CURSO HIBERNATE Y JPA

LABORATORIO FINAL SGA CON HIBERNATE Y JPA

Por el experto: Ing. Ubaldo Acosta

Global Mentoring

Experiencia y Conocimiento para tu vida

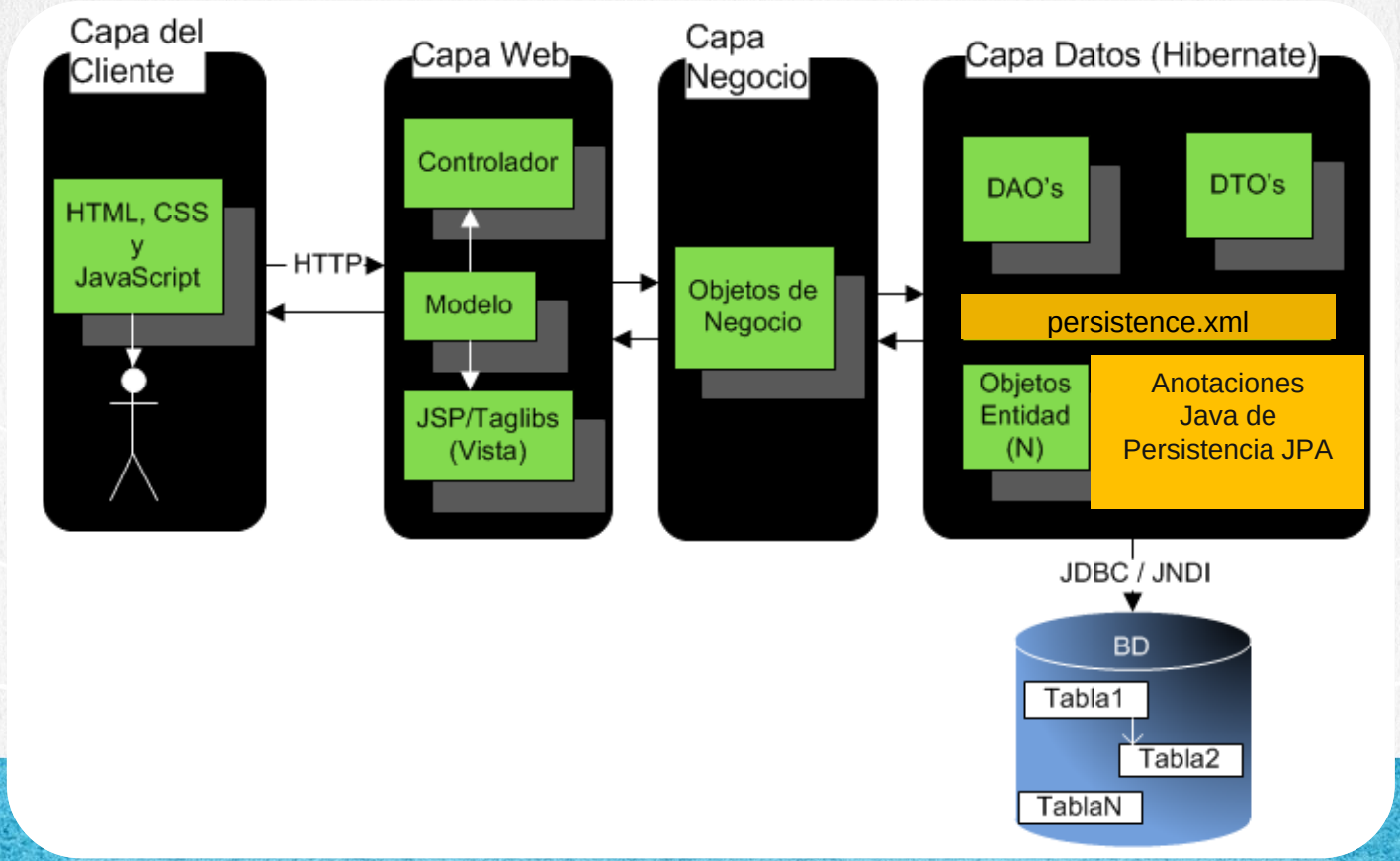
UNIVERSIDAD JAVA CURSO ONLINE

LABORATORIO FINAL SGA

Laboratorio Final SGA (Sistema de Gestión de Alumnos). Algunas de las características que tendrá el sistema son:

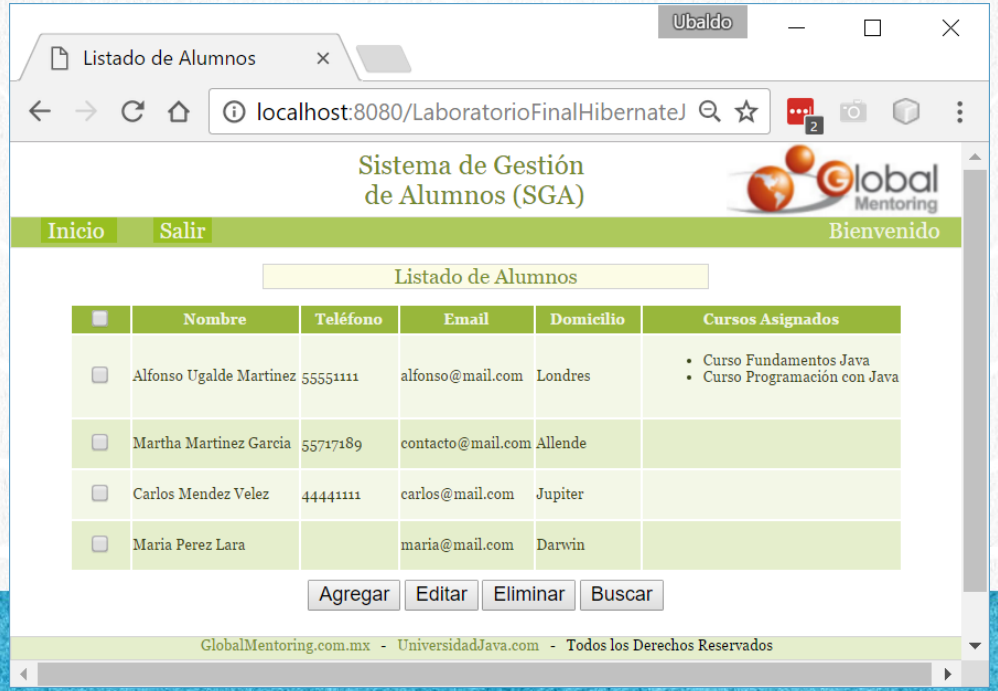
- ✓ Login al sistema.
- ✓ Consulta de Alumnos
- ✓ Modificación de Alumnos
- ✓ Eliminación de Alumnos
- ✓ Búsqueda Avanzada de Alumnos

ARQUITECTURA DEL PROYECTO WEB



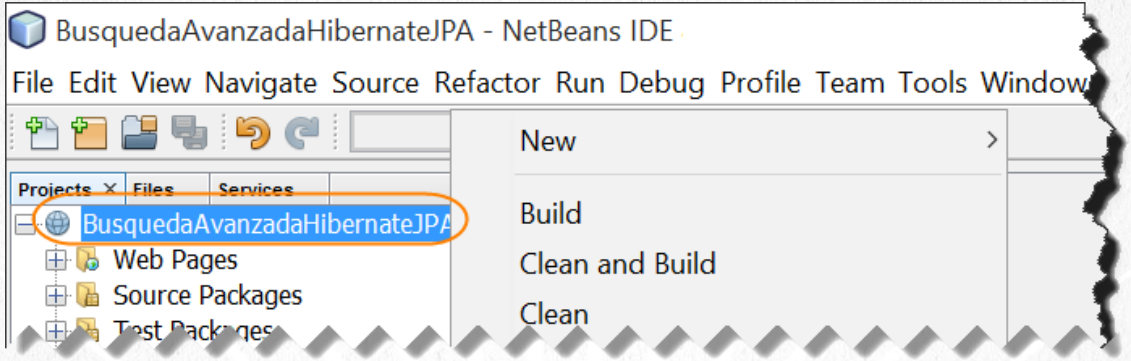
OBJETIVO DEL EJERCICIO

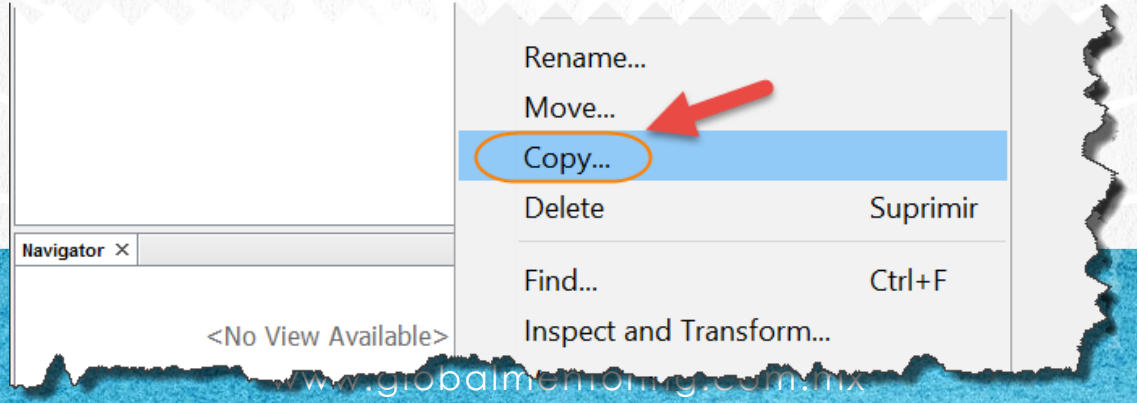
Crear el proyecto final del curso de Hibernate/JPA. Al finalizar deberemos ver:



PASO 1. COPIAMOS EL PROYECTO

Copiamos el proyecto BusquedaAvanzadaHibernateJPA:

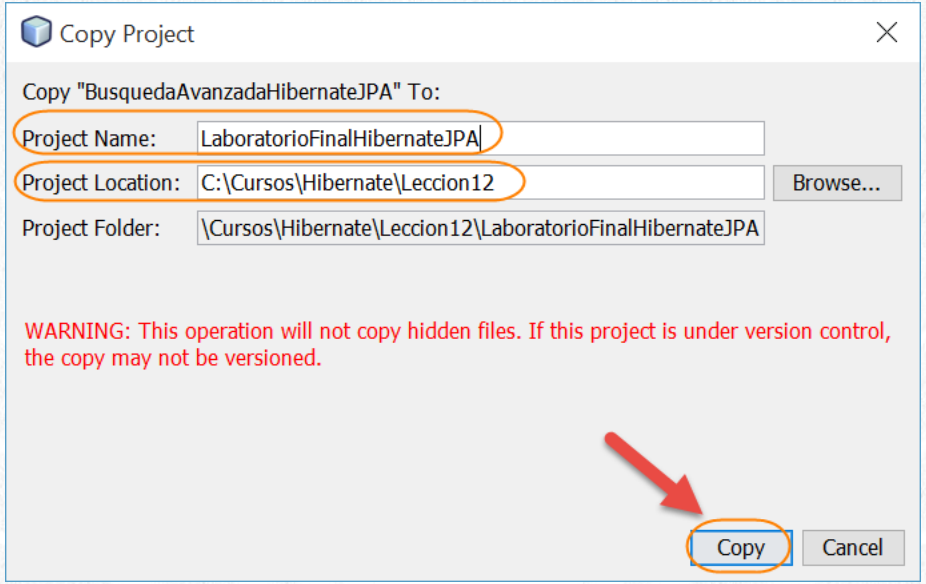




www.globalmentoring.com.mx

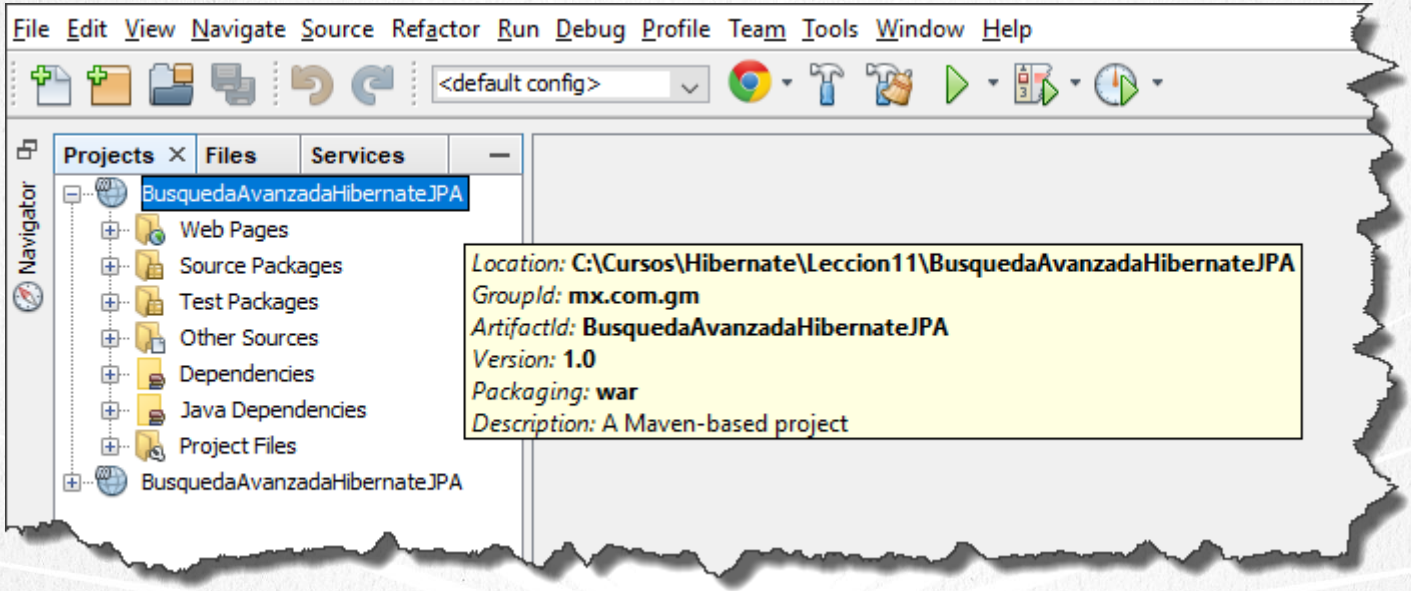
PASO 1. COPIAMOS EL PROYECTO

Creamos el nuevo proyecto LaboratorioFinalHibernateJPA:



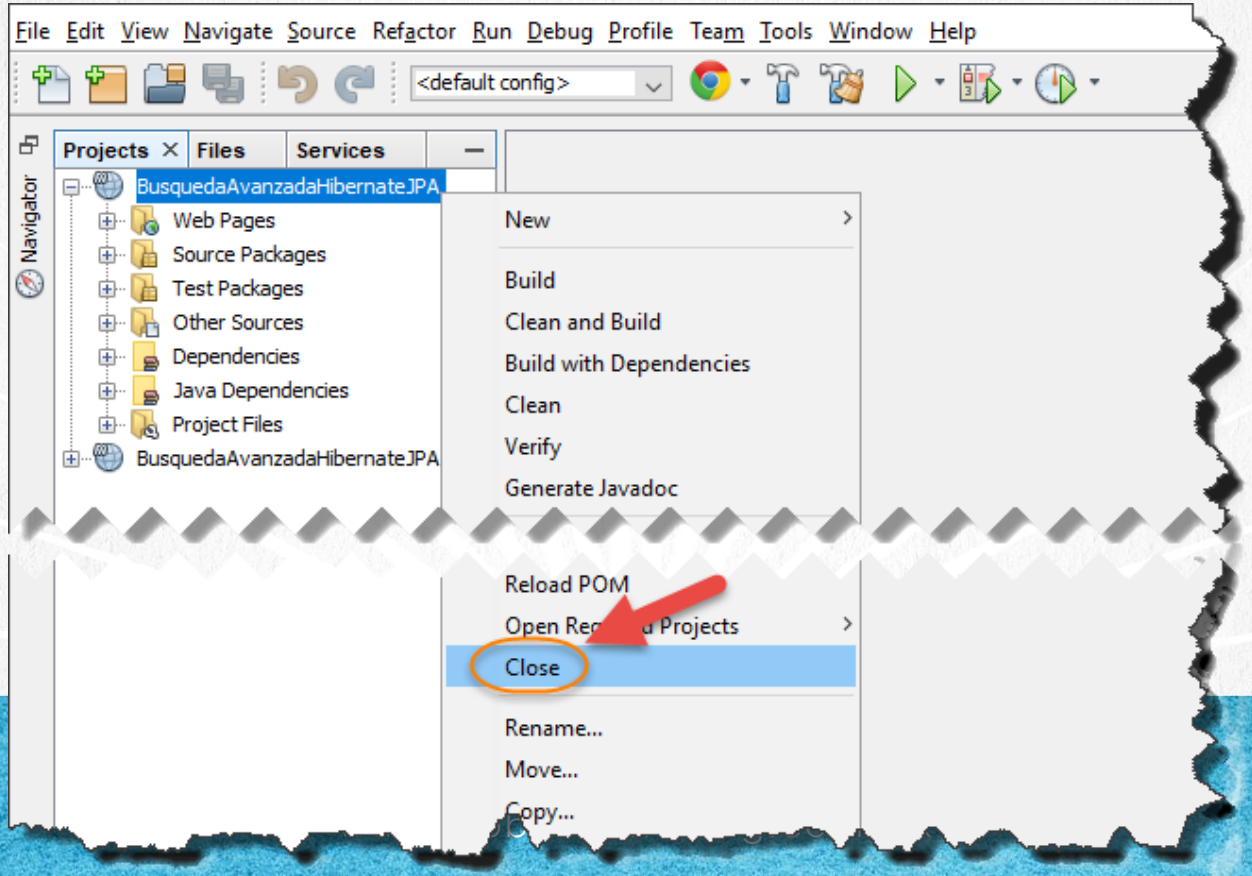
PASO 2. CERRAMOS EL PROYECTO NO USADO

Cerramos el proyecto que ya no utilizaremos:



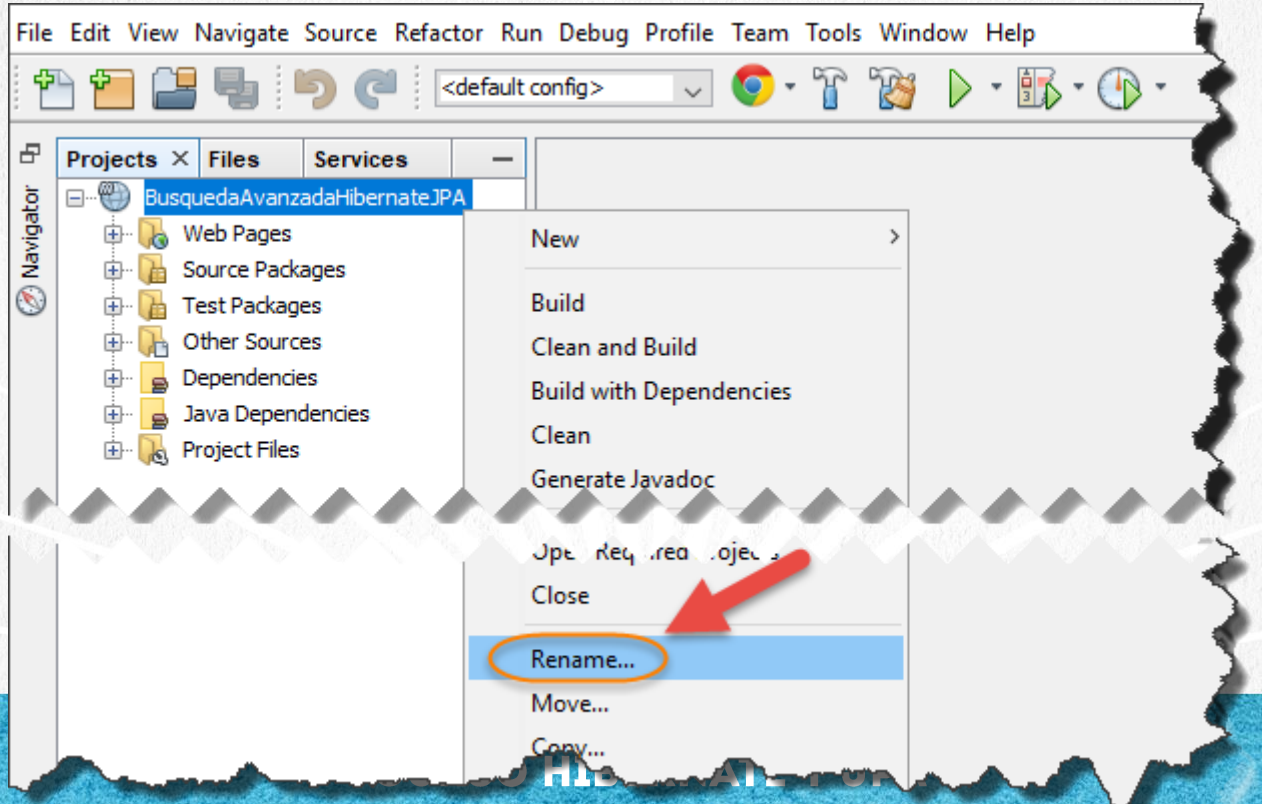
PASO 2. CERRAMOS EL PROYECTO NO USADO

Cerramos el proyecto que ya no utilizaremos:



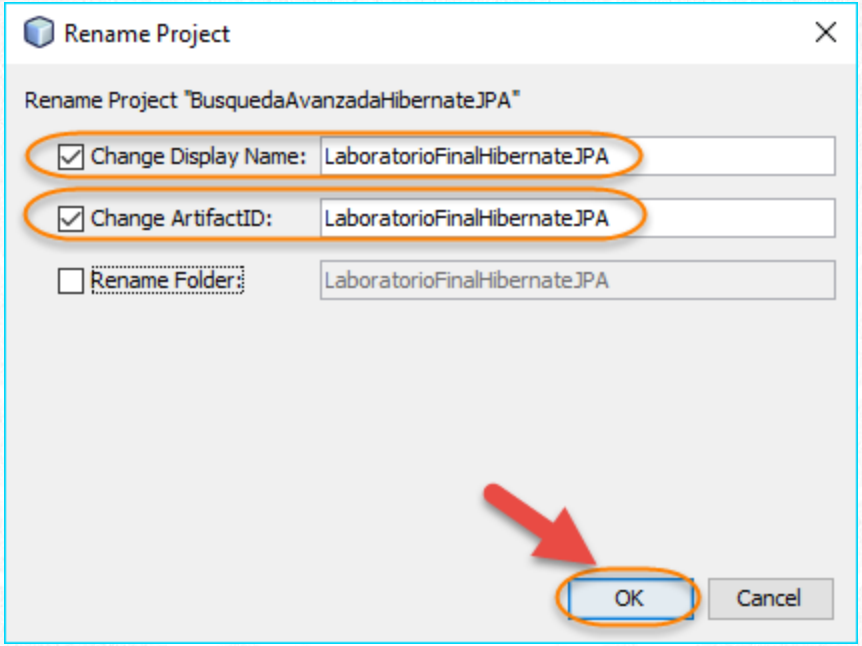
PASO 3. RENOMBRAMOS EL PROYECTO

Renombramos el proyecto:



PASO 3. RENOMBRAMOS EL PROYECTO

Renombramos el proyecto:



SIGUIENTES PASOS

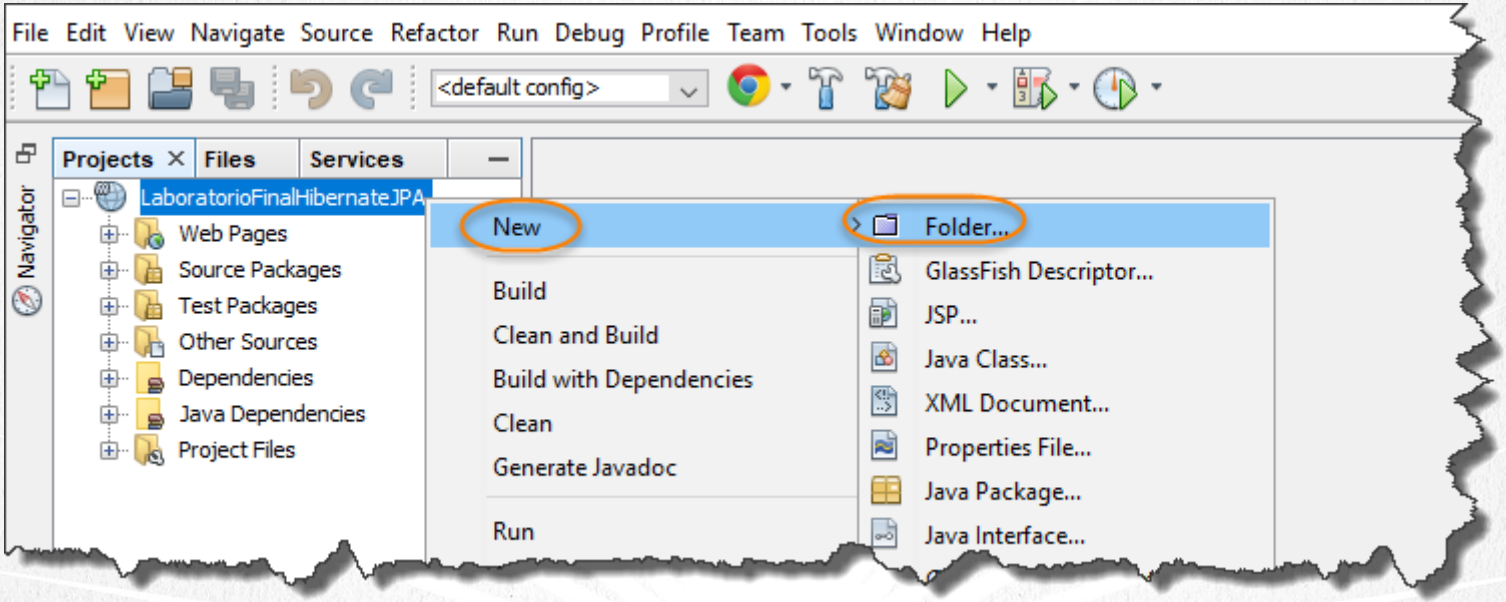
En los siguientes pasos primero vamos a crear toda la estructura del proyecto, sin modificar ni agregar contenido a cada archivo, y posteriormente ya que hayamos terminado de crear toda la estructura del proyecto, vamos a ir agregando, modificando o eliminando el contenido del proyecto.

Veamos cómo queda la estructura de nuestro proyecto.

Experiencia y Conocimiento para tu vida

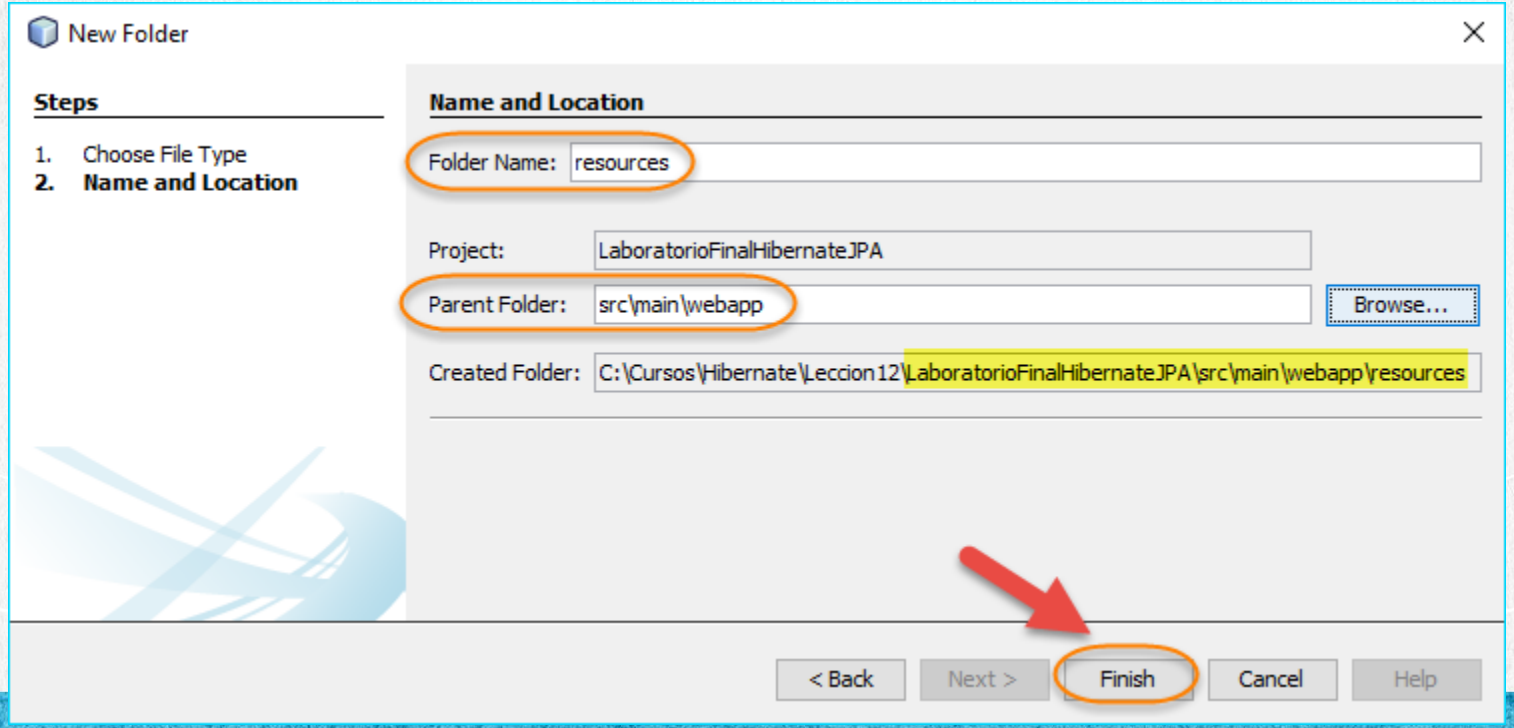
PASO 4. CREAMOS UNA CARPETA

Creamos la carpeta de resources:



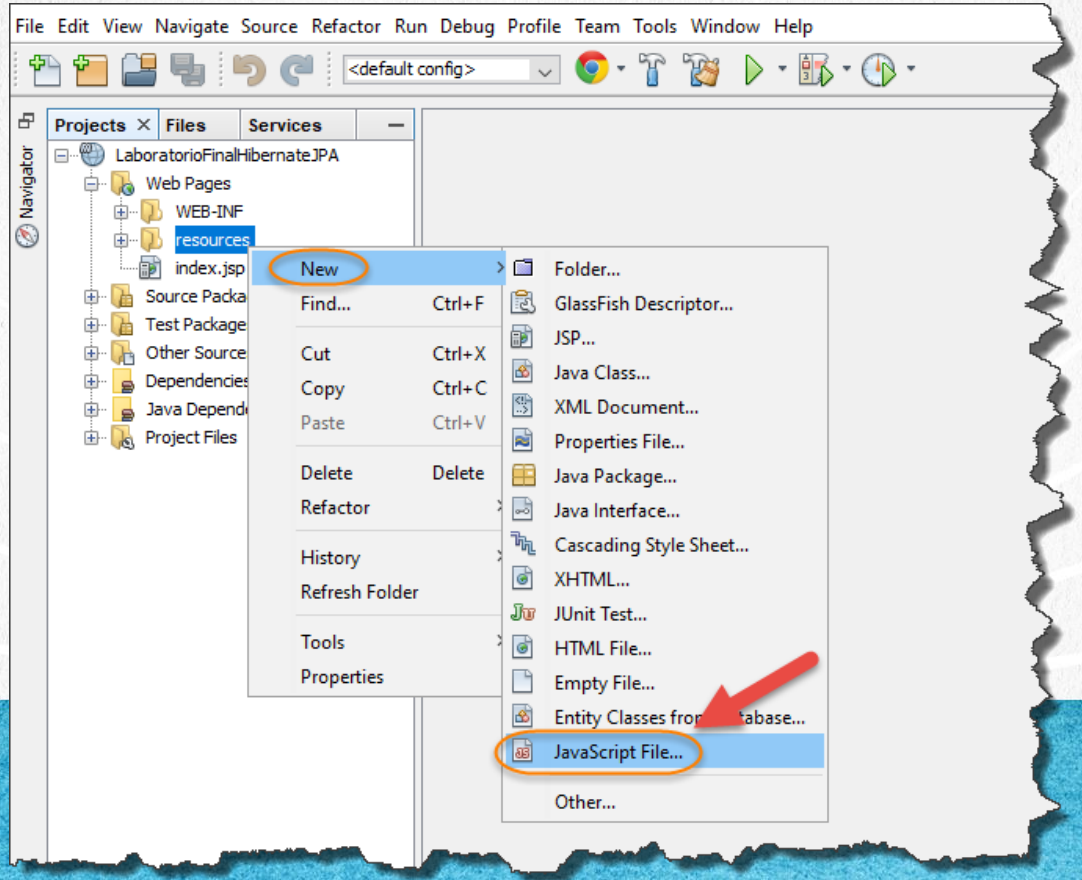
PASO 4. CREAMOS UNA CARPETA

Creamos la carpeta de resources:



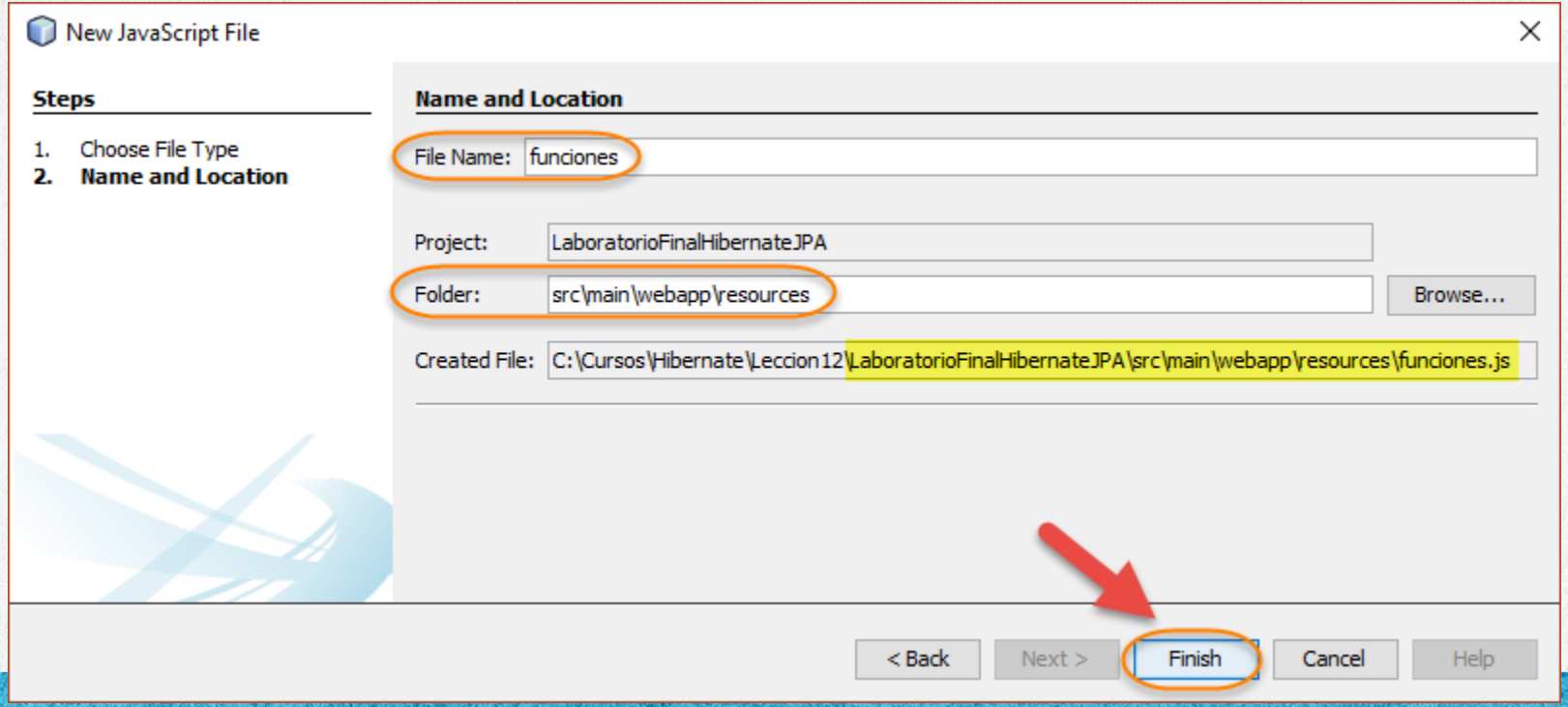
PASO 5. CREAMOS UN ARCHIVO JS

Creamos el archivo funciones.js:



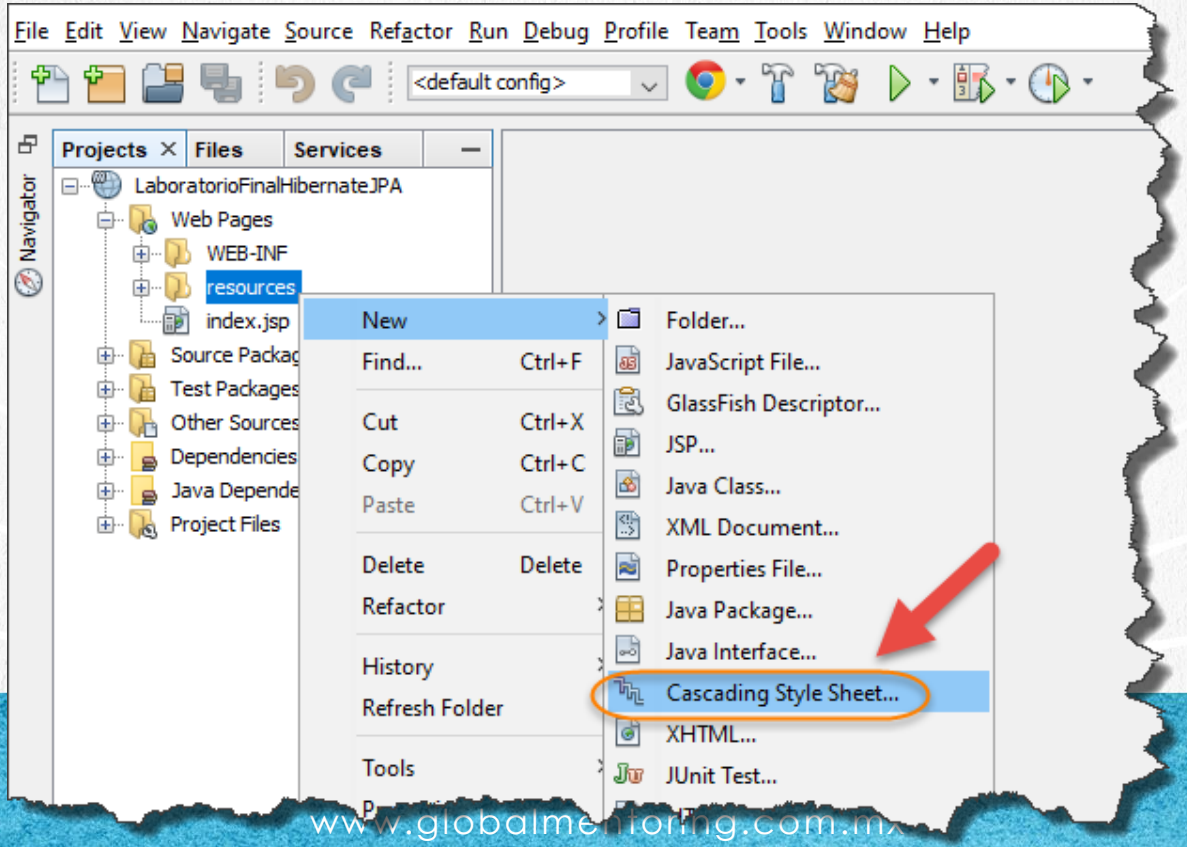
PASO 5. CREAMOS UN ARCHIVO JS

Creamos el archivo funciones.js:



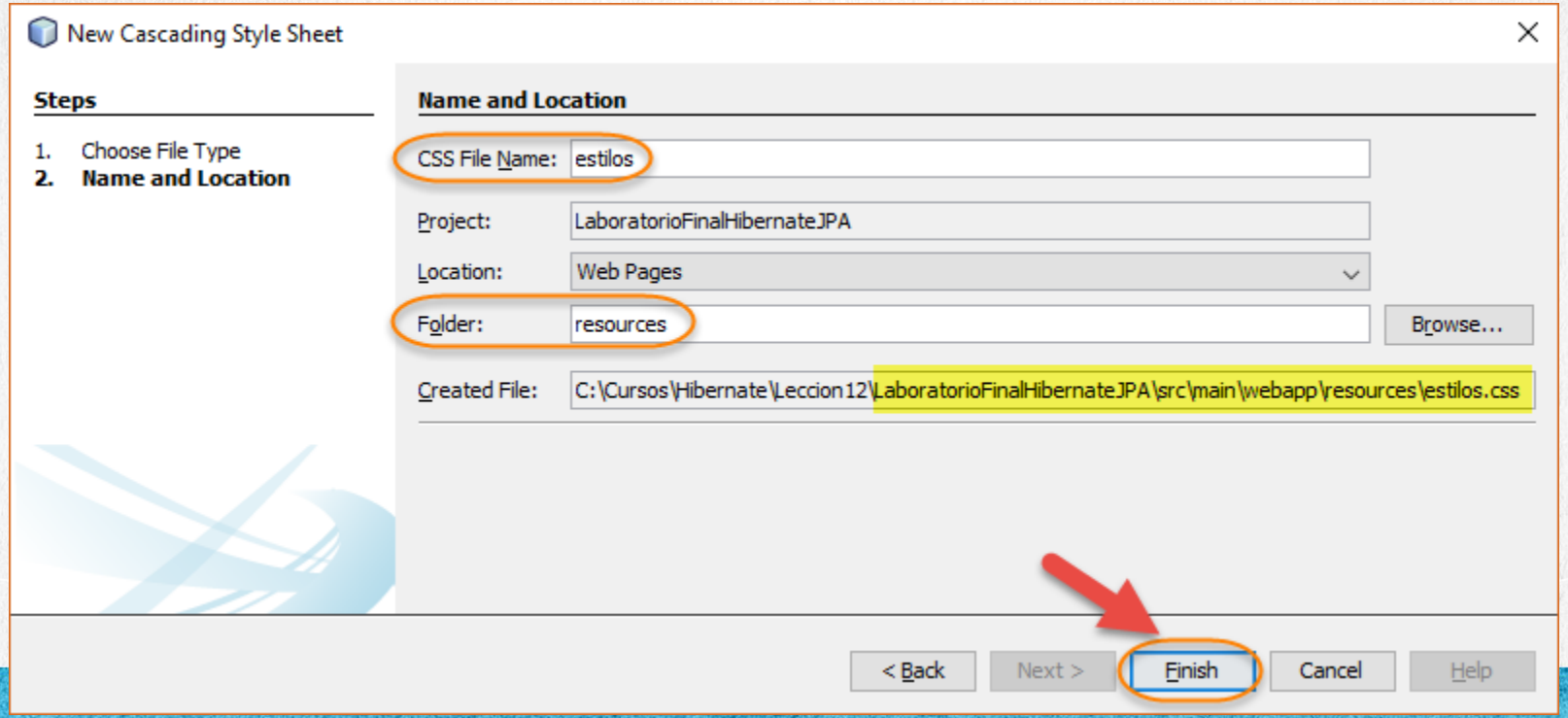
PASO 6. CREAMOS UN ARCHIVO JS

Creamos el archivo estilos.css:



PASO 6. CREAMOS UN ARCHIVO JS

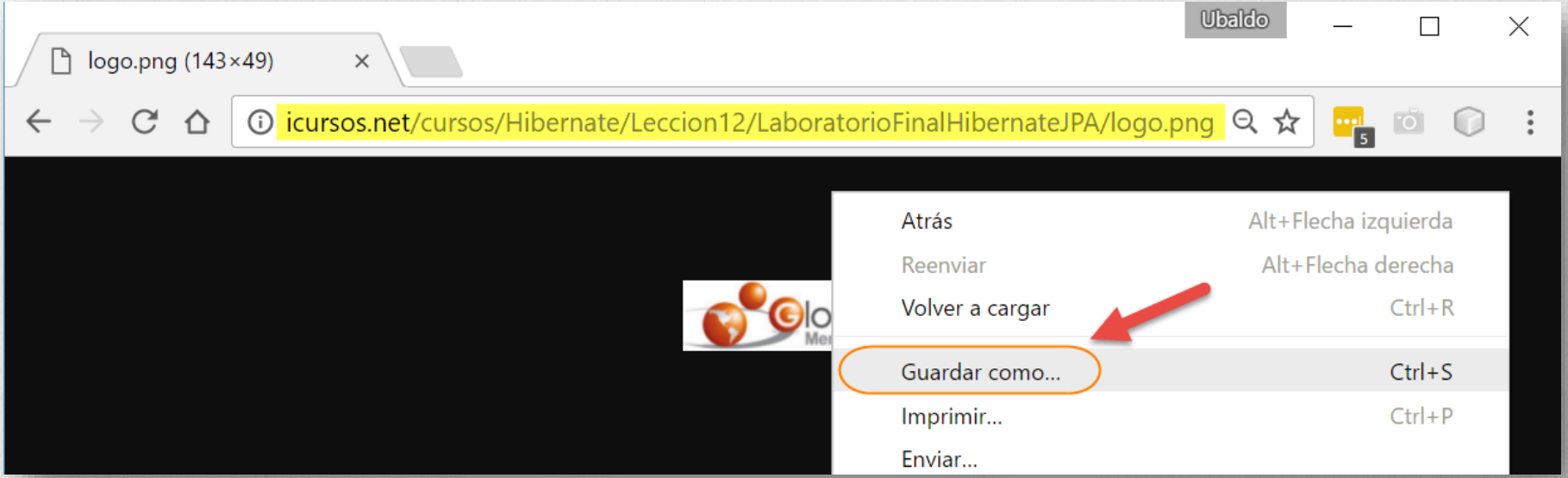
Creamos el archivo estilos.css:



PASO 7. COPIAMOS UN ARCHIVO PNG

Descargamos el archivo el archivo logo.png:

http://icursos.net/cursos/Hibernate/Leccion12/LaboratorioFinalHibernateJPA/logo.png

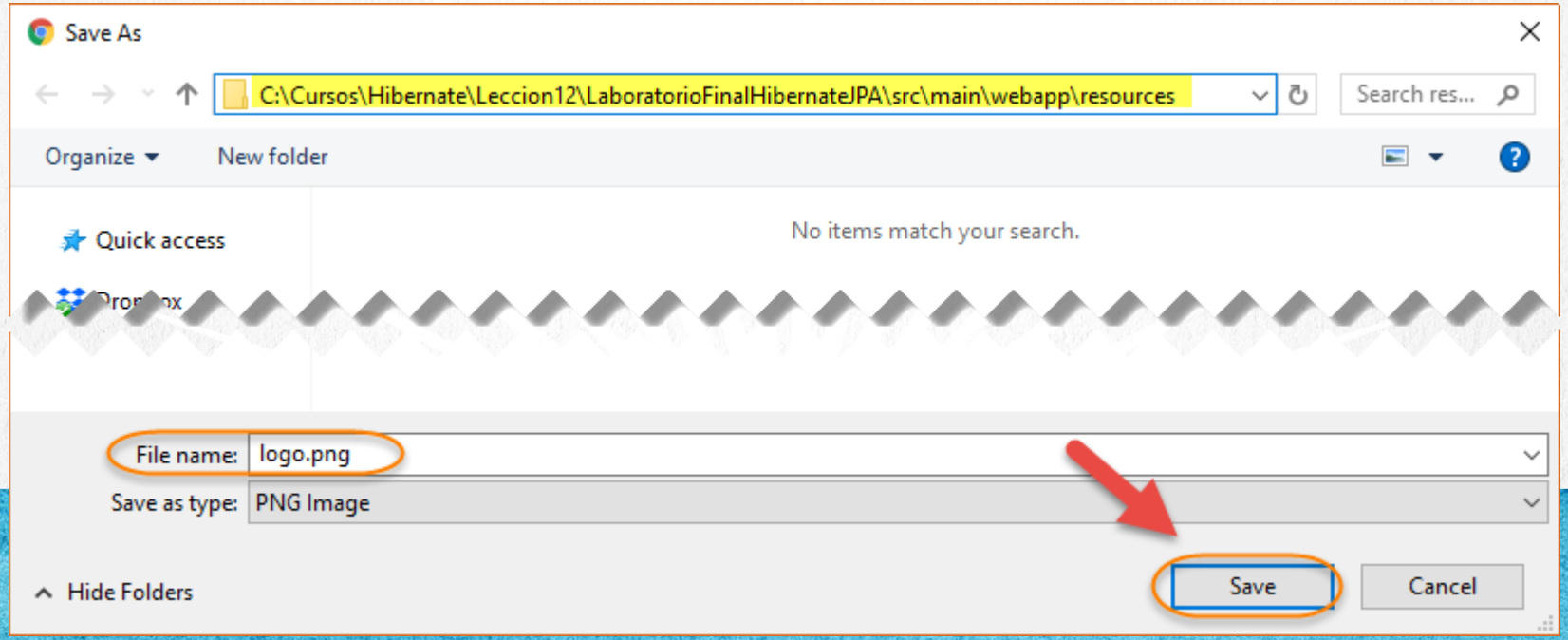


PASO 7. COPIAMOS UN ARCHIVO PNG

Guardamos la imagen en la carpeta de resources recién creada del proyecto:

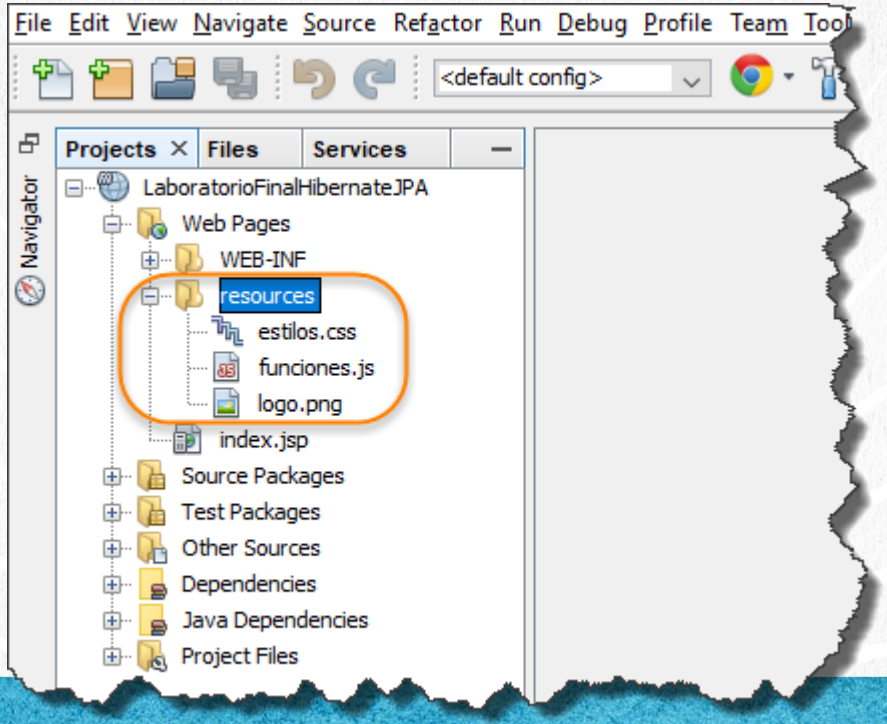
C:\Cursos\Hibernate\Leccion12\LaboratorioFinalHibernateJPA\src\main\webapp\resources

El valor de la ruta puede variar según la ubicación real del proyecto:



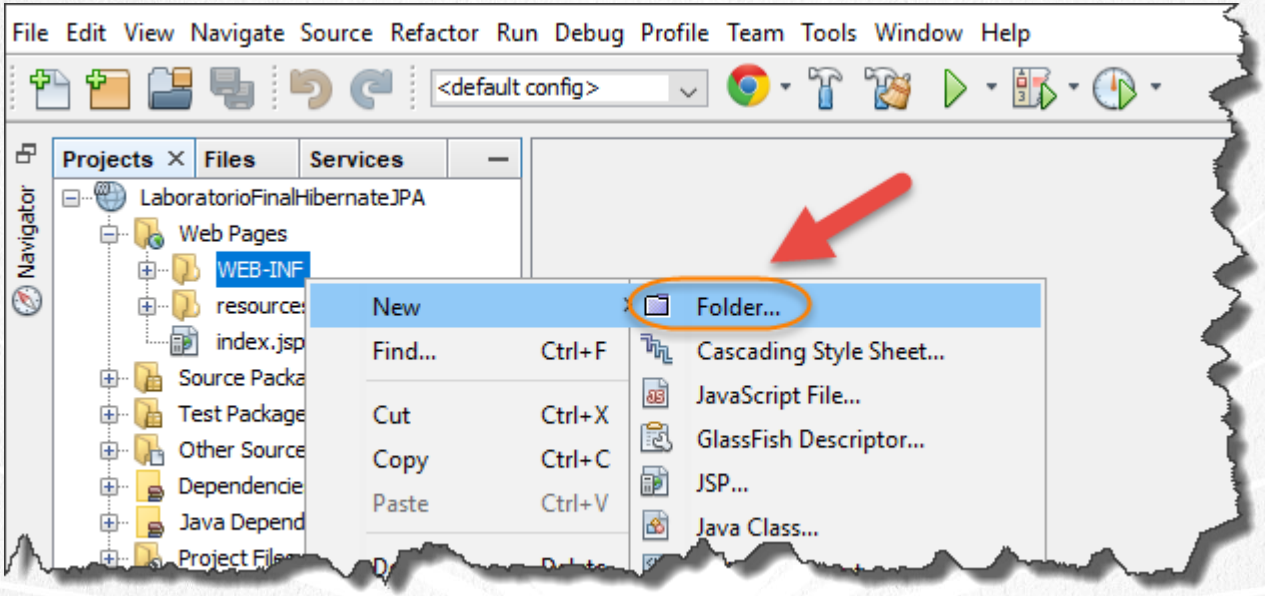
PASO 7. COPIAMOS UN ARCHIVO PNG

Así queda la carpeta de resources:



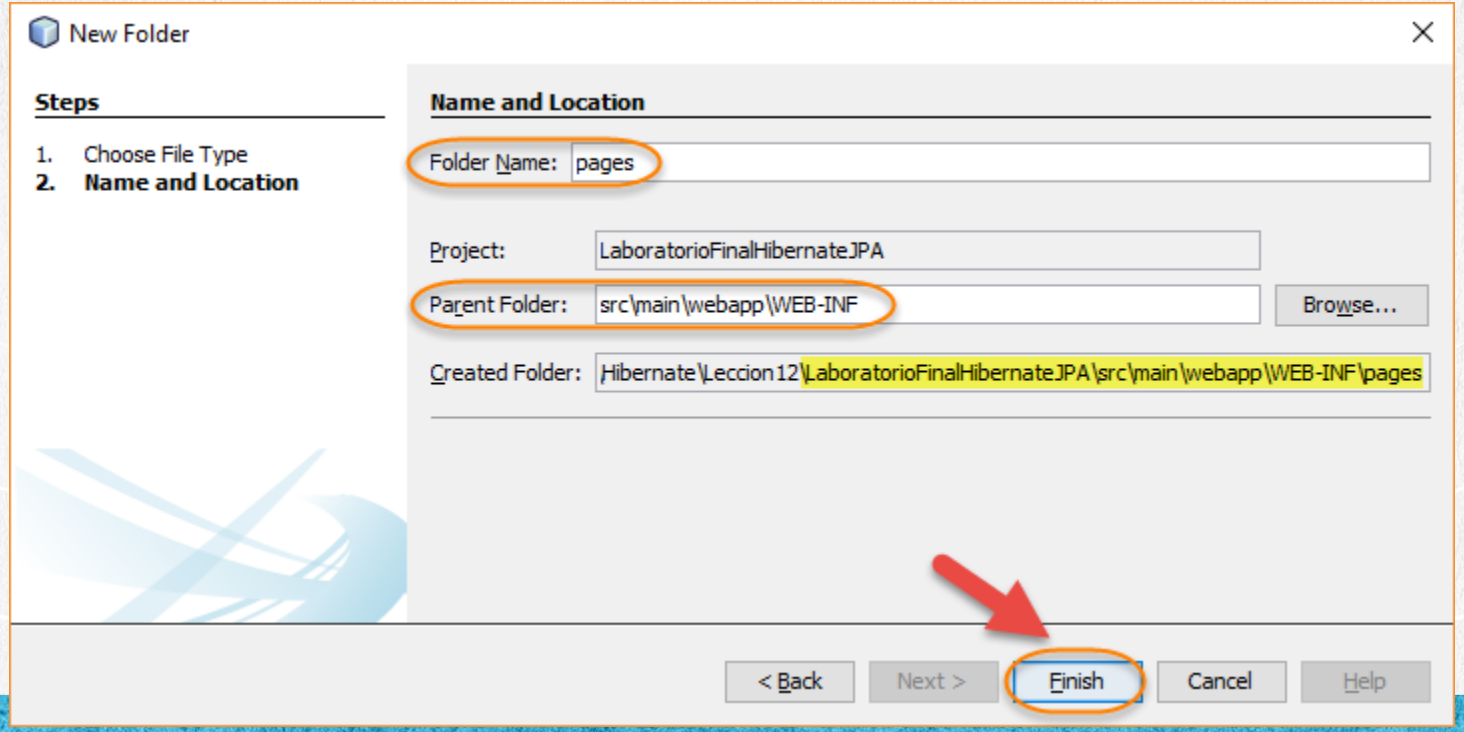
PASO 8. CREAMOS UNA CARPETA

Crear la carpeta llamada pages:



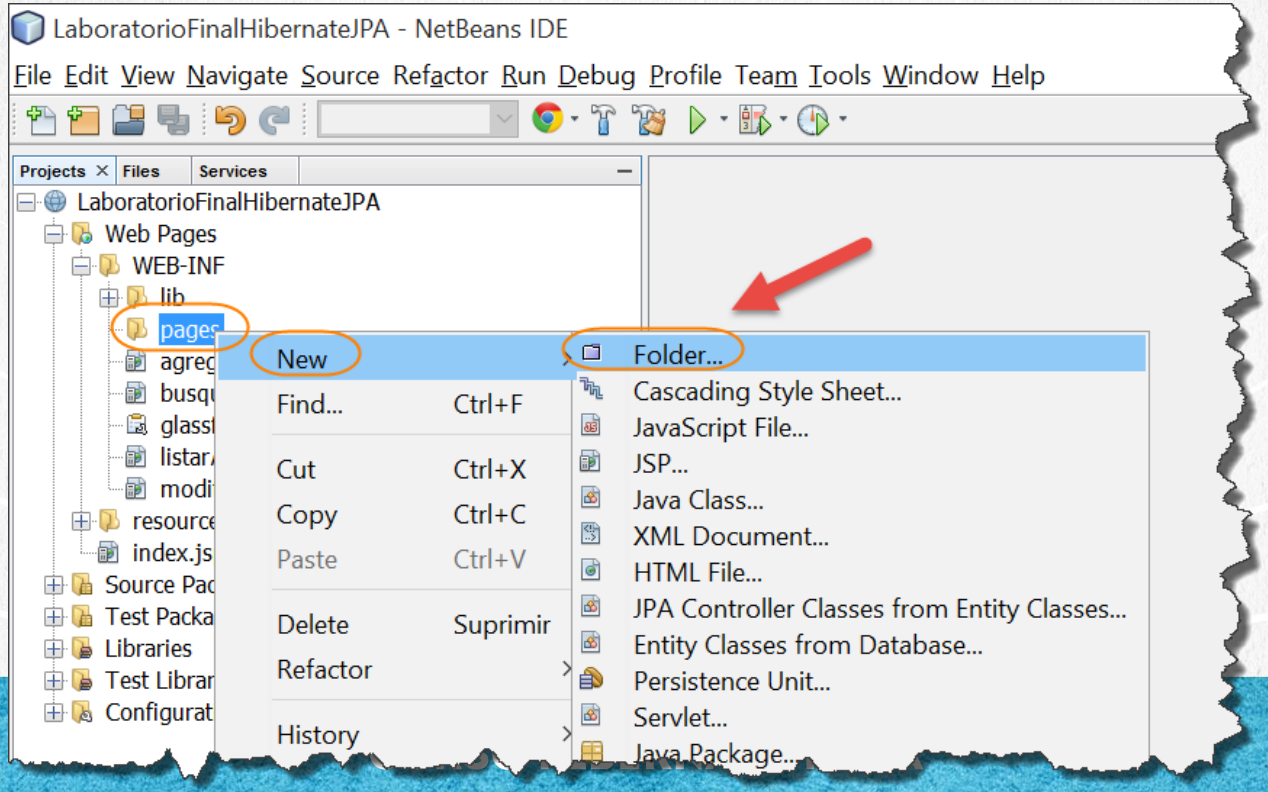
PASO 8. CREAMOS UNA CARPETA

Crear la carpeta llamada pages:



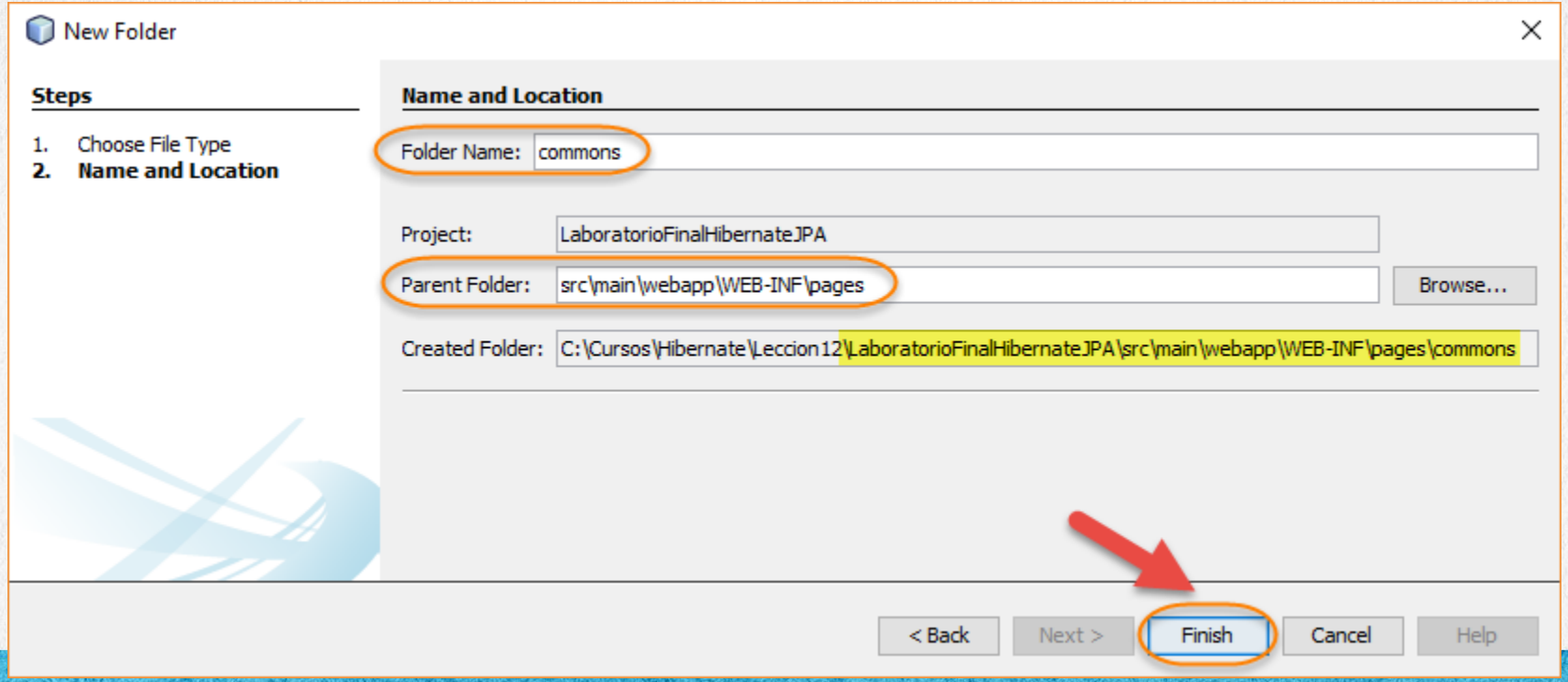
PASO 9. CREAMOS UNA CARPETA

Crear la carpeta de commons:



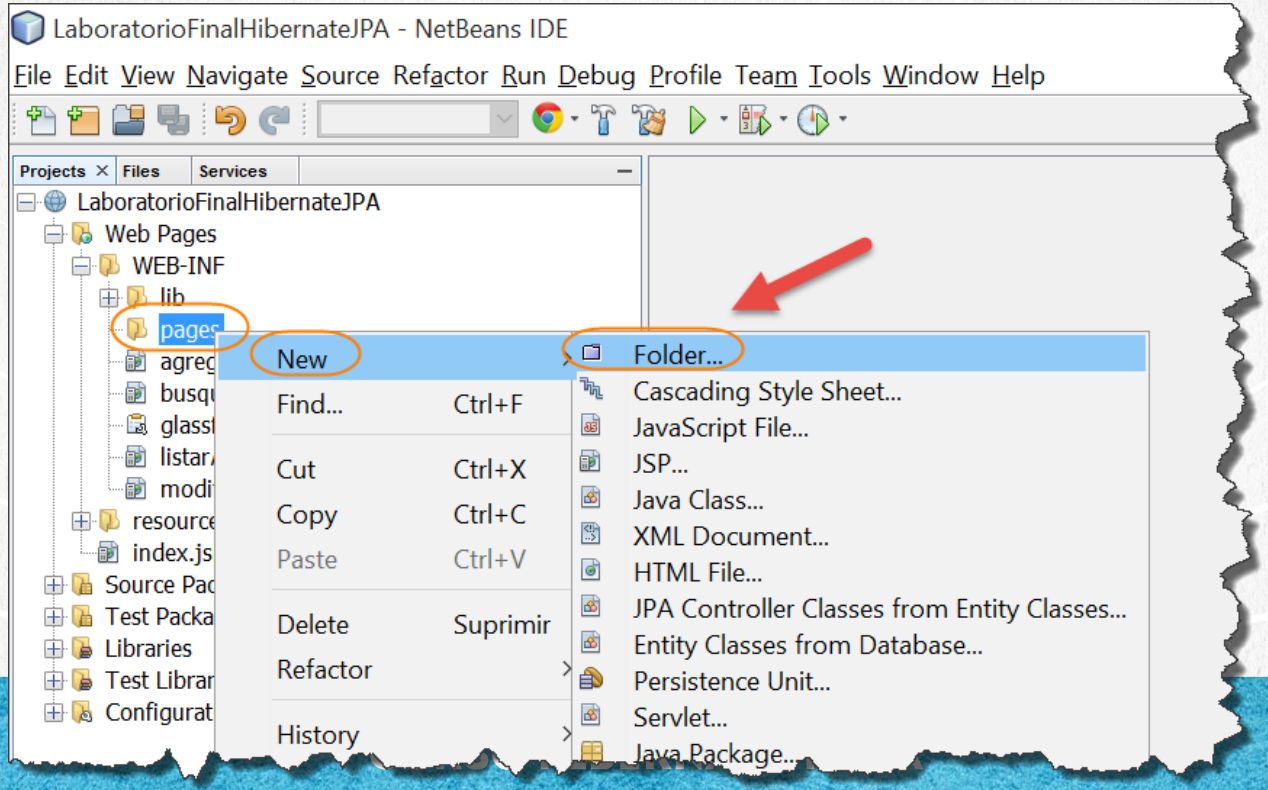
PASO 9. CREAMOS UNA CARPETA

Crear la carpeta de commons:



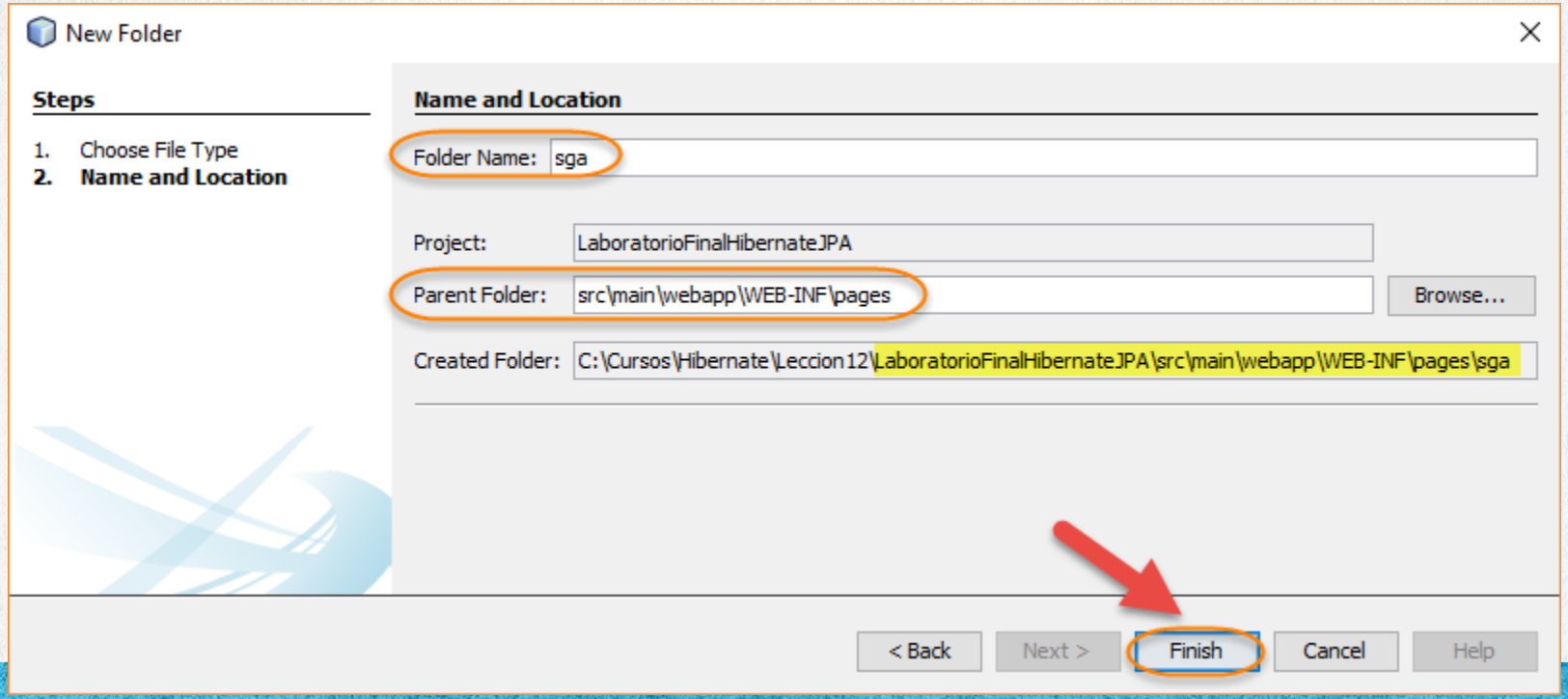
PASO 10. CREAMOS UNA CARPETA

Crear la carpeta de sga:



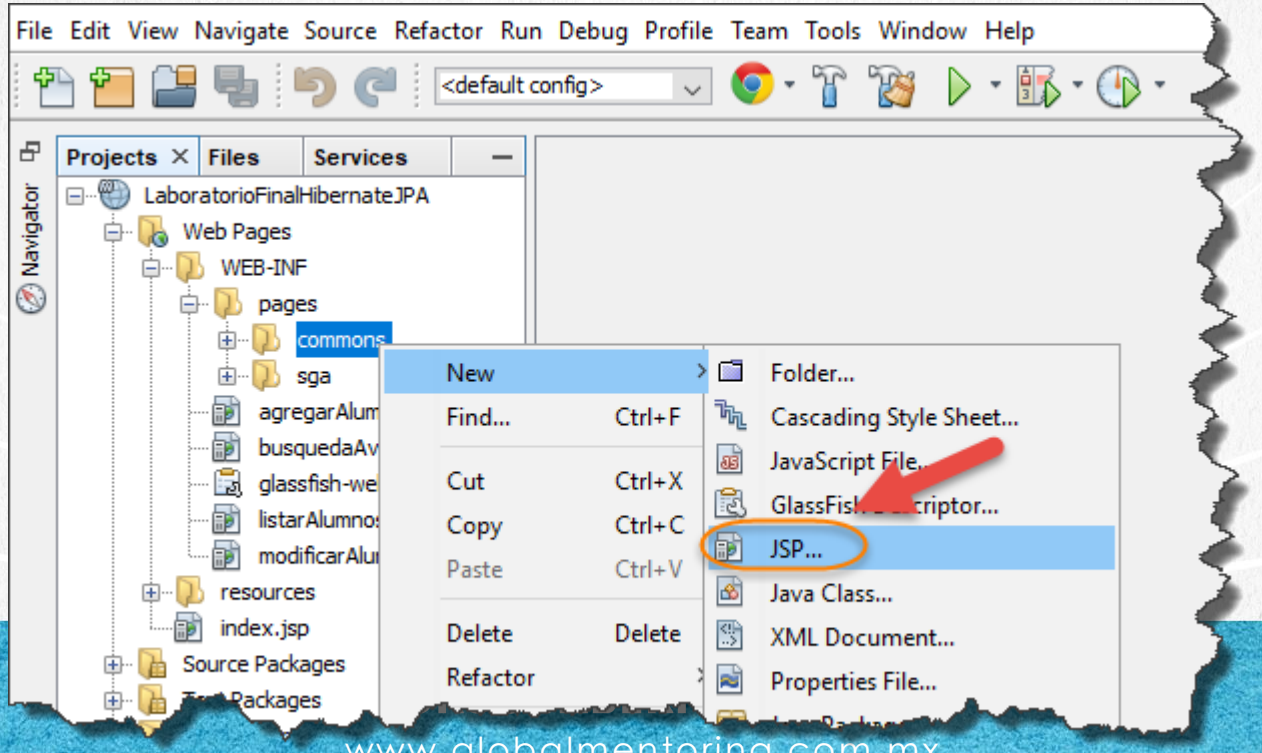
PASO 10. CREAMOS UNA CARPETA

Crear la carpeta de sga:



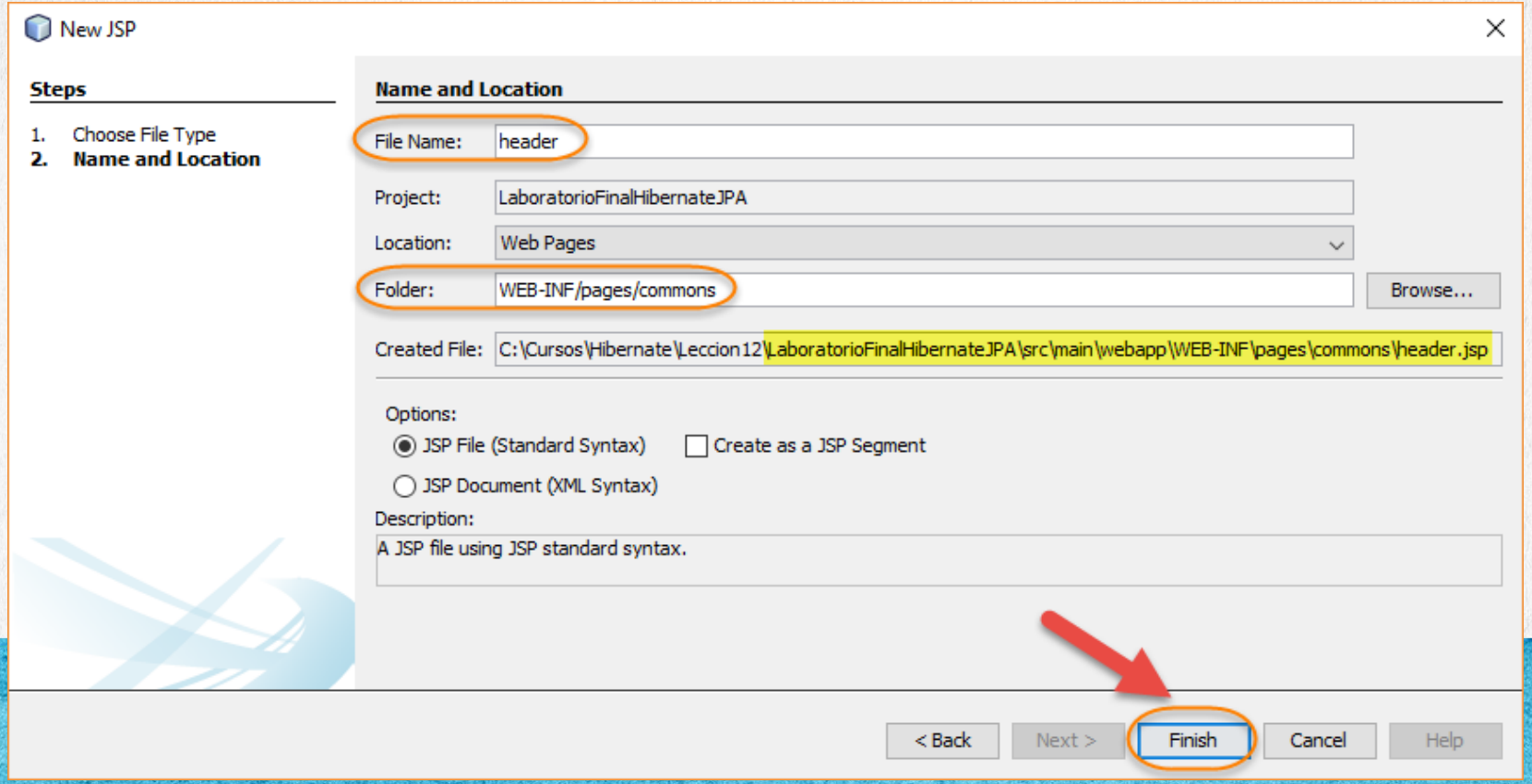
PASO 11. CREAMOS UN ARCHIVO JSP

Creamos varios archivos jsp, en la carpeta de commons, iniciamos con el archivo header.jsp:



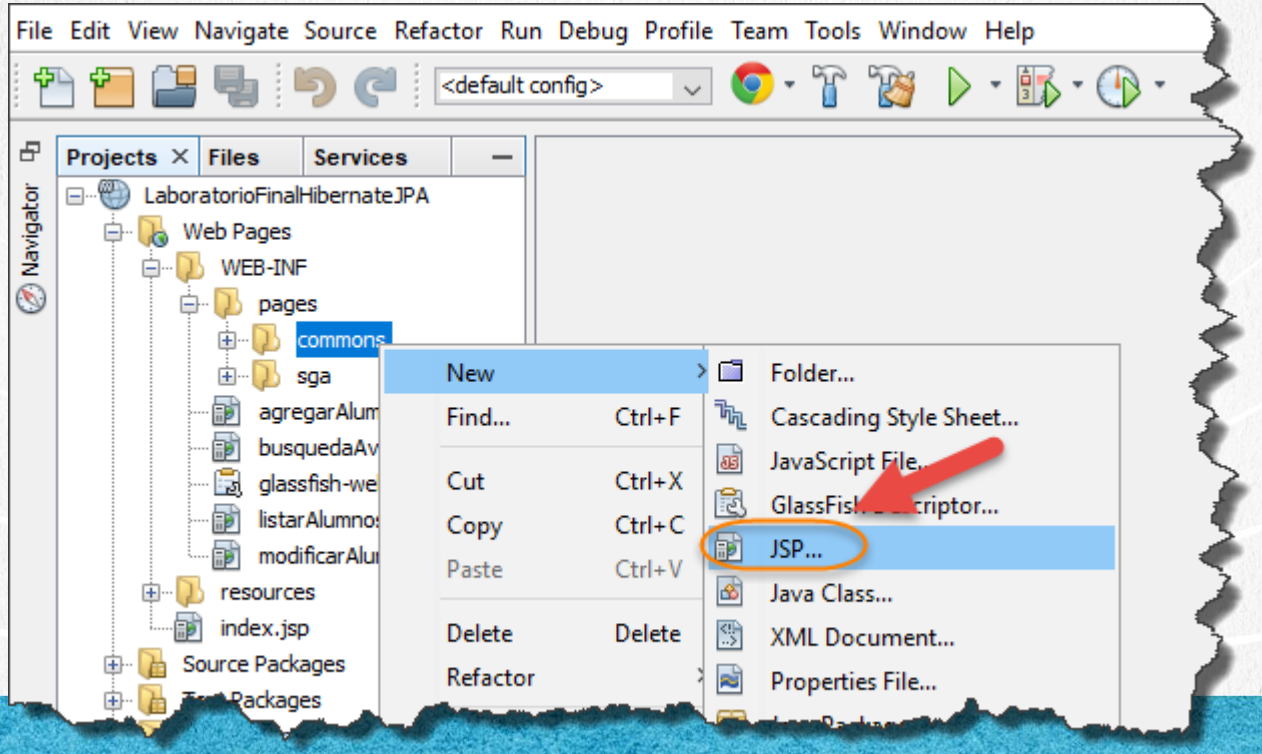
PASO 11. CREAMOS UN ARCHIVO JSP

Crear un archivo header.jsp:



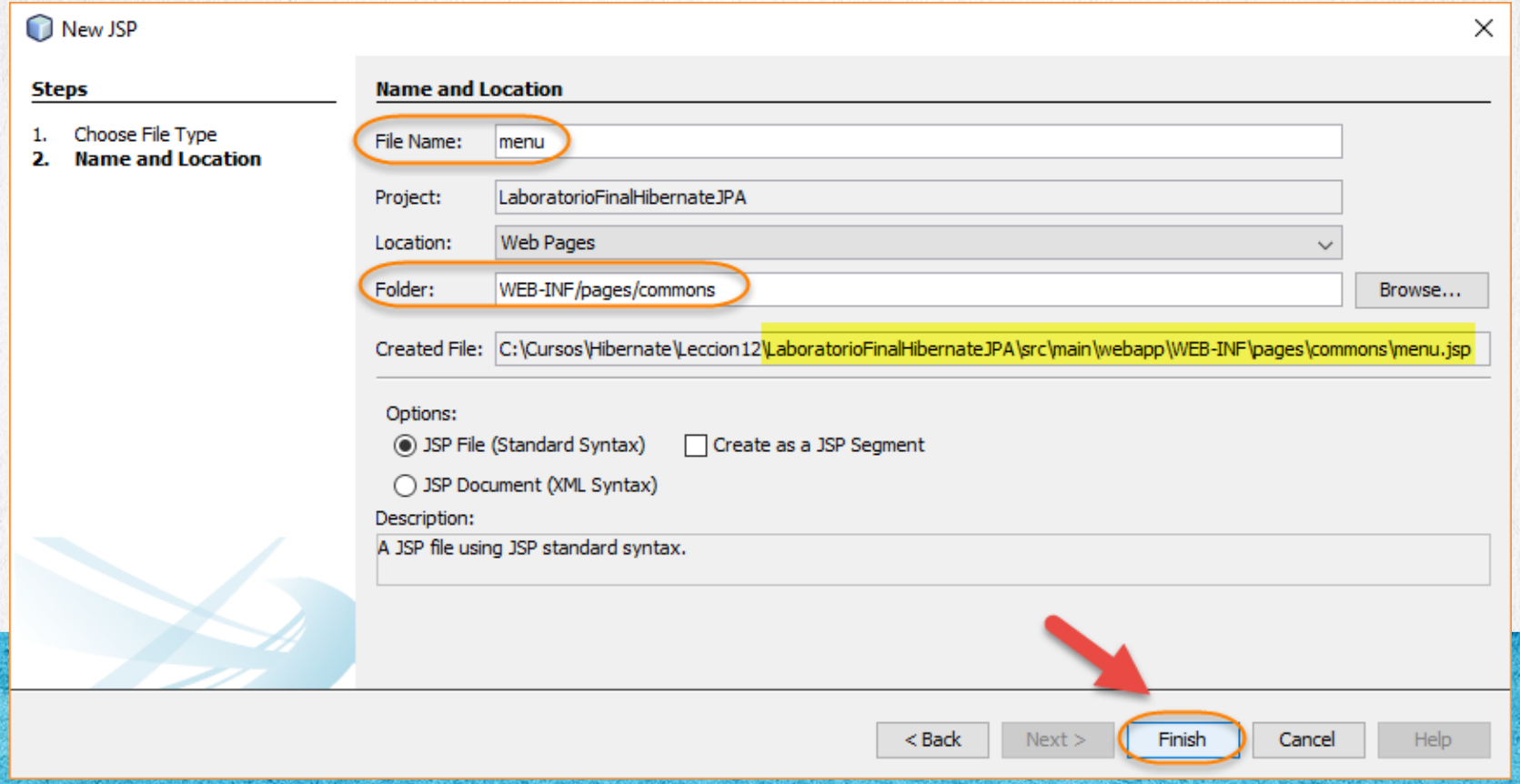
PASO 12. CREAMOS UN ARCHIVO JSP

Crear un archivo menu.jsp:



