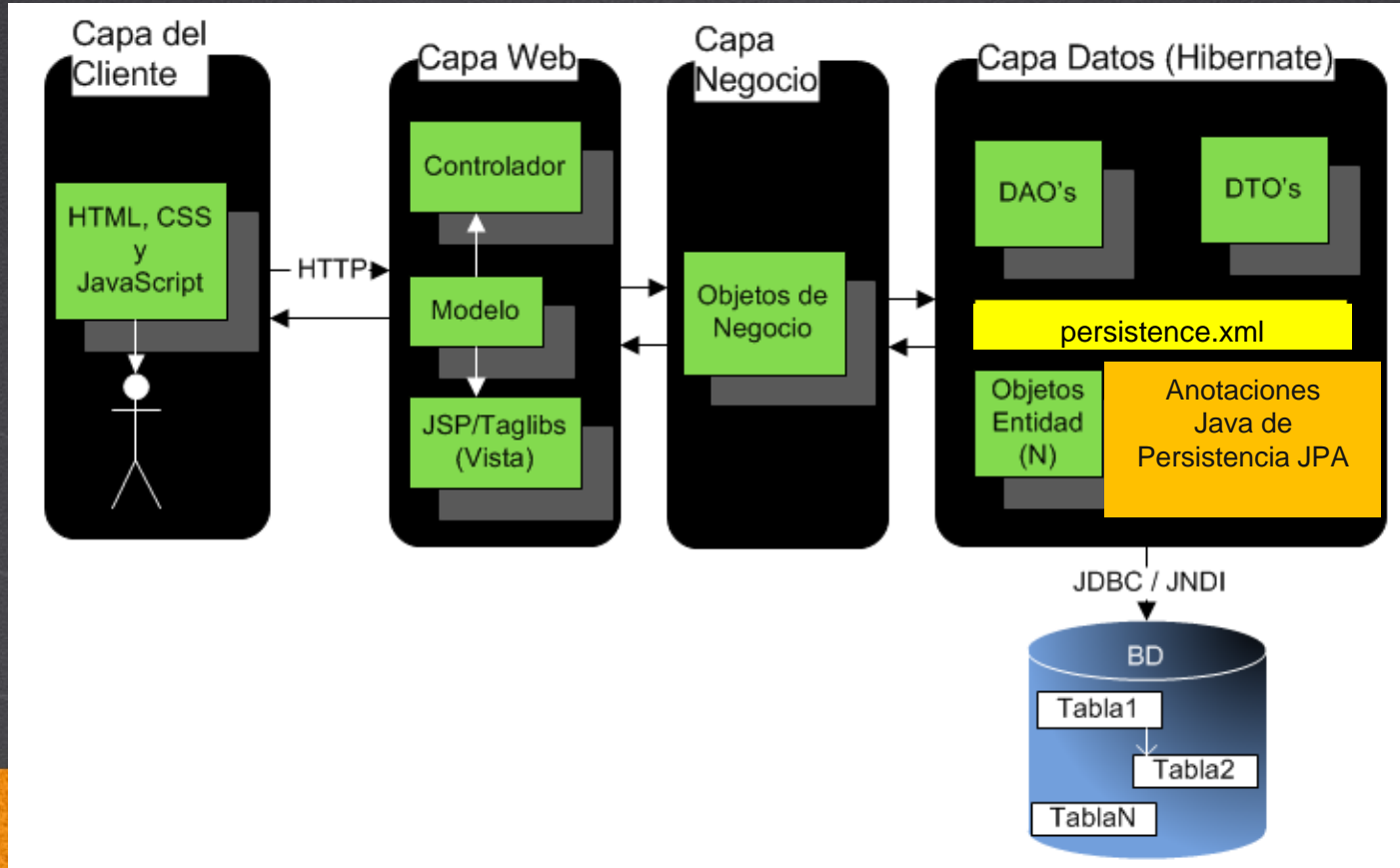
Alumno ID	Nombre	Apellido Paterno	Apellido Materno	Domicilio	Teléfono	Email
1	nombre	apellido_pat	apellido_mat	calle	55551111	email@mail.com
2	Martha	Martinez	Garcia	Allende	55717189	contacto@mail.com

CONTENIDO DEL LABORATORIO

Crear una aplicación Web que contenga lo siguiente:

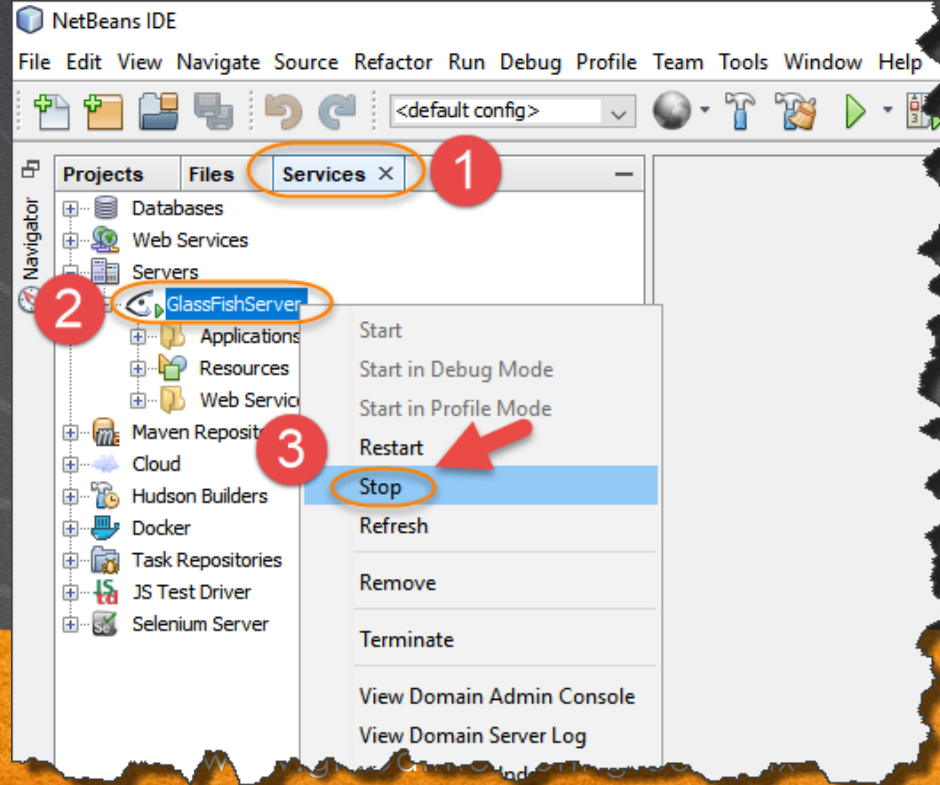
- ✓ Listar el contenido de la tabla de Alumnos.
- ✓ Crear las opciones para agregar, editar y eliminar un nuevo Alumno.
- ✓ En las opciones de agregar y editar se debe crear una pantalla que permita capturar los valores del Alumno junto con su dirección y datos de contacto, tales como teléfono e e-mail, utilizando el concepto de persistencia en Cascada.
- ✓ Utilizar el proyecto de PersistenciaCascada creado anteriormente para utilizar los objetos de Entidad y objetos DAO ya creados en este proyecto.

ARQUITECTURA DEL PROYECTO WEB



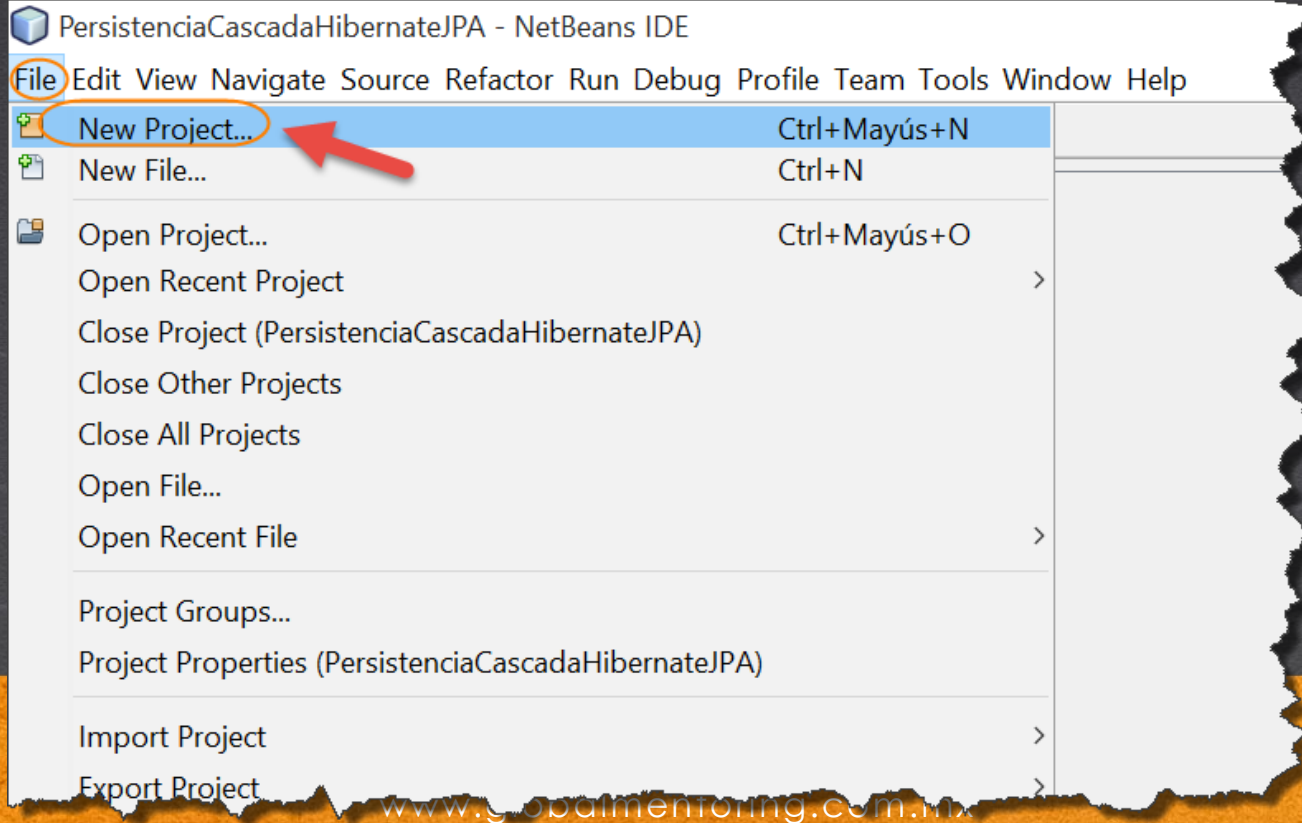
DETENEMOS GLASSFISH SI ESTÁ EJECUTÁNDOSE

Detenemos Glassfish si se estuviera ejecutando:



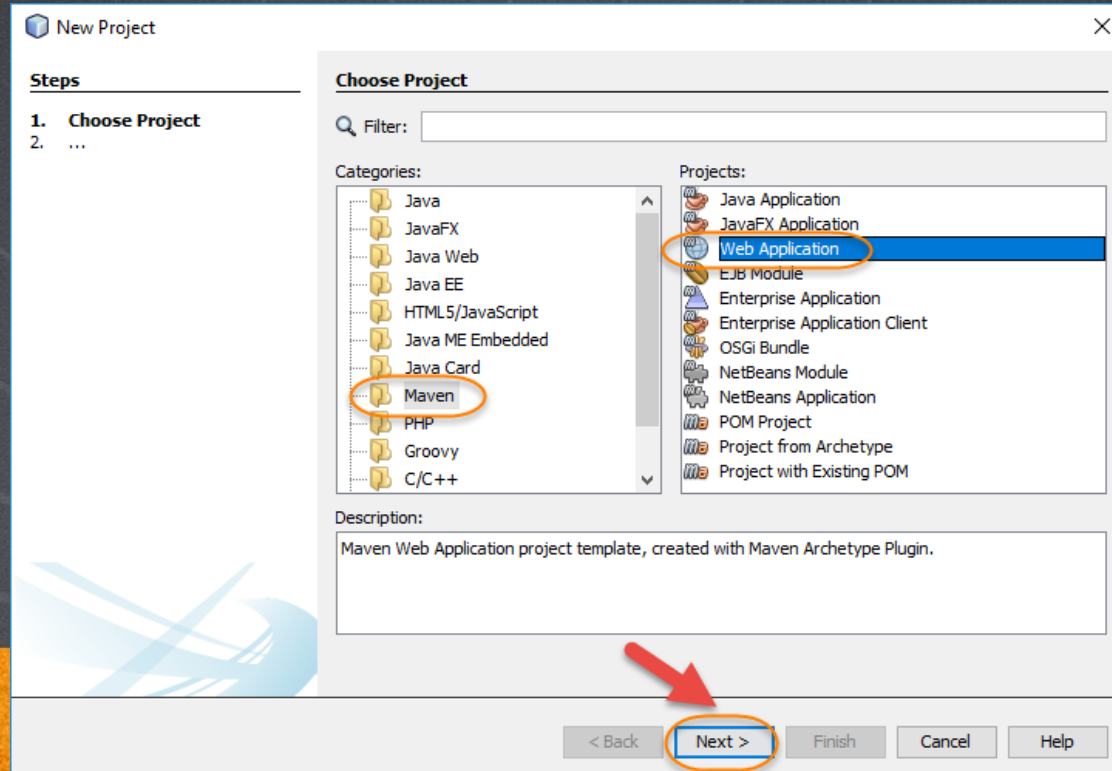
PASO 1. CREAMOS EL PROYECTO

Crear un nuevo proyecto Web:



PASO 1. CREAMOS EL PROYECTO

Crear un nuevo proyecto Web:



PASO 1. CREAMOS EL PROYECTO

Crear un nuevo proyecto Web:

New Web Application

Steps

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

Name and Location

Project Name:

Project Location:

Project Folder:

Artifact Id:

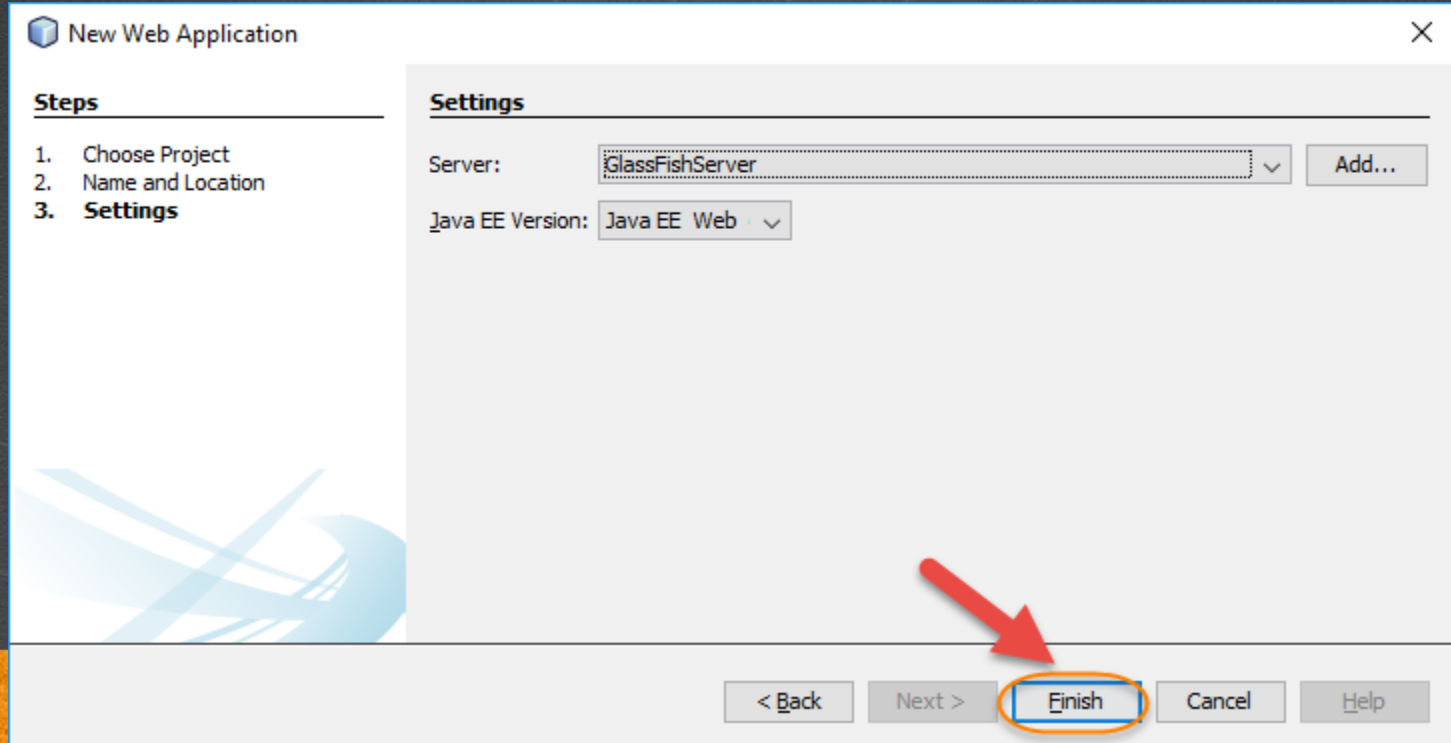
Group Id:

Version:

Package: (Optional)

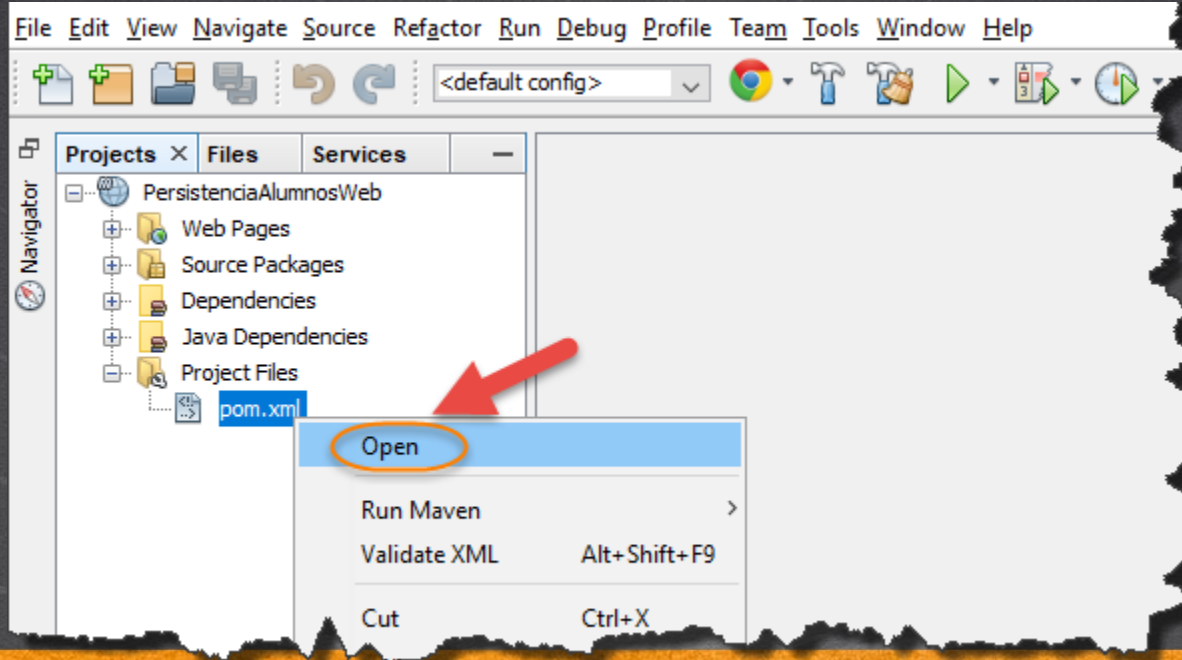
PASO 1. CREAMOS EL PROYECTO

Crear un nuevo proyecto Web:



2. ABRIMOS EL ARCHIVO POM.XML DE MAVEN

- El archivo pom.xml de maven administra las librerías Java vamos a utilizar:



CURSO STRUTS FRAMEWORK

www.globalmentoring.com.mx

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>PersistenciaAlumnosWeb</artifactId>
    <version>1.0</version>
    <packaging>war</packaging>

    <name>PersistenciaAlumnosWeb</name>

    <properties>
        <project.build.sourceEncoding>UTF-8</project.build.sourceEncoding>
    </properties>

    <dependencies>
        <dependency>
            <groupId>javax</groupId>
            <artifactId>javaee-web-api</artifactId>
            <version>8.0</version>
            <scope>provided</scope>
        </dependency>
        <dependency>
            <groupId>org.hibernate</groupId>
            <artifactId>hibernate-core</artifactId>
            <version>5.3.0.CR2</version>
        </dependency>
```


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>
<dependency>
  <groupId>mysql</groupId>
  <artifactId>mysql-connector-java</artifactId>
  <version>5.1.42</version>
</dependency>
</dependencies>
```

CURSO DE JAVA CON JDBC

www.globalmentoring.com.mx

PASO 3. MODIFICAMOS EL CÓDIGO

Archivo pom.xml:

Clic para ver el
archivo

```
<build>
  <plugins>
    <plugin>
      <groupId>org.apache.maven.plugins</groupId>
      <artifactId>maven-war-plugin</artifactId>
      <version>2.3</version>
      <configuration>
        <failOnMissingWebXml>>false</failOnMissingWebXml>
      </configuration>
    </plugin>
    <plugin>
      <groupId>org.apache.maven.plugins</groupId>
      <artifactId>maven-compiler-plugin</artifactId>
      <version>3.7.0</version>
      <configuration>
        <source>1.8</source>
        <target>1.8</target>
      </configuration>
    </plugin>
  </plugins>
</build>

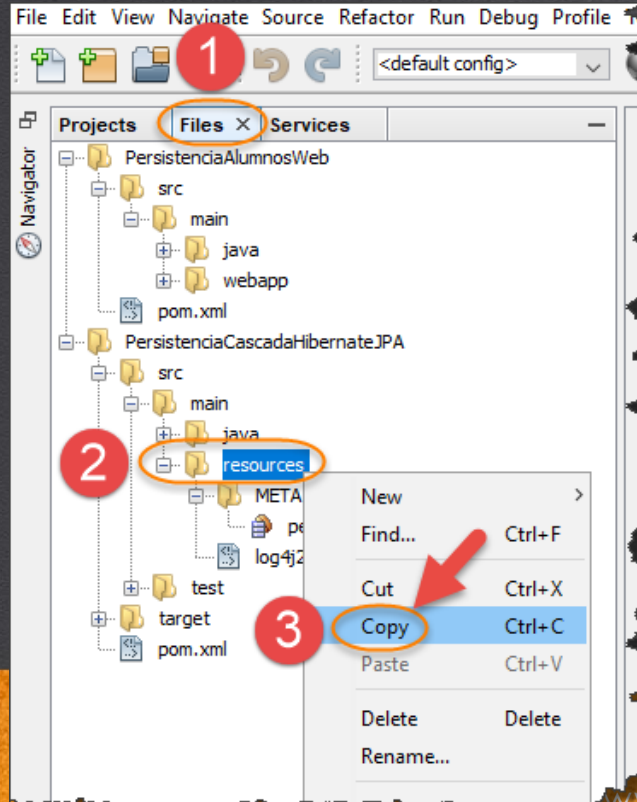
</project>
```

CURSO DE JAVA CON JDBC

www.globalmentoring.com.mx

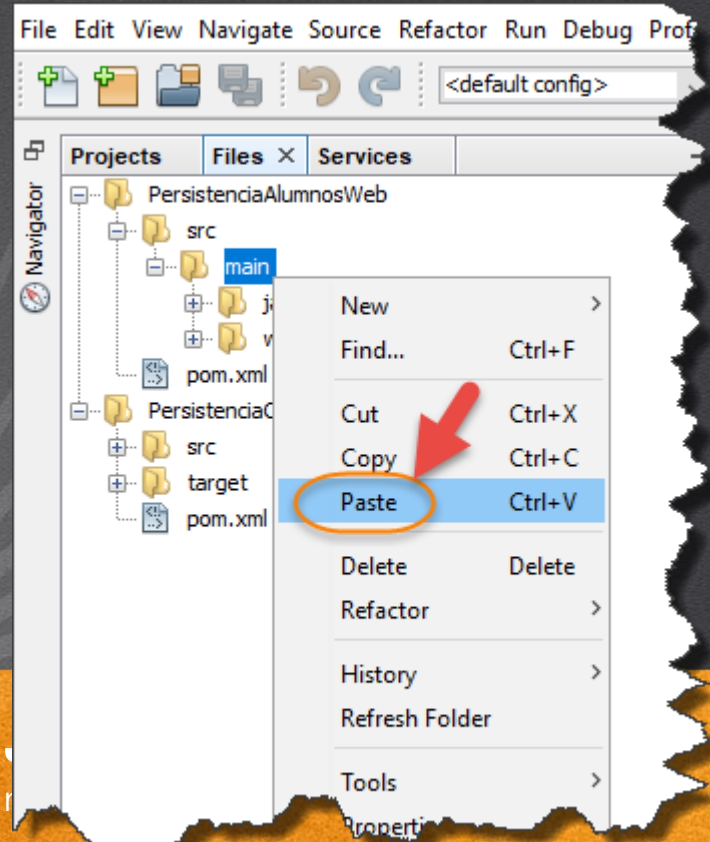
PASO 4. COPIAR CLASES Y ARCHIVOS

Copiamos el archivo log4j2.xml y META-INF\persistence.xml



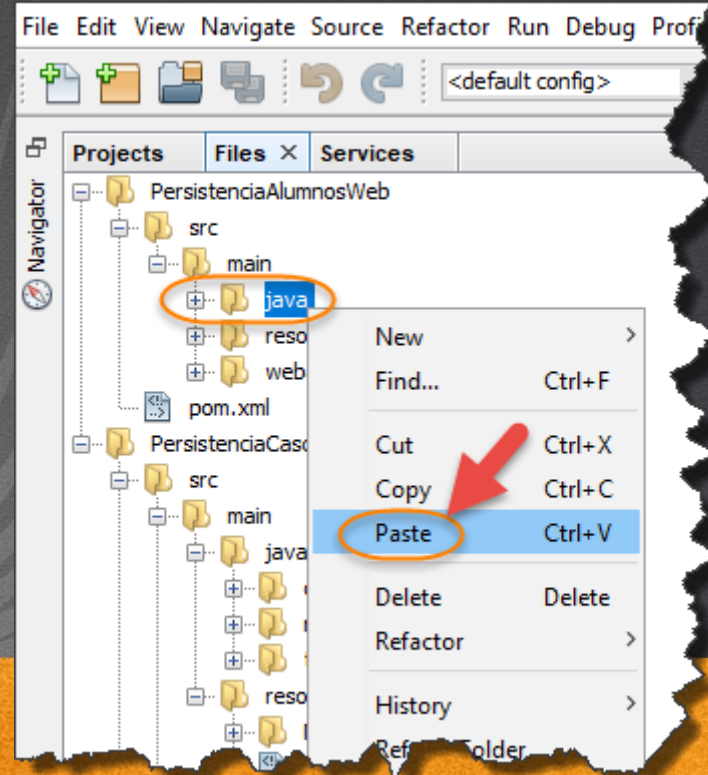
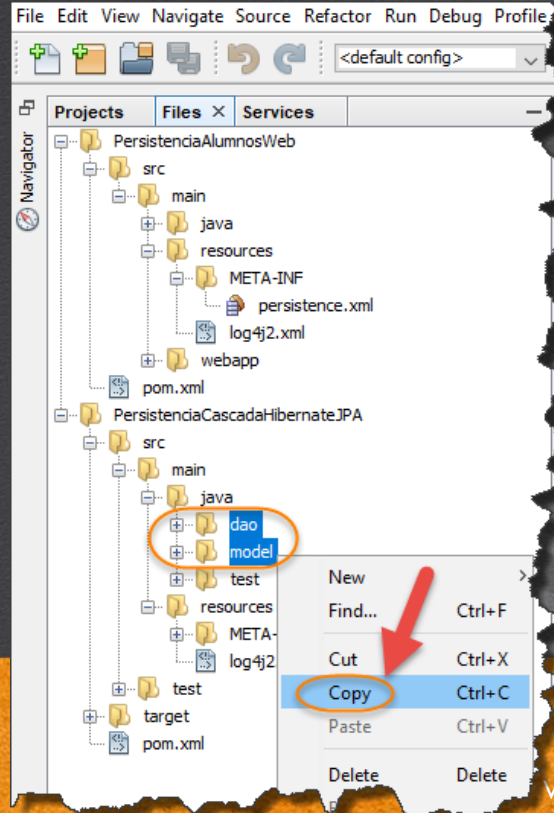
CURSO HIBERNATE Y

www.globalmentoring.com



PASO 5. COPIAMOS LAS CLASES DE MODELO

Copiamos las clases del paquete model y dao:

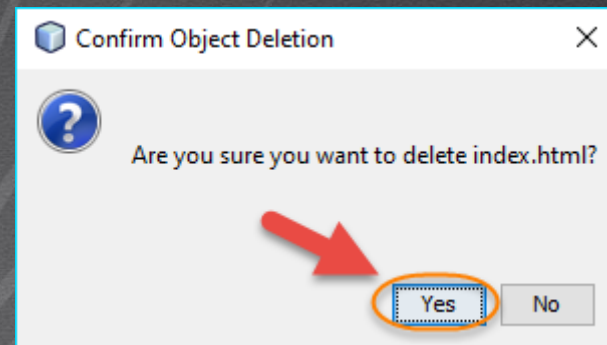
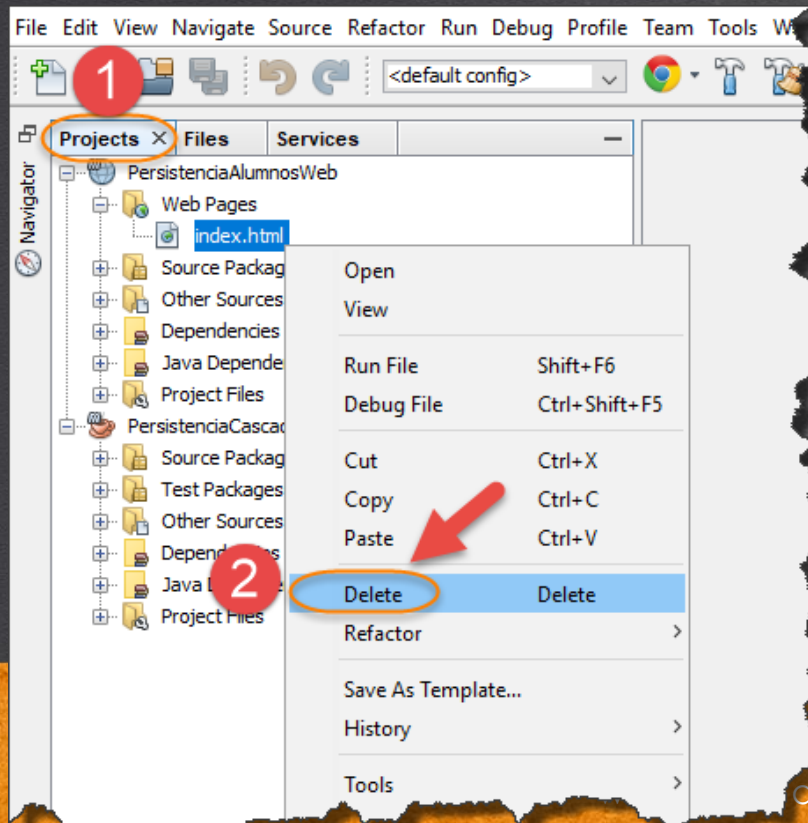


CURSO HIBERNATE

www.globalmentoring.com.mx

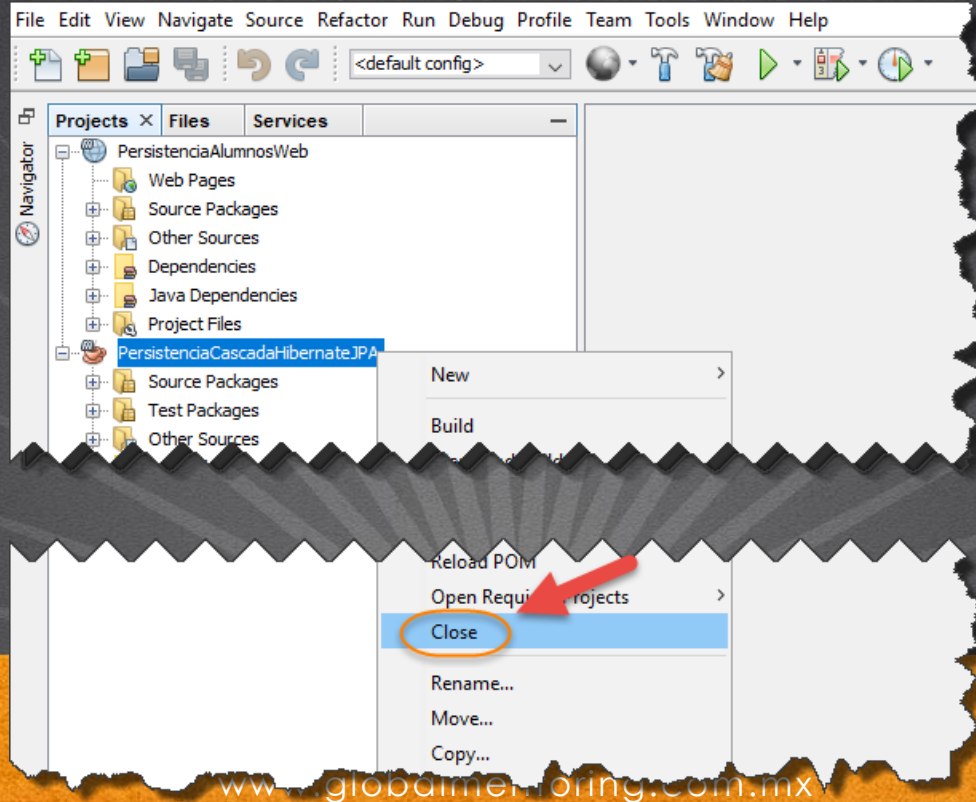
PASO 6. ELIMINAR EL ARCHIVO INDEX.HTML

Eliminamos el archivo index.html si existe:



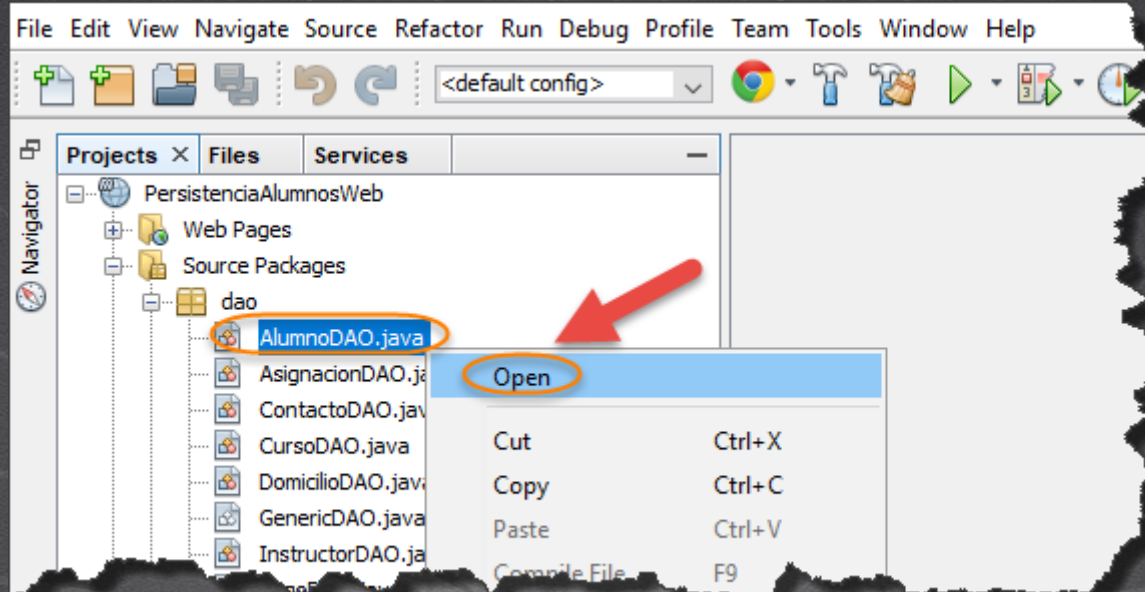
PASO 7. CERRAMOS EL PROYECTO QUE NO USEMOS

Cerramos el proyecto que ya no usamos:



PASO 8. CREAR UNA CLASE

Modificamos la clase AlumnosDAO.java



PASO 8. MODIFICAMOS EL CÓDIGO

Archivo AlumnoDao.java:

Dar click para ir al código

```
package dao;

import java.util.List;
import javax.persistence.Query;
import model.Alumno;

public class AlumnoDAO extends GenericDAO {

    public void listar() {
        // Consulta a ejecutar
        // No necesitamos crear una nueva transaccion
        String hql = "SELECT a FROM Alumno a";
        em = getEntityManager();
        Query query = em.createQuery(hql);
        List<Alumno> list = query.getResultList();
        for (Alumno a : list) {
            System.out.println(a);
        }
    }

    public List<Alumno> listarAlumnos() {
        String hql = "SELECT a FROM Alumno a";
        em = getEntityManager();
        Query query = em.createQuery(hql);
        return query.getResultList();
    }
}
```

PASO 8. MODIFICAMOS EL CÓDIGO

Archivo AlumnoDao.java:

Dar click para ir al código

```
public void insertar(Alumno alumno) {  
    try {  
        em = getEntityManager();  
        // Iniciamos una transaccion  
        em.getTransaction().begin();  
        // Insertamos nuevo objeto  
        em.persist(alumno);  
        // Terminamos la transaccion  
        em.getTransaction().commit();  
    } catch (Exception ex) {  
        System.out.println("Error al insetar objeto:" + ex.getMessage());  
        ex.printStackTrace(System.out);  
    }  
}
```


PASO 8. MODIFICAMOS EL CÓDIGO

Archivo AlumnoDao.java:

Dar click para ir al código

```
public void actualizar(Alumno alumno) {  
    try {  
        em = getEntityManager();  
        // Iniciamos una transaccion  
        em.getTransaction().begin();  
        // Actualizamos al objeto  
        em.merge(alumno);  
        // Terminamos la transaccion  
        em.getTransaction().commit();  
    } catch (Exception ex) {  
        System.out.println("Error al actualizar objeto:" + ex.getMessage());  
        ex.printStackTrace(System.out);  
    }  
}
```

CURSO HIBERNATE Y JPA

www.globalmentoring.com.mx

PASO 8. MODIFICAMOS EL CÓDIGO

Archivo AlumnoDao.java:

Dar click para ir al código

```
public void eliminar(Alumno alumno) {
    try {
        em = getEntityManager();
        // Iniciamos una transaccion
        em.getTransaction().begin();
        // Sincronizamos y eliminamos
        em.remove(em.merge(alumno));
        // Terminamos la transaccion
        em.getTransaction().commit();
    } catch (Exception ex) {
        System.out.println("Error al eliminar objeto:" + ex.getMessage());
        ex.printStackTrace(System.out);
    }
}

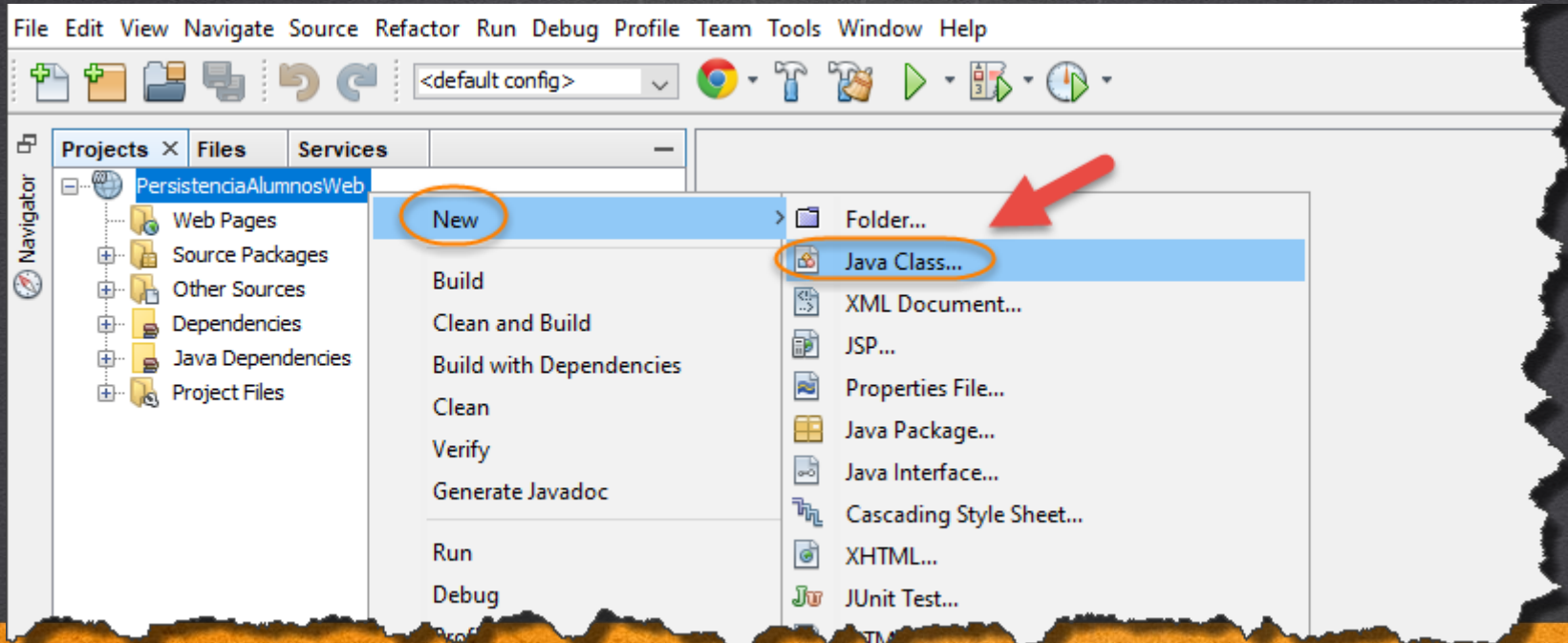
public Alumno buscarPorId(Alumno alumno) {
    em = getEntityManager();
    return em.find(Alumno.class, alumno.getIdAlumno());
}
}
```

CURSO HIBERNATE Y JPA

www.globalmentoring.com.mx

PASO 9. CREAR UNA CLASE

Creamos la clase ServicioAlumnos.java



PASO 9. CREAR UNA CLASE

Creamos la clase ServicioAlumnos.java

New Java Class

Steps

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

Name and Location

Class Name:

Project:

Location:

Package:

Created File:

PASO 10. MODIFICAMOS EL CÓDIGO

Archivo ServicioAlumnos.java:

Dar click para ir al código

```
package servicio;

import dao.AlumnoDAO;
import model.Alumno;
import java.util.List;

public class ServicioAlumnos {

    // El Dao se asocia con la clase
    AlumnoDAO alumnoDao = new AlumnoDAO();

    // Esta clase se comunica con los DAO's que necesite
    public List<Alumno> listarAlumnos() {
        return alumnoDao.listarAlumnos();
    }
}
```

PASO 10. MODIFICAMOS EL CÓDIGO

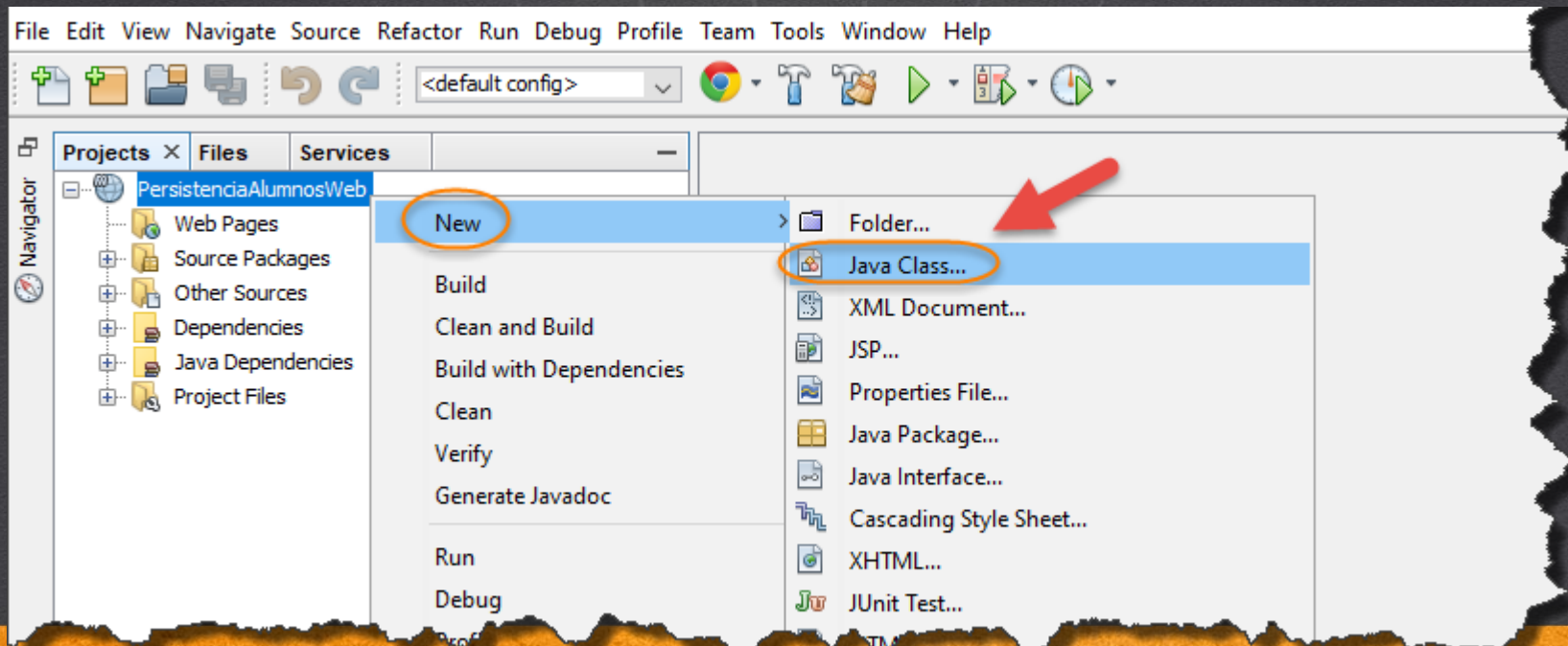
Archivo ServicioAlumnos.java:

Dar click para ir al código

```
public boolean guardarAlumno(Alumno alumno) {  
    if(alumno != null && alumno.getIdAlumno() == null){  
        //insert  
        alumnoDao.insertar(alumno);  
    }  
    else{  
        //update  
        alumnoDao.actualizar(alumno);  
    }  
    return true;// si nada fallo, regresamos verdadero  
}  
  
public boolean eliminarAlumno(Integer idAlumno) {  
    //delete  
    alumnoDao.eliminar(new Alumno(idAlumno));  
    return true;// si nada falla regresamos verdadero  
}  
  
public Alumno encontrarAlumno(Integer idAlumno) {  
    return alumnoDao.buscarPorId(new Alumno(idAlumno));  
}  
}
```


PASO 11. CREAR UNA CLASE

Creamos la clase ServletControlador.java:



CURSO HIBERNATE Y JPA

www.globalmentoring.com.mx

PASO 11. CREAR UNA CLASE

Creamos la clase ServletControlador.java:

New Java Class

Steps

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

Name and Location

Class Name:

Project:

Location:

Package:

Created File:

PASO 12. MODIFICAMOS EL CÓDIGO

Archivo ServletControlador.java:

Dar click para ir al código

```
package web;

import java.io.IOException;
import java.util.List;
import javax.servlet.ServletException;
import javax.servlet.annotation.WebServlet;
import javax.servlet.http.*;
import model.Alumno;
import servicio.ServicioAlumnos;

@WebServlet(name = "ServletControlador", urlPatterns = {"/ServletControlador"})
public class ServletControlador extends HttpServlet {

    @Override
    protected void doGet(HttpServletRequest request, HttpServletResponse response)
        throws ServletException, IOException {
        //1. Nos comunicamos con la capa de servicio
        ServicioAlumnos servicioAlumnos = new ServicioAlumnos();

        //2. Recuperamos todos los alumnos
        List<Alumno> alumnos = servicioAlumnos.listarAlumnos();

        //3. Compartimos la informacion (Modelo) con la vista
        request.setAttribute("alumnos", alumnos);

        //4. Seleccionamos la vista a mostrar la info de alumnos
        request.getRequestDispatcher("/WEB-INF/listarAlumnos.jsp").forward(request, response);
    }
}
```


PASO 12. MODIFICAMOS EL CÓDIGO

Archivo ServletControlador.java:

Dar click para ir al código

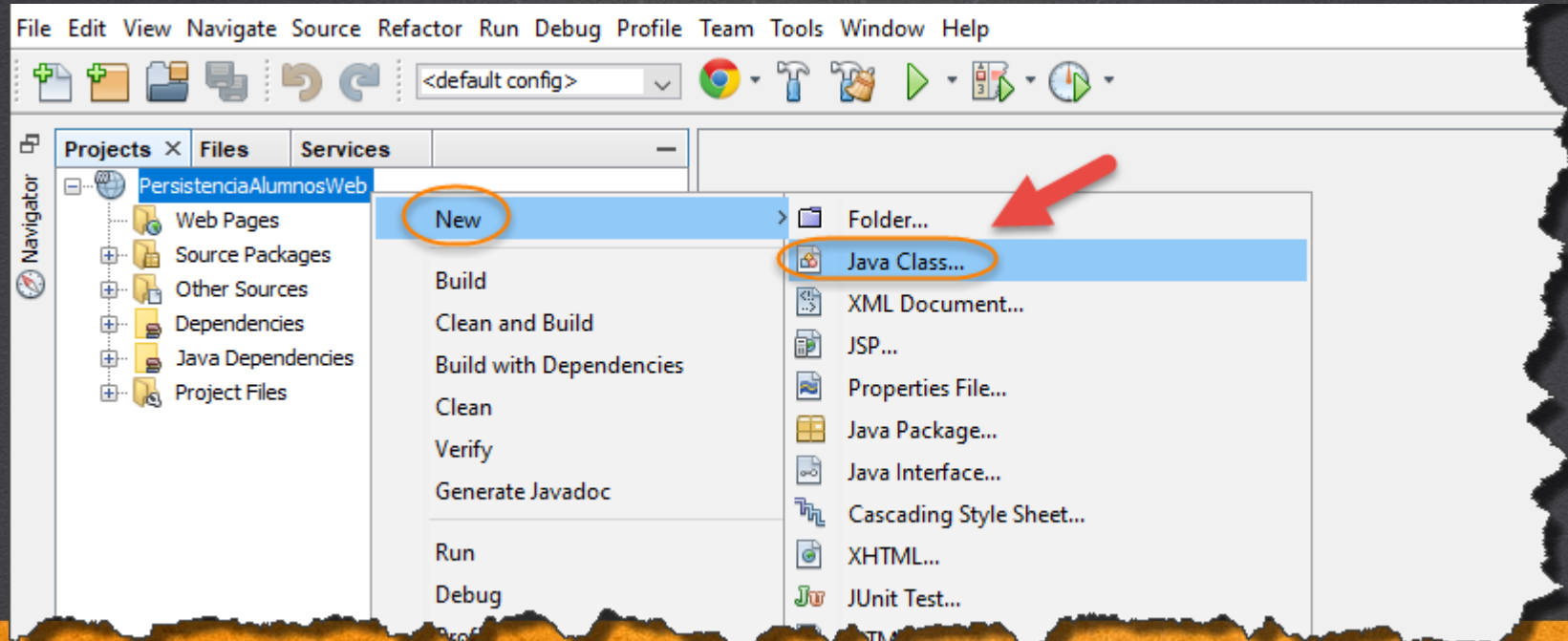
```
@Override
public void doPost(HttpServletRequest request, HttpServletResponse response)
    throws ServletException, IOException {
    this.doGet(request, response);
}
}
```

CURSO HIBERNATE Y JPA

www.globalmentoring.com.mx

PASO 13. CREAR UNA CLASE

Creamos la clase ServletAgregar.java:



CURSO HIBERNATE Y JPA

www.globalmentoring.com.mx

PASO 13. CREAR UNA CLASE

Creamos la clase ServletAgregar.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 14. MODIFICAMOS EL CÓDIGO

Archivo ServletAgregar.java:

Dar click para ir al código

```
package web;

import java.io.IOException;
import javax.servlet.ServletException;
import javax.servlet.annotation.WebServlet;
import javax.servlet.http.*;
import model.*;
import servicio.ServicioAlumnos;

@WebServlet(name = "ServletAgregar", urlPatterns = {"/ServletAgregar"})
public class ServletAgregar extends HttpServlet {

    @Override
    protected void doPost(HttpServletRequest request, HttpServletResponse response)
        throws ServletException, IOException {

        //1. Recuperamos los parametros del formulario
        String nombre = request.getParameter("nombre");
        String apellidoPaterno = request.getParameter("apellidoPaterno");
        String apellidoMaterno = request.getParameter("apellidoMaterno");
        String calle = request.getParameter("calle");
        String email = request.getParameter("email");
        String telefono = request.getParameter("telefono");
```

PASO 14. MODIFICAMOS EL CÓDIGO

Archivo ServletAgregar.java:

Dar click para ir al código

```
//2. Llenamos el objeto DTO de Alumno a insertar
//Creamos el objeto Domicilio
Domicilio domicilio = new Domicilio();
domicilio.setCalle(calle);
domicilio.setVersion(0);
domicilio.setDeleted(0);

//Creamos el objeto de Contacto
Contacto contacto = new Contacto();
contacto.setEmail(email);
contacto.setTelefono(telefono);
contacto.setVersion(0);
contacto.setDeleted(0);

Alumno alumno = new Alumno();
alumno.setNombre(nombre);
alumno.setApellidoPaterno(apellidoPaterno);
alumno.setApellidoMaterno(apellidoMaterno);
alumno.setVersion(0); //llenamos con valores por default
alumno.setDeleted(0); //estos son campos requeridos
//Inyectamos las dependencias de los objetos Domicilio y Contacto
alumno.setDomicilio(domicilio);
alumno.setContacto(contacto);
```

PASO 14. MODIFICAMOS EL CÓDIGO

Archivo ServletAgregar.java:

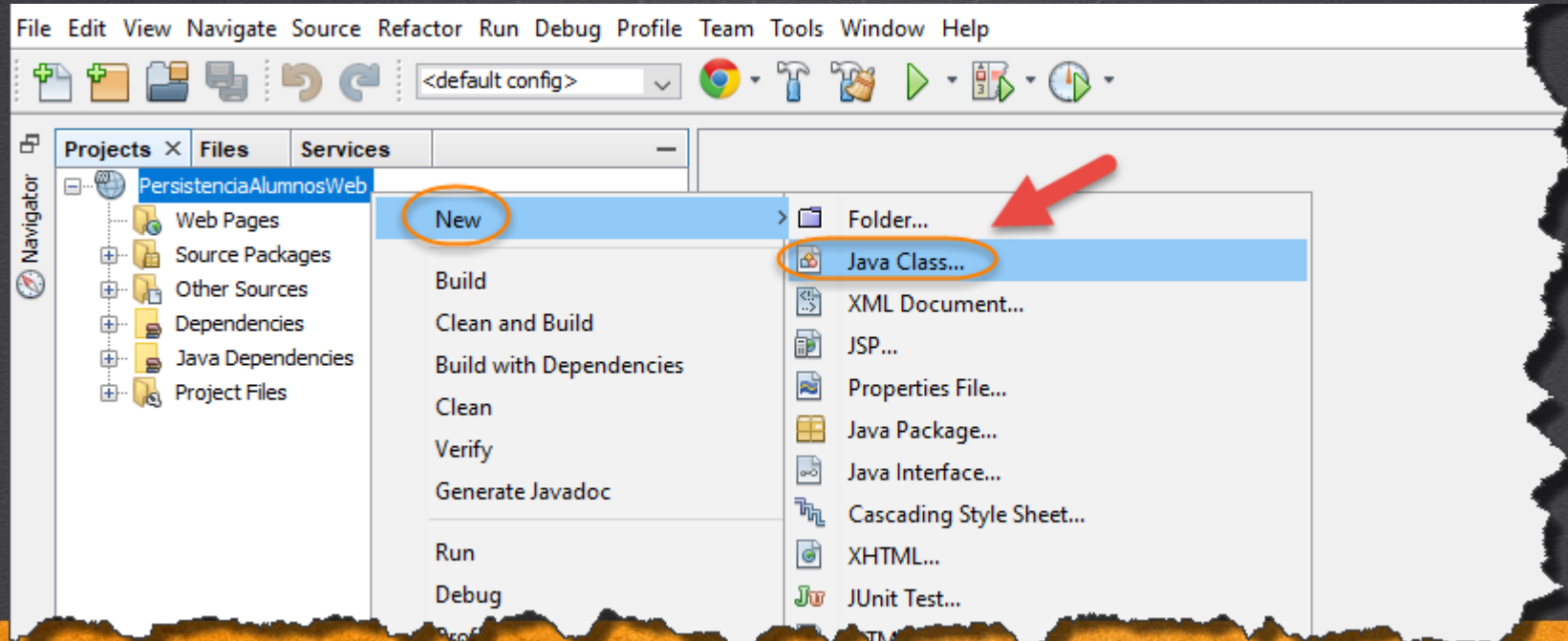
Dar click para ir al código

```
//3. Nos apoyamos de la capa de servicio
ServicioAlumnos servicioAlumnos = new ServicioAlumnos();
servicioAlumnos.guardarAlumno(alumno);

//4. Redireccionamos a la pagina de inicio
//ya que de lo contrario tendríamos q volver a cargar la lista de alumnos
request.getRequestDispatcher("/index.jsp").forward(request, response);
}
}
```

PASO 15. CREAR UNA CLASE

Creamos la clase ServletModificar.java:



CURSO HIBERNATE Y JPA

www.globalmentoring.com.mx

PASO 15. CREAR UNA CLASE

Creamos la clase ServletModificar.java:

New Java Class

Steps

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

Name and Location

Class Name:

Project:

Location:

Package:

Created File:

PASO 16. MODIFICAMOS EL CÓDIGO

Archivo ServletModificar.java:

Dar click para ir al código

```
package web;

import java.io.IOException;
import javax.servlet.ServletException;
import javax.servlet.annotation.WebServlet;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.*;
import model.Alumno;
import servicio.ServicioAlumnos;

@WebServlet(name = "ServletModificar", urlPatterns = {"/ServletModificar"})
public class ServletModificar extends HttpServlet {

    @Override
    public void doGet(HttpServletRequest request, HttpServletResponse response)
        throws ServletException, IOException {

        // Este metodo get se va a encargar de recuperar
        // el objeto a modificar para mostrarlo en el formulario
        //1. Recuperamos el idAlumno de la peticion
        String idAlumno = request.getParameter("idAlumno");
```

PASO 16. MODIFICAMOS EL CÓDIGO

Archivo ServletModificar.java:

Dar click para ir al código

```
//2. Nos apoyamos de la clase de servicio para recuperar
//el objeto a modificar
ServicioAlumnos servicioAlumnos = new ServicioAlumnos();
Alumno alumno = servicioAlumnos.encontrarAlumno(Integer.parseInt(idAlumno));

//3. Compartimos el objeto encontrado con la vista
//Vamos a compartirlo en la session
HttpSession session = request.getSession();
session.setAttribute("alumno", alumno);

//4. Redireccionamos a la vista de modificar
request.getRequestDispatcher("/WEB-INF/modificarAlumno.jsp").forward(request, response);
}
```

PASO 16. MODIFICAMOS EL CÓDIGO

Archivo ServletModificar.java:

Dar click para ir al código

```
@Override
public void doPost(HttpServletRequest request, HttpServletResponse response)
    throws ServletException, IOException {

    // Este metodo post se va encargar de modificar
    //los valores que se envien del formulario
    //Revisamos si se presiono el boton de modificar o eliminar
    String accion = request.getParameter("modificar");
    if ("modificar".equals(accion)) {

        //1. Recuperamos los parametros
        String nombre = request.getParameter("nombre");
        String apellidoPaterno = request.getParameter("apellidoPaterno");
        String apellidoMaterno = request.getParameter("apellidoMaterno");
        String calle = request.getParameter("calle");
        String email1 = request.getParameter("email");
        String telefono = request.getParameter("telefono");

        //2. Recuperamos el objeto de la sesion
        HttpSession session = request.getSession();
        Alumno alumno = (Alumno) session.getAttribute("alumno");
```


PASO 16. MODIFICAMOS EL CÓDIGO

Archivo ServletModificar.java:

Dar click para ir al código

```
//3. Actualizamos los valores de la modificacion del alumno
alumno.setNombre(nombre);
alumno.setApellidoPaterno(apellidoPaterno);
alumno.setApellidoMaterno(apellidoMaterno);
//Actualizamos los valores de los objetos embebidos
//de contacto y domicilio
alumno.getDomicilio().setCalle(calle);
alumno.getContacto().setEmail(email);
alumno.getContacto().setTelefono(telefono);

//4. Nos comunicamos con la capa de servicio para guardar
//los cambios
ServicioAlumnos servicioAlumnos = new ServicioAlumnos();
servicioAlumnos.guardarAlumno(alumno);

//5. Eliminamos el objeto modificado de la sesion
//ya que solo lo utilizamos mientras modificabamos el alumno
session.removeAttribute("alumno");

} //else if("eliminar".equals( accion )){
```

CURSO HIBERNATE Y JPA

www.globalmentoring.com.mx

PASO 16. MODIFICAMOS EL CÓDIGO

Archivo ServletModificar.java:

Dar click para ir al código

```
else { //se selecciono eliminar
    //Eliminamos el objeto de alumno

    //1. Recuperamos el id del alumno a eliminar
    String idAlumno = request.getParameter("idAlumno");

    //2. Nos apoyamos de la capa de servicio
    //para eliminar el objeto de Alumno
    ServicioAlumnos servicioAlumnos = new ServicioAlumnos();
    servicioAlumnos.eliminarAlumno(Integer.parseInt(idAlumno));
}

//6.Redireccionamos a la pagina de inicio
request.getRequestDispatcher("/index.jsp").forward(request, response);
}

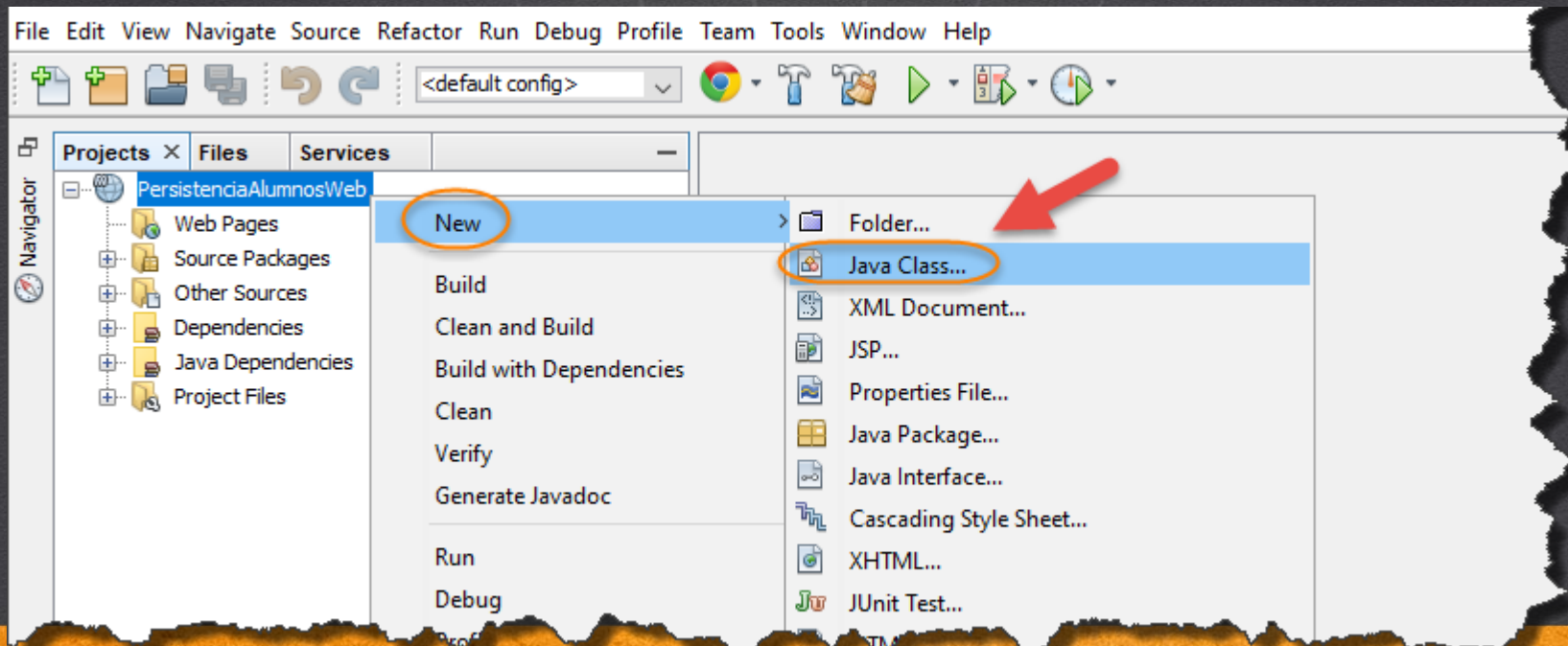
}
```

CURSO HIBERNATE Y JPA

www.globalmentoring.com.mx

PASO 17. CREAR UNA CLASE

Creamos la clase ServletRedireccionar.java:



CURSO HIBERNATE Y JPA

www.globalmentoring.com.mx

PASO 17. CREAR UNA CLASE

Creamos la clase ServletRedireccionar.java:

New Java Class

Steps

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

Name and Location

Class Name:

Project:

Location:

Package:

Created File:

PASO 18. MODIFICAMOS EL CÓDIGO

[Archivo ServletRedireccionar.java:](#)

Dar click para ir al código

```
package web;

import java.io.IOException;
import javax.servlet.ServletException;
import javax.servlet.annotation.WebServlet;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;

@WebServlet(name = "ServletRedireccionar", urlPatterns = {"/ServletRedireccionar"})
public class ServletRedireccionar extends HttpServlet {

    @Override
    public void doGet(HttpServletRequest request, HttpServletResponse response)
        throws ServletException, IOException {

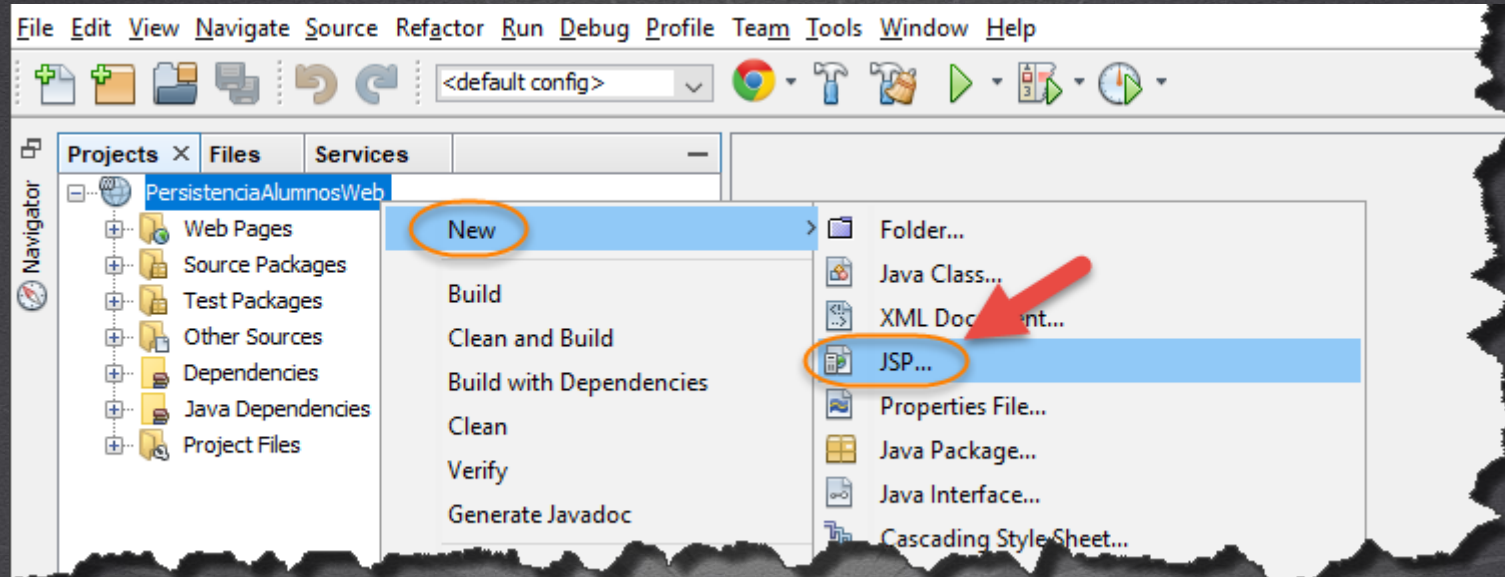
        //Redireccionamos a la pagina de Agregar Alumno
        request.getRequestDispatcher("WEB-INF/agregarAlumno.jsp").forward(request, response);
    }
}
```

CURSO HIBERNATE Y JPA

www.globalmentoring.com.mx

PASO 19. CREAR UN ARCHIVO JSP

Creamos el archivo index.jsp



CURSO HIBERNATE Y JPA

www.globalmentoring.com.mx

PASO 19. CREAR UN ARCHIVO JSP

Creamos el archivo index.jsp

New JSP

Steps

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

Name and Location

File Name:

Project:

Location:

Folder:

Created File:

Options:

☒ JSP File (Standard Syntax) ☐ Create as a JSP Segment

☐ JSP Document (XML Syntax)

Description:

A JSP file using JSP standard syntax.

PASO 20. MODIFICAMOS EL CÓDIGO

Archivo index.jsp:

Dar click para ir al código

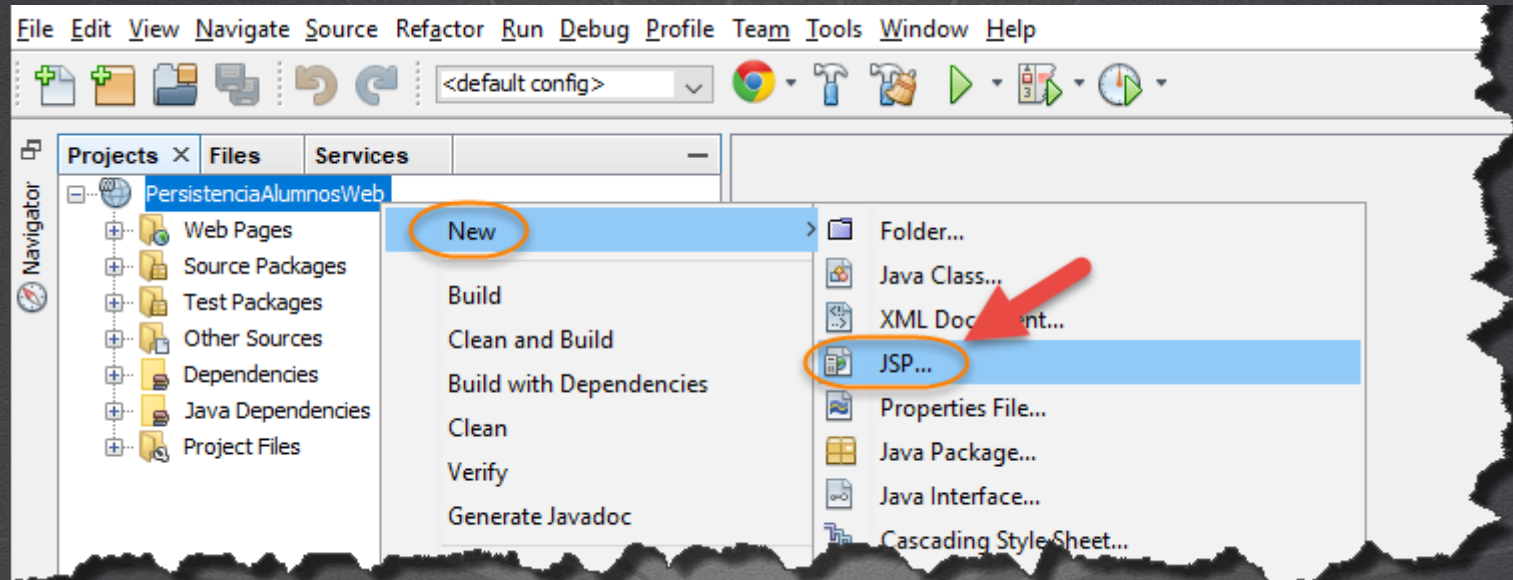
```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="UTF-8">
    <title>Inicio</title>
  </head>
  <body>
    <a href="${pageContext.request.contextPath}/ServletControlador">
      Cat&aacute;logo de Alumnos
    </a>
  </body>
</html>
```

CURSO HIBERNATE Y JPA

www.globalmentoring.com.mx

PASO 21. CREAR UN ARCHIVO JSP

Creamos el archivo listarAlumnos.jsp



CURSO HIBERNATE Y JPA

www.globalmentoring.com.mx

PASO 21. CREAR UN ARCHIVO JSP

Creamos el archivo listarAlumnos.jsp

New JSP

Steps

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

Name and Location

File Name:

Project:

Location:

Folder:

Created File:

Options:

☒ JSP File (Standard Syntax) ☐ Create as a JSP Segment

☐ JSP Document (XML Syntax)

Description:

PASO 22. MODIFICAMOS EL CÓDIGO

Archivo listarAlumnos.jsp:

Dar click para ir al código

```
<%@taglib uri="http://java.sun.com/jsp/jstl/core" prefix="c" %>

<!DOCTYPE html>
<html>
  <head>
    <meta charset="UTF-8">
    <title>Listar Alumnos</title>
  </head>
  <body>
    Listar Alumnos
    <br/>
    <a href="{pageContext.request.contextPath}/ServletRedireccionar">
      Agregar
    </a>
    <br/>
```


PASO 22. MODIFICAMOS EL CÓDIGO

[Archivo listarAlumnos.jsp:](#)

Dar click para ir al código

```
<table border="1">
  <tr>
    <th>Alumno ID</th>
    <th>Nombre</th>
    <th>Apellido Paterno</th>
    <th>Apellido Materno</th>
    <th>Domicilio</th>
    <th>Tel&eacute;fono</th>
    <th>Email</th>
  </tr>
```


PASO 22. MODIFICAMOS EL CÓDIGO

Archivo listarAlumnos.jsp:

Dar click para ir al código

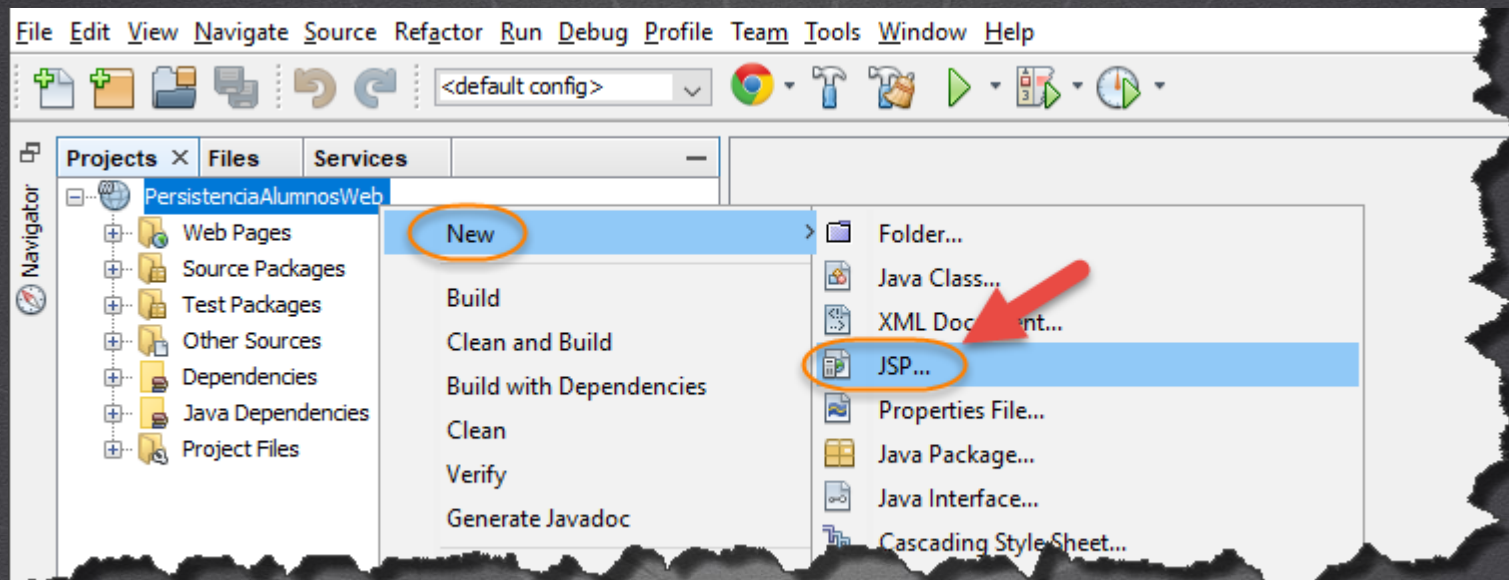
```
<c:forEach var="alumno" items="${alumnos}">
  <tr>
    <td>
      <a
href="${pageContext.request.contextPath}/ServletModificar?idAlumno=${alumno.idAlumno}">
        ${alumno.idAlumno}
      </a>
    </td>
    <td>${alumno.nombre}</td>
    <td>${alumno.apellidoPaterno }</td>
    <td>${alumno.apellidoMaterno }</td>
    <td>${alumno.domicilio.calle }</td>
    <td>${alumno.contacto.telefono }</td>
    <td>${alumno.contacto.email1 }</td>
  </tr>
</c:forEach>
</table>
</body>
</html>
```

CURSO HIBERNATE Y JPA

www.globalmentoring.com.mx

PASO 23. CREAR UN ARCHIVO JSP

Creamos el archivo agregarAlumno.jsp



CURSO HIBERNATE Y JPA

www.globalmentoring.com.mx

PASO 23. CREAR UN ARCHIVO JSP

Creamos el archivo agregarAlumno.jsp

New JSP

Steps

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

Name and Location

File Name:

Project:

Location:

Folder:

Created File:

Options:

☒ JSP File (Standard Syntax) ☐ Create as a JSP Segment

☐ JSP Document (XML Syntax)

Description:

PASO 24. MODIFICAMOS EL CÓDIGO

Archivo agregarAlumno.jsp:

Dar click para ir al código

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="UTF-8">
    <title>Agregar Alumno</title>
  </head>
  <body>
    Agregar Alumno
    <br/>
    <form name ="form1" action="${pageContext.request.contextPath}/ServletAgregar" method="post">
      Nombre: <input type="text" name="nombre" >
      <br/>
      Apellido Paterno: <input type="text" name="apellidoPaterno">
      <br/>
      Apellido Materno: <input type="text" name="apellidoMaterno">
      <br/>
      <br/>
    </form>
  </body>
</html>
```

CURSO HIBERNATE Y JPA

www.globalmentoring.com.mx

PASO 24. MODIFICAMOS EL CÓDIGO

Archivo agregarAlumno.jsp:

Dar click para ir al código

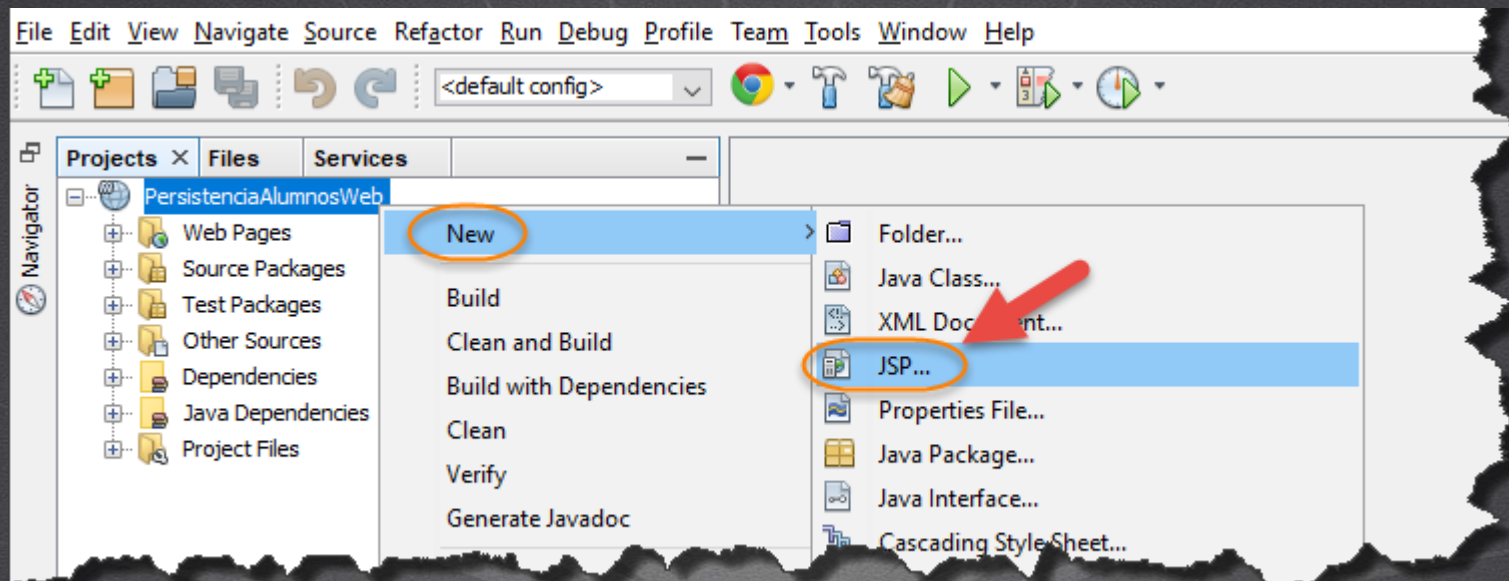
```
Datos Domicilio:
<br/>
Calle: <input type="text" name="calle">
<br/>
<br/>
Datos Contacto:
<br/>
email: <input type="text" name="email">
<br/>
telefono: <input type="text" name="telefono">
<br/>
<input type="submit" name="agregar" value="Agregar">
</form>
</body>
</html>
```

CURSO HIBERNATE Y JPA

www.globalmentoring.com.mx

PASO 25. CREAR UN ARCHIVO JSP

Creamos el archivo modificarAlumno.jsp



CURSO HIBERNATE Y JPA

www.globalmentoring.com.mx

PASO 25. CREAR UN ARCHIVO JSP

Creamos el archivo modificarAlumno.jsp

New JSP

Steps

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

Name and Location

File Name:

Project:

Location:

Folder:

Created File:

Options:

☒ JSP File (Standard Syntax) ☐ Create as a JSP Segment

☐ JSP Document (XML Syntax)

Description:

PASO 26. MODIFICAMOS EL CÓDIGO

Archivo modificarAlumno.jsp:

Dar click para ir al código

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="UTF-8">
    <title>Modificar Alumno</title>
  </head>
  <body>
    Modificar Alumno
    <br/>
    <form name ="form1" action="${pageContext.request.contextPath}/ServletModificar" method="post">
      Id Alumno: <input type="text" name="idAlumno" value="${alumno.idAlumno}" readonly="readonly">
      <br/>
      Nombre: <input type="text" name="nombre" value="${alumno.nombre}">
      <br/>
      Apellido Paterno: <input type="text" name="apellidoPaterno" value="${alumno.apellidoPaterno}">
      <br/>
      Apellido Materno: <input type="text" name="apellidoMaterno" value="${alumno.apellidoMaterno}">
      <br/>
```

CURSO HIBERNATE Y JPA

www.globalmentoring.com.mx

PASO 26. MODIFICAMOS EL CÓDIGO

Archivo modificarAlumno.jsp:

Dar click para ir al código

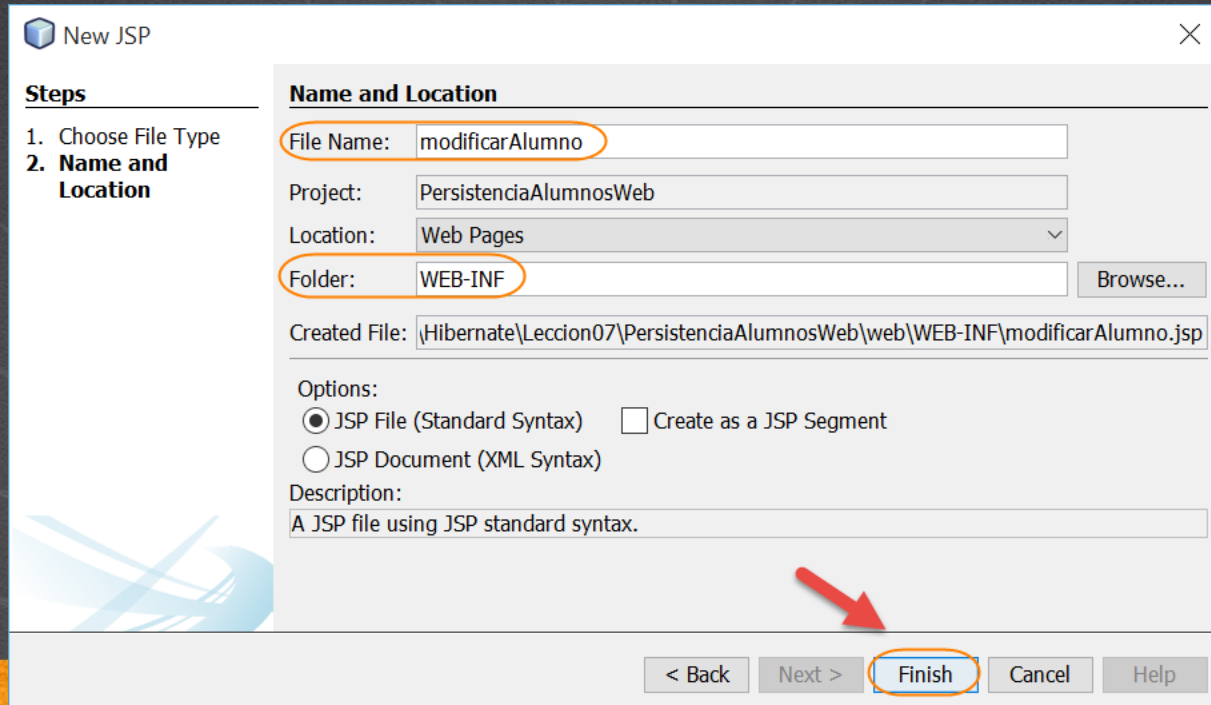
```
<br/>
Datos Domicilio:
<br/>
Calle: <input type="text" name="calle" value="\${alumno.domicilio.calle}">
<br/>
<br/>
Datos Contacto:
<br/>
email: <input type="text" name="email" value="\${alumno.contacto.email1}">
<br/>
telefono: <input type="text" name="telefono" value="\${alumno.contacto.telefono}">
<br/>

<input type="submit" name="modificar" value="modificar">
<input type="submit" name="eliminar" value="eliminar">

<!-- No necesitamos manejar valores ocultos
ya que nos apoyaremos del objeto en sesion para
la edicion -->
</form>
</body>
</html>
```

PASO 27. CREAR UN ARCHIVO LOG4J2

Creamos el archivo log4j2.xml



New JSP

Steps

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

Name and Location

File Name:

Project:

Location:

Folder:

Created File:

Options:

☒ JSP File (Standard Syntax) ☐ Create as a JSP Segment

☐ JSP Document (XML Syntax)

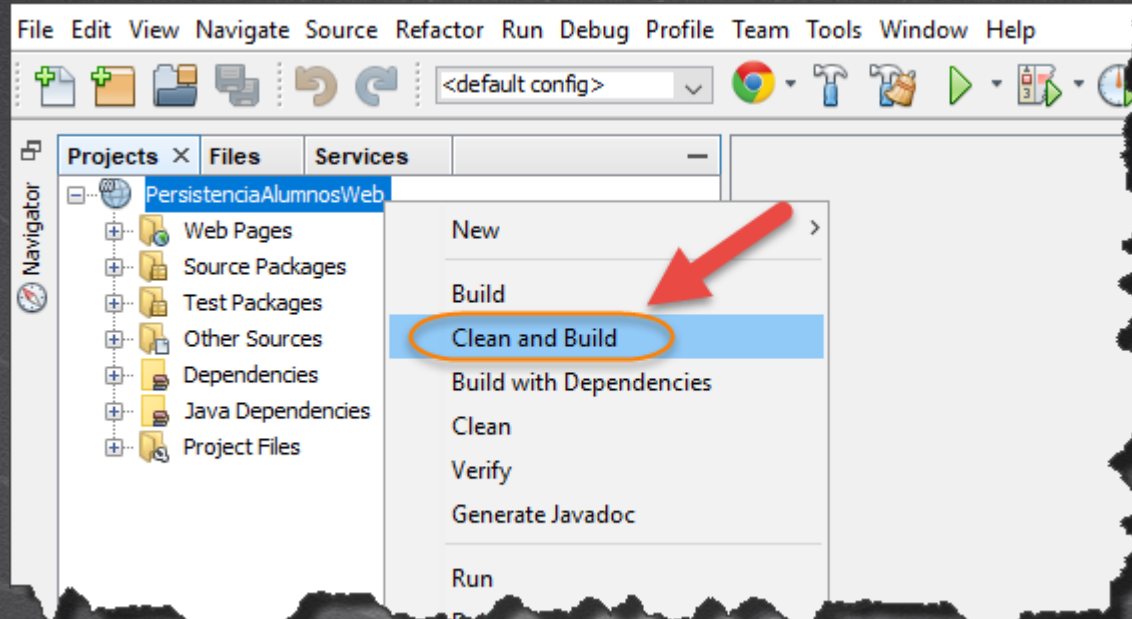
Description:

CURSO HIBERNATE Y JPA

www.globalmentoring.com.mx

PASO 28. HACEMOS CLEAN & BUILD

Hacemos Clean & Build al proyecto:

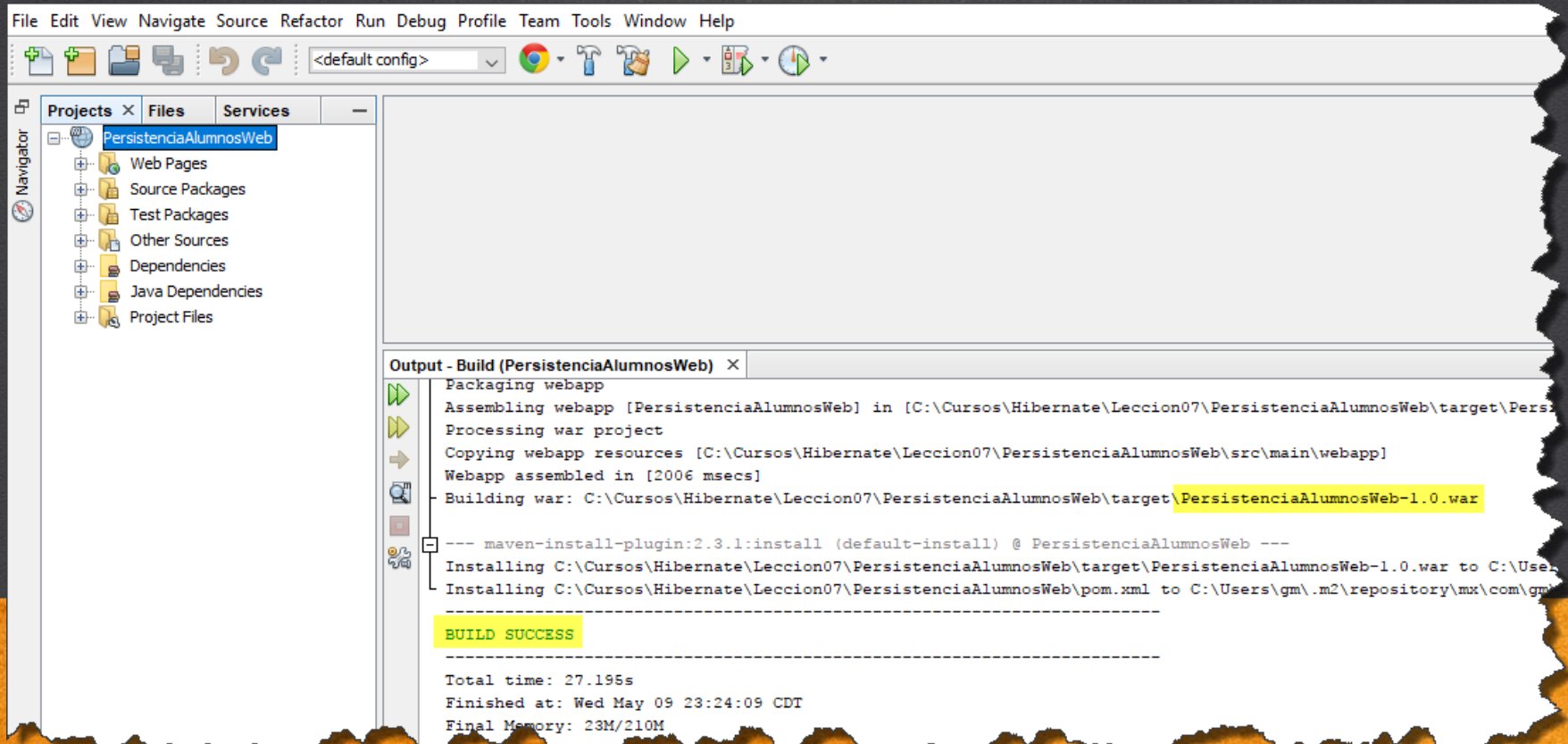


CURSO HIBERNATE Y JPA

www.globalmentoring.com.mx

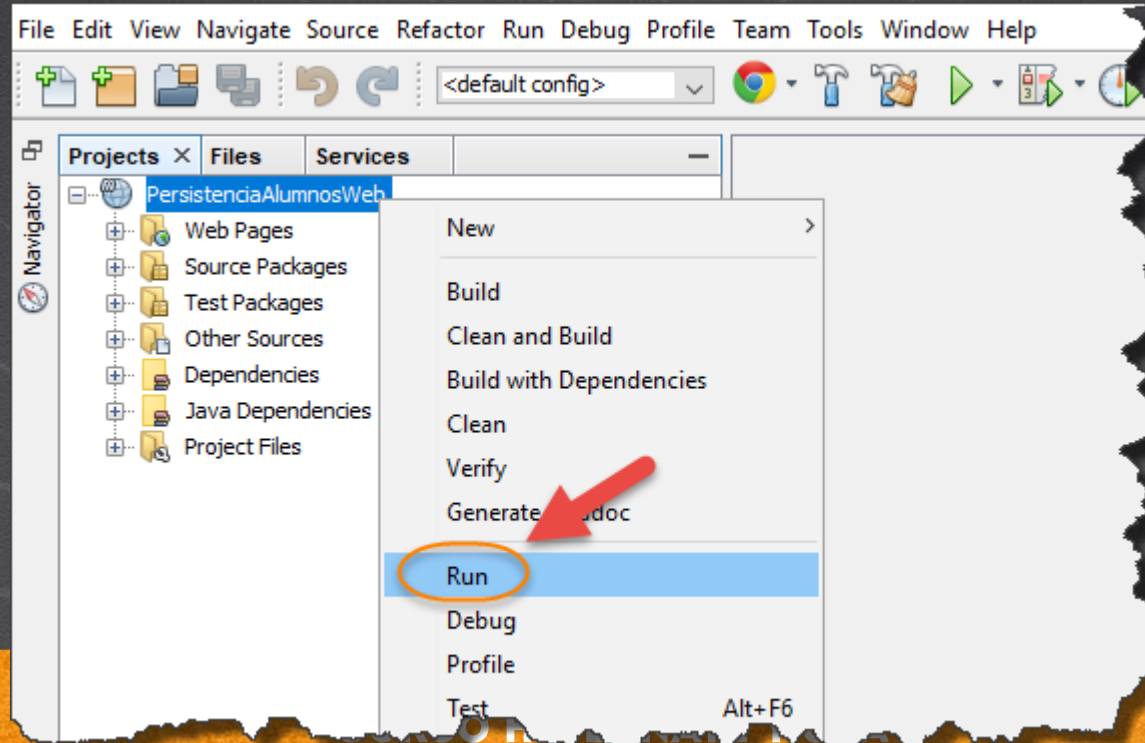
PASO 28. HACEMOS CLEAN & BUILD

Hacemos Clean & Build al proyecto:



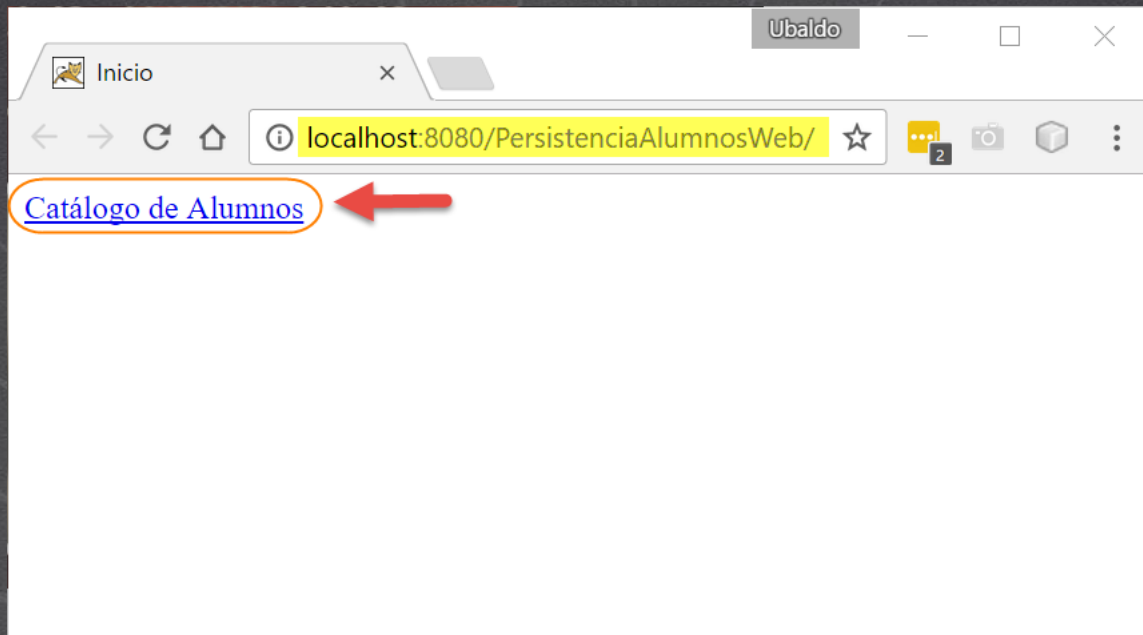
PASO 29. EJECUTAMOS EL PROYECTO

Ejecutamos el proyecto:



PASO 29. EJECUTAMOS EL PROYECTO (CONT)

Entramos al listado de alumnos:

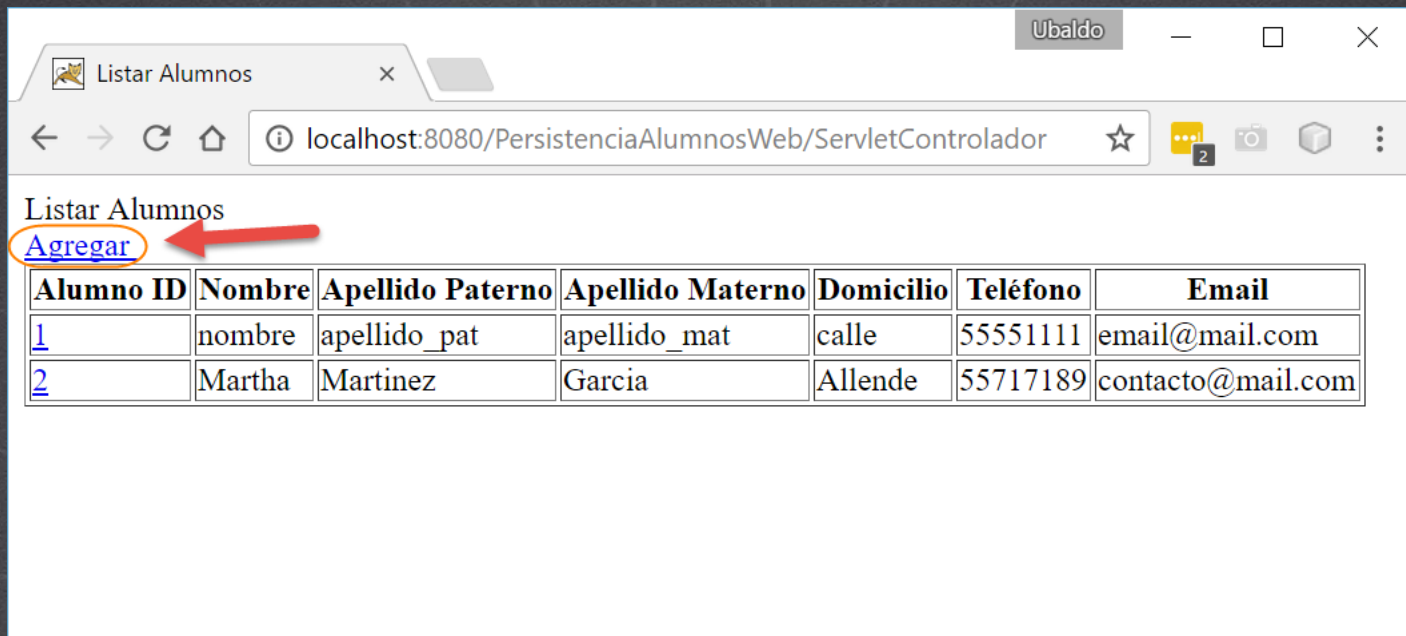


CURSO HIBERNATE Y JPA

www.globalmentoring.com.mx

PASO 29. EJECUTAMOS EL PROYECTO (CONT)

Agregamos un alumno:



The screenshot shows a web browser window with the title 'Ubaldo'. The address bar displays 'localhost:8080/PersistenciaAlumnosWeb/ServletControlador'. The page content includes a link 'Listar Alumnos' and a button 'Agregar' which is circled in blue and pointed to by a red arrow. Below the button is a table with student information.

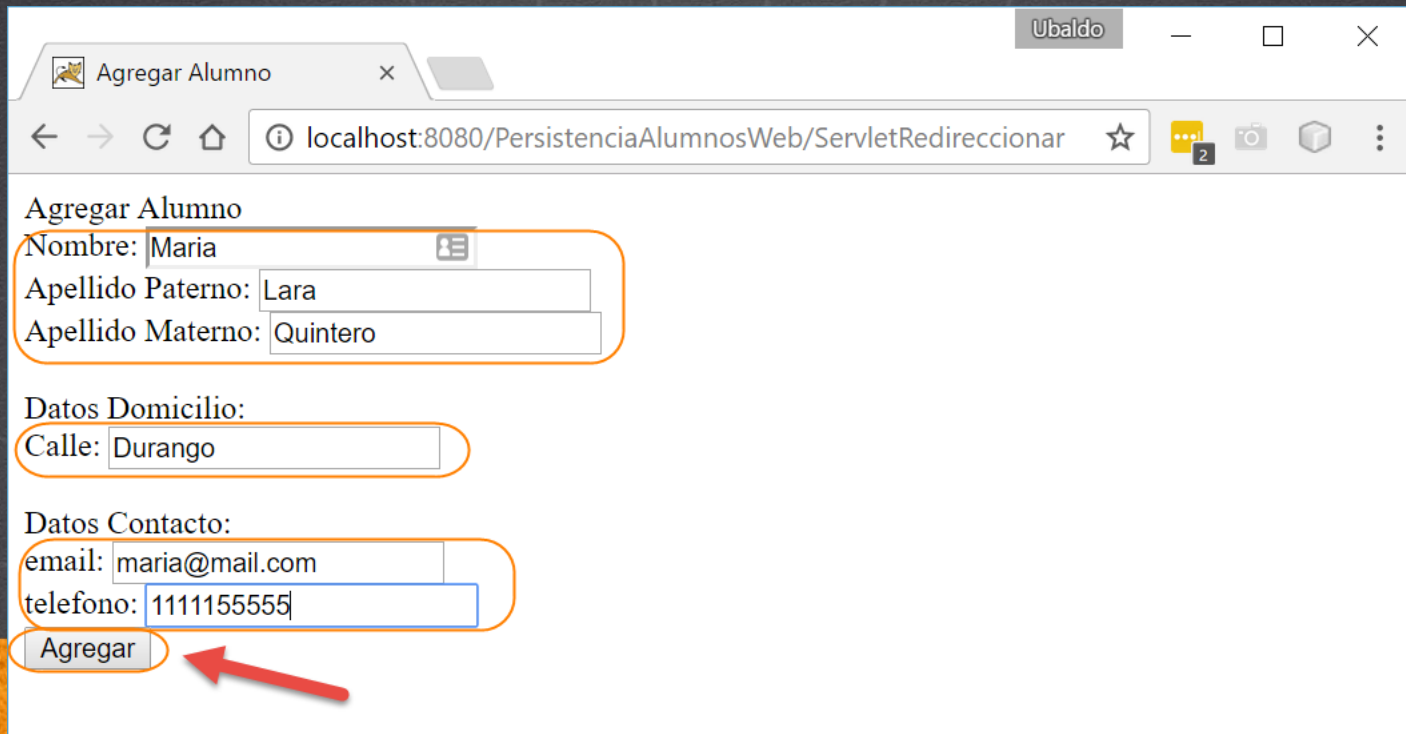
Alumno ID	Nombre	Apellido Paterno	Apellido Materno	Domicilio	Teléfono	Email
1	nombre	apellido_pat	apellido_mat	calle	55551111	email@mail.com
2	Martha	Martinez	Garcia	Allende	55717189	contacto@mail.com

CURSO HIBERNATE Y JPA

www.globalmentoring.com.mx

PASO 29. EJECUTAMOS EL PROYECTO (CONT)

Agregamos un Alumno:



Ubaldo

Agregar Alumno

Nombre:

Apellido Paterno:

Apellido Materno:

Datos Domicilio:

Calle:

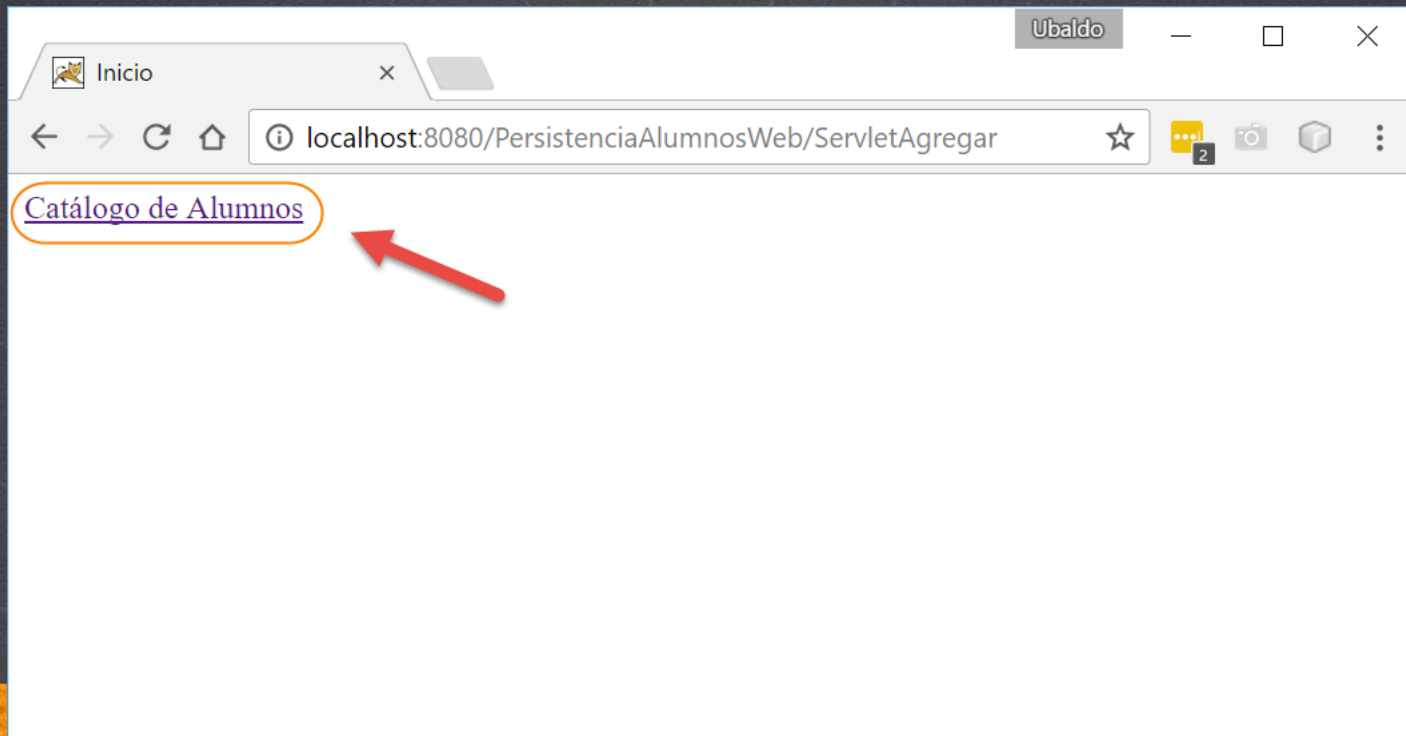
Datos Contacto:

email:

telefono:

PASO 29. EJECUTAMOS EL PROYECTO (CONT)

Agregamos un Alumno:

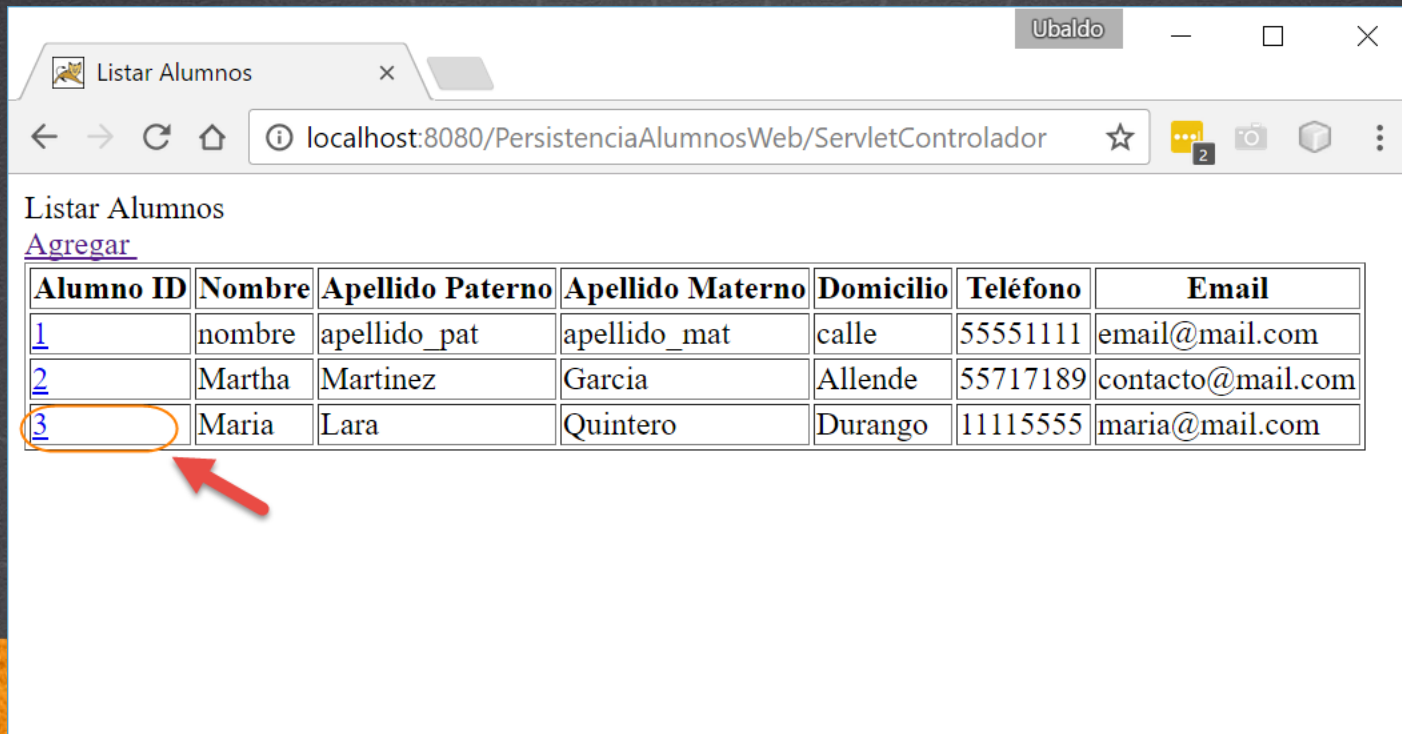


CURSO HIBERNATE Y JPA

www.globalmentoring.com.mx

PASO 29. EJECUTAMOS EL PROYECTO (CONT)

Revisamos el alumno agregado, y lo editamos:



Ubaldo

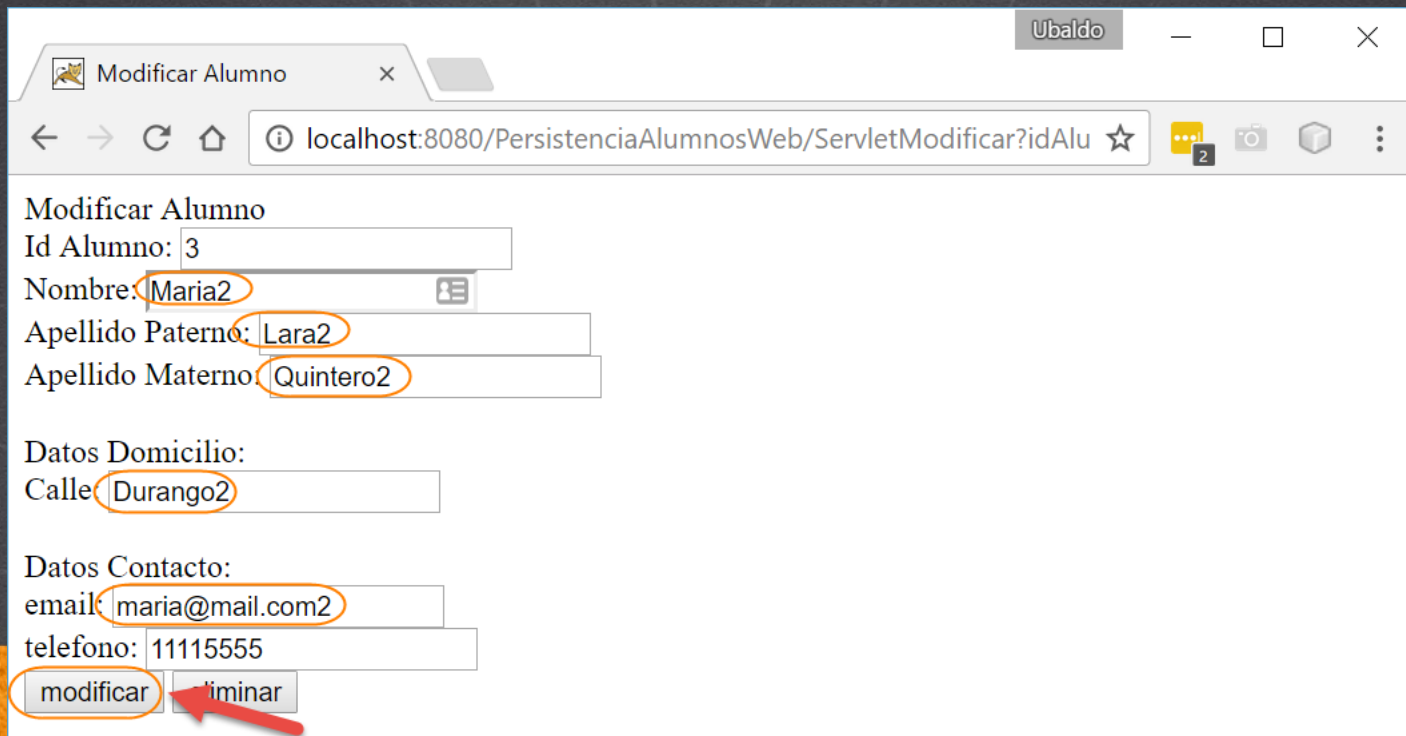
Listar Alumnos

[Agregar](#)

Alumno ID	Nombre	Apellido Paterno	Apellido Materno	Domicilio	Teléfono	Email
1	nombre	apellido_pat	apellido_mat	calle	55551111	email@mail.com
2	Martha	Martinez	Garcia	Allende	55717189	contacto@mail.com
3	Maria	Lara	Quintero	Durango	11115555	maria@mail.com

PASO 29. EJECUTAMOS EL PROYECTO (CONT)

Modificamos el alumno seleccionado:



Ubaldo

Modificar Alumno

localhost:8080/PersistenciaAlumnosWeb/ServletModificar?idAlu

Modificar Alumno

Id Alumno: 3

Nombre: Maria2

Apellido Paterno: Lara2

Apellido Materno: Quintero2

Datos Domicilio:

Calle: Durango2

Datos Contacto:

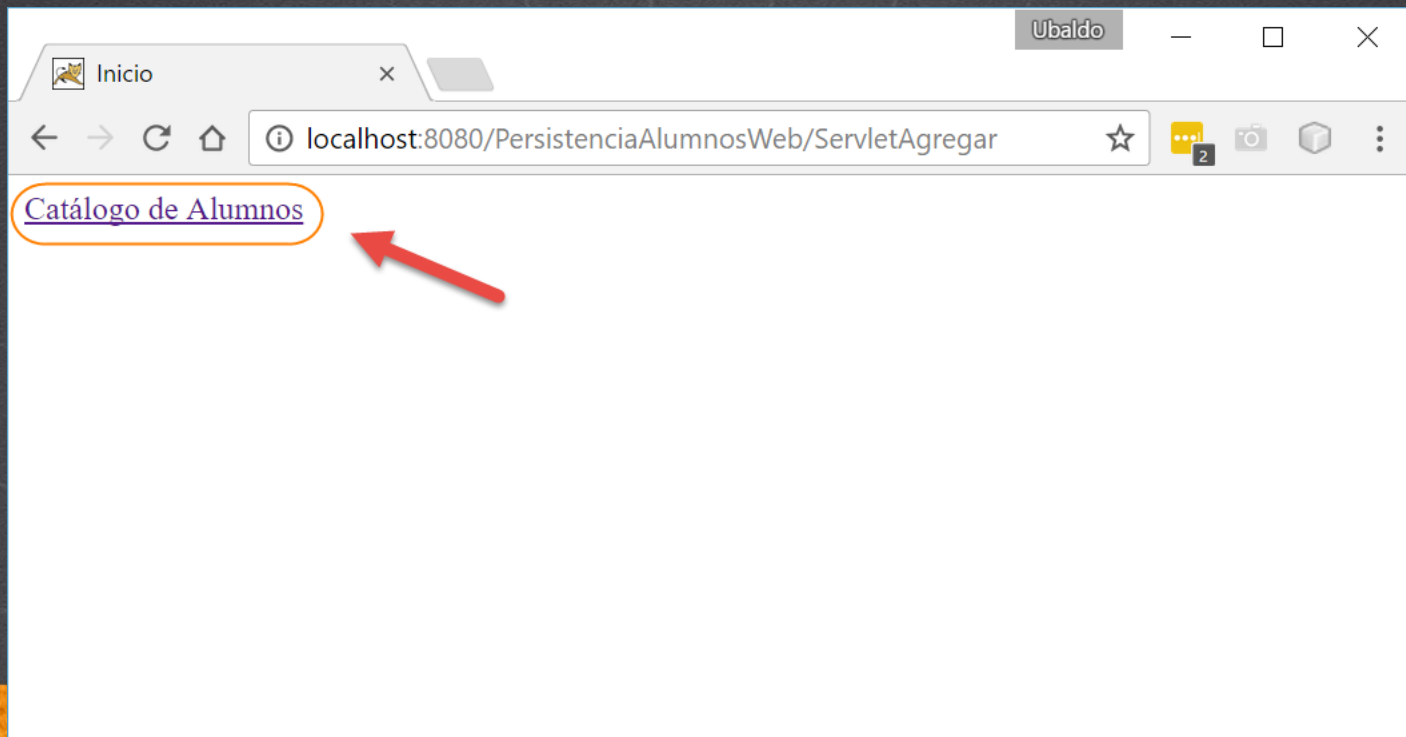
email: maria@mail.com2

telefono: 11115555

modificar eliminar

PASO 29. EJECUTAMOS EL PROYECTO (CONT)

Eliminamos un Alumno:

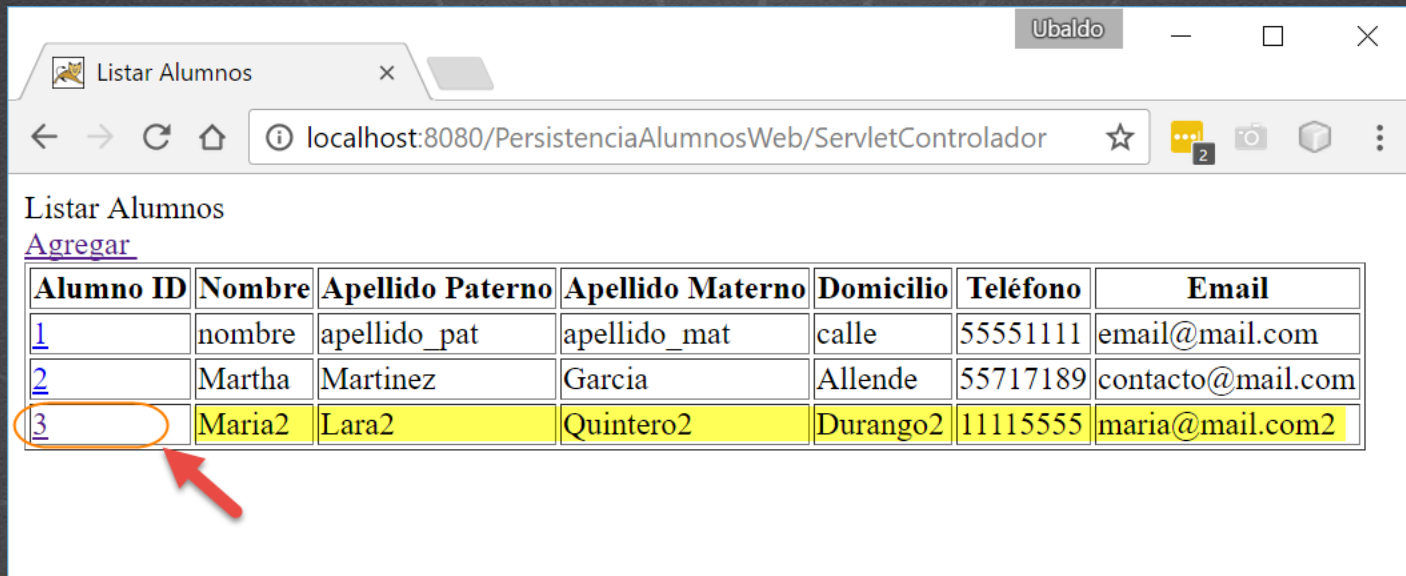


CURSO HIBERNATE Y JPA

www.globalmentoring.com.mx

PASO 29. EJECUTAMOS EL PROYECTO (CONT)

Vemos el alumnos modificado y lo seleccionamos para eliminarlo:

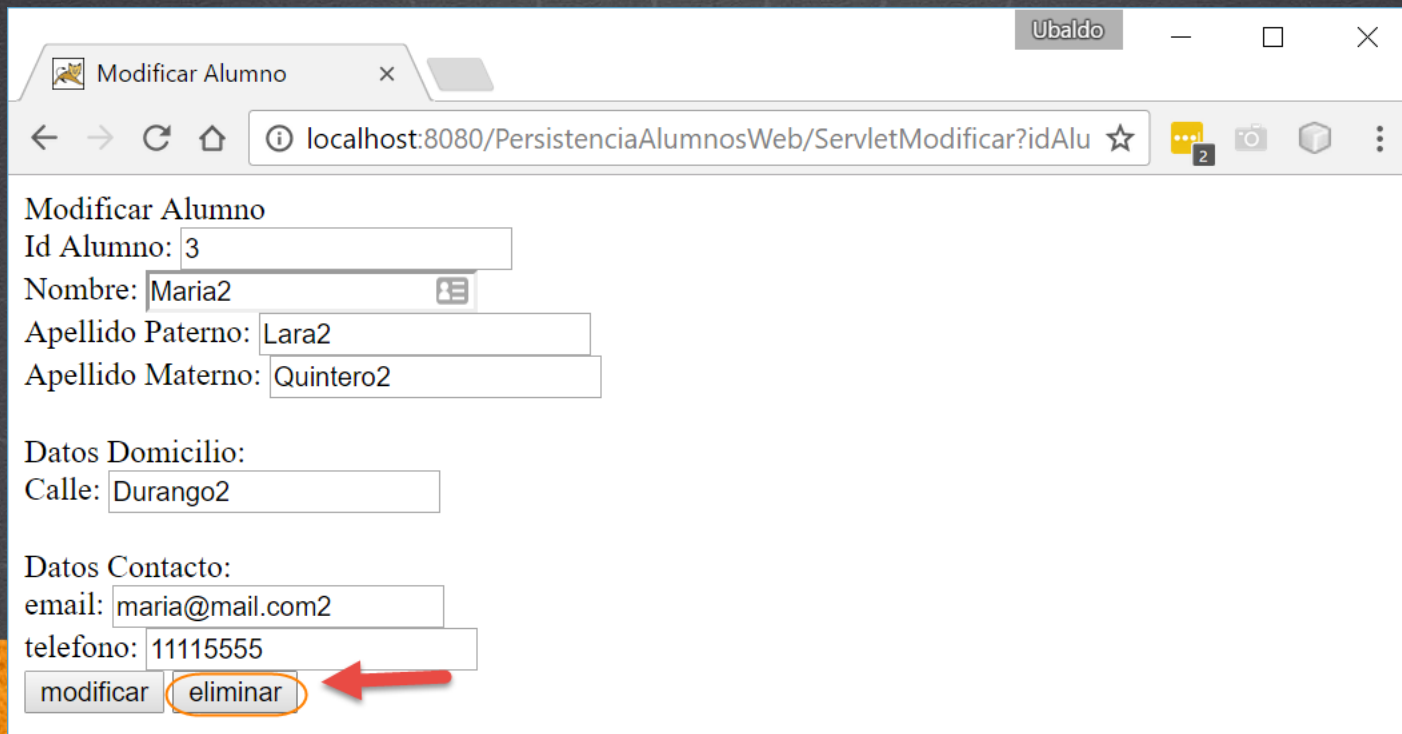


The screenshot shows a web browser window with the title 'Ubaldo'. The address bar displays 'localhost:8080/PersistenciaAlumnosWeb/ServletControlador'. The page content includes a link 'Agregar' and a table with student information. The third row of the table is highlighted in yellow, and the ID '3' is circled in orange with a red arrow pointing to it.

Alumno ID	Nombre	Apellido Paterno	Apellido Materno	Domicilio	Teléfono	Email
1	nombre	apellido_pat	apellido_mat	calle	55551111	email@mail.com
2	Martha	Martinez	Garcia	Allende	55717189	contacto@mail.com
3	Maria2	Lara2	Quintero2	Durango2	11115555	maria@mail.com2

PASO 29. EJECUTAMOS EL PROYECTO (CONT)

Eliminamos el alumno seleccionado:



Ubaldo

Modificar Alumno

localhost:8080/PersistenciaAlumnosWeb/ServletModificar?idAlu

Modificar Alumno

Id Alumno: 3

Nombre: Maria2

Apellido Paterno: Lara2

Apellido Materno: Quintero2

Datos Domicilio:

Calle: Durango2

Datos Contacto:

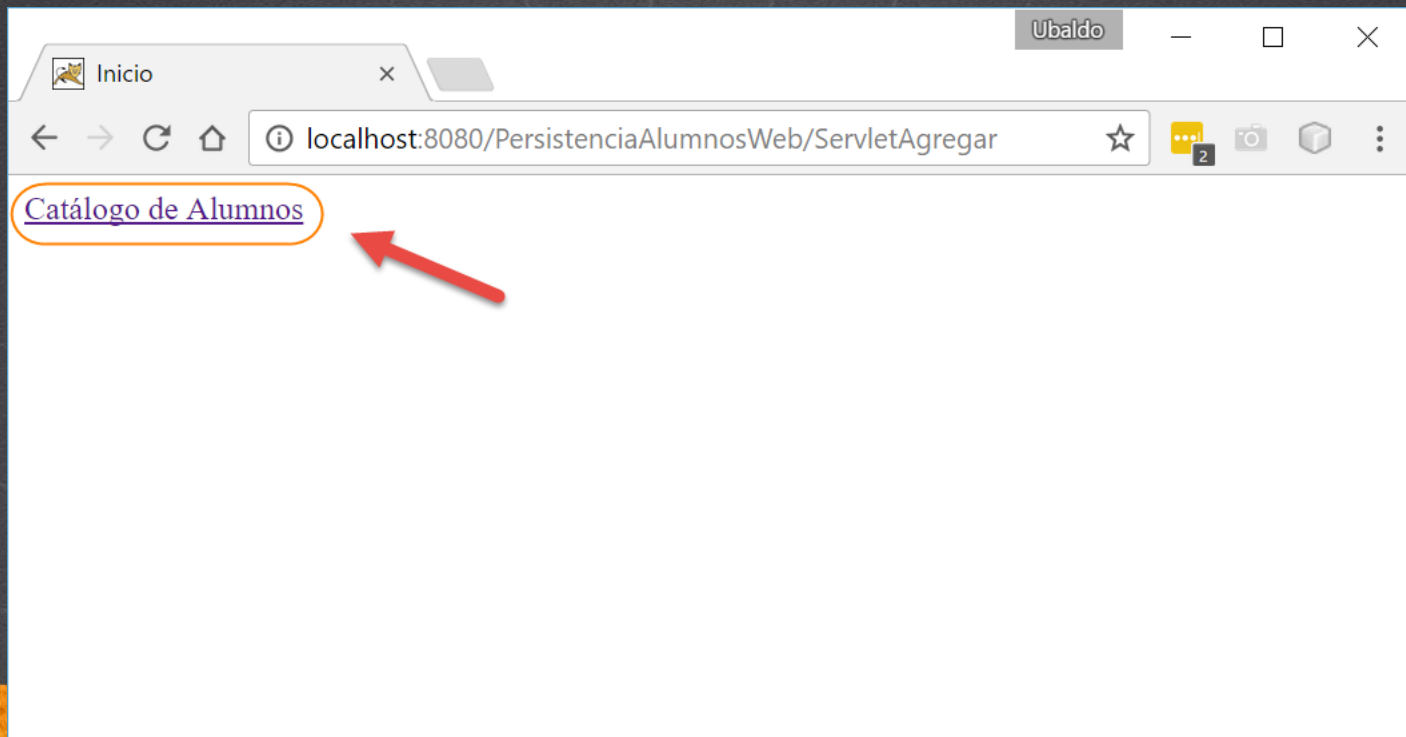
email: maria@mail.com2

telefono: 11115555

modificar eliminar

PASO 29. EJECUTAMOS EL PROYECTO (CONT)

Eliminamos un Alumno:

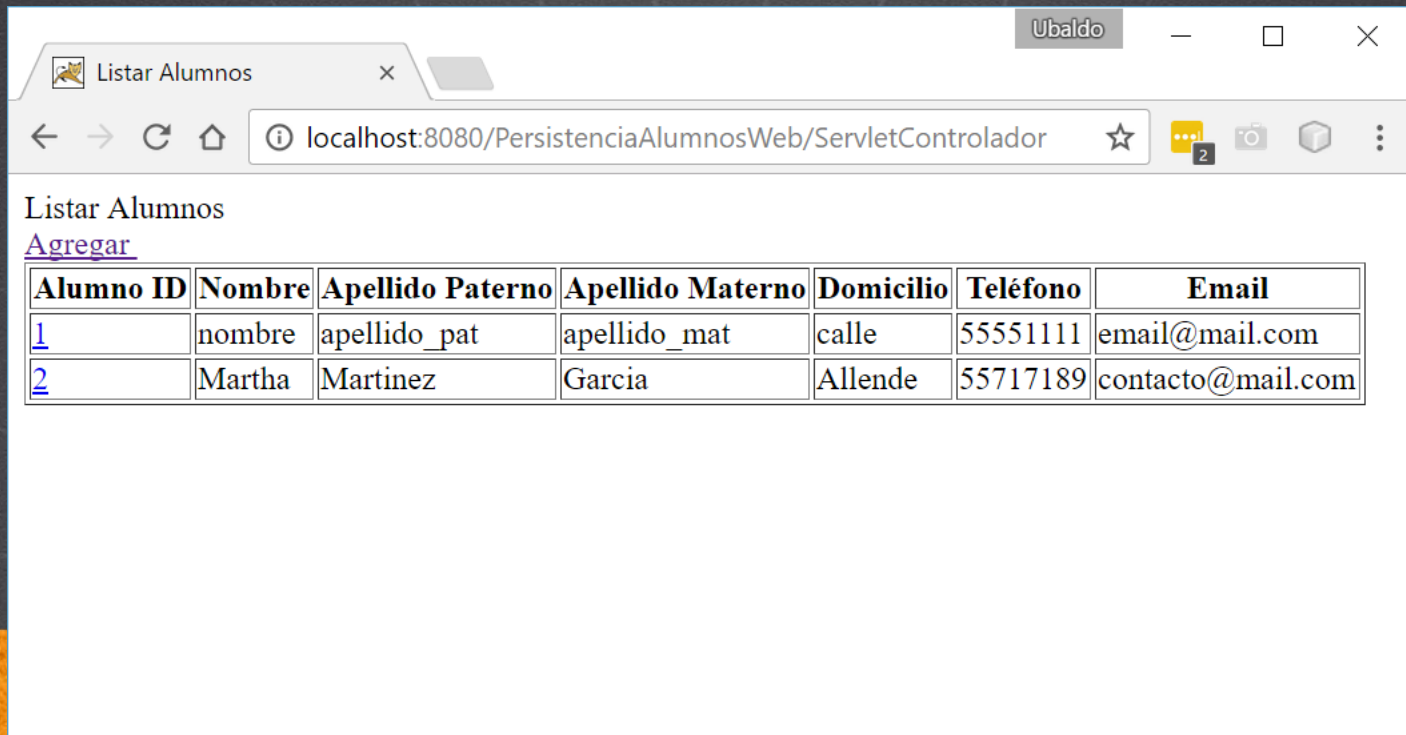


CURSO HIBERNATE Y JPA

www.globalmentoring.com.mx

PASO 29. EJECUTAMOS EL PROYECTO (CONT)

Vemos que ya se haya eliminado el Alumno:



The screenshot shows a web browser window with the title 'Ubaldo'. The address bar displays 'localhost:8080/PersistenciaAlumnosWeb/ServletControlador'. The page content includes the text 'Listar Alumnos' and a link 'Agregar'. Below this is a table with 7 columns: 'Alumno ID', 'Nombre', 'Apellido Paterno', 'Apellido Materno', 'Domicilio', 'Teléfono', and 'Email'. The table contains two rows of data. The first row has values: '1', 'nombre', 'apellido_pat', 'apellido_mat', 'calle', '55551111', and 'email@mail.com'. The second row has values: '2', 'Martha', 'Martinez', 'Garcia', 'Allende', '55717189', and 'contacto@mail.com'.

Alumno ID	Nombre	Apellido Paterno	Apellido Materno	Domicilio	Teléfono	Email
1	nombre	apellido_pat	apellido_mat	calle	55551111	email@mail.com
2	Martha	Martinez	Garcia	Allende	55717189	contacto@mail.com

CONCLUSIÓN DEL EJERCICIO

- Con este ejercicio hemos integrado una aplicación Web con Hibernate y JPA utilizando Glassfish como servidor de aplicaciones.
- Aplicamos varios patrones de diseño como MVC, DAO y DTO para en su conjunto poder desplegar realizar las acciones respectivas, todo esto con ayuda de la capa de persistencia apoyados de Hibernate y JPA.
- Agregamos las acciones de listar, agregar, modificar y eliminar el listado de Alumnos.
- También aprovechamos para aplicar el concepto de persistencia en cascada, tanto al agregar, modificar o eliminar un objeto de tipo Alumno junto con las relaciones marcadas como Cascade, que son los objetos de Domicilio y Contacto.

CURSO HIBERNATE Y JPA

www.globalmentoring.com.mx

CURSO ONLINE

HIBERNATE & JPA

Por: Ing. Ubaldo Acosta



Experiencia y Conocimiento para tu vida

CURSO DE HIBERNATE Y JPA

www.globalmentoring.com.mx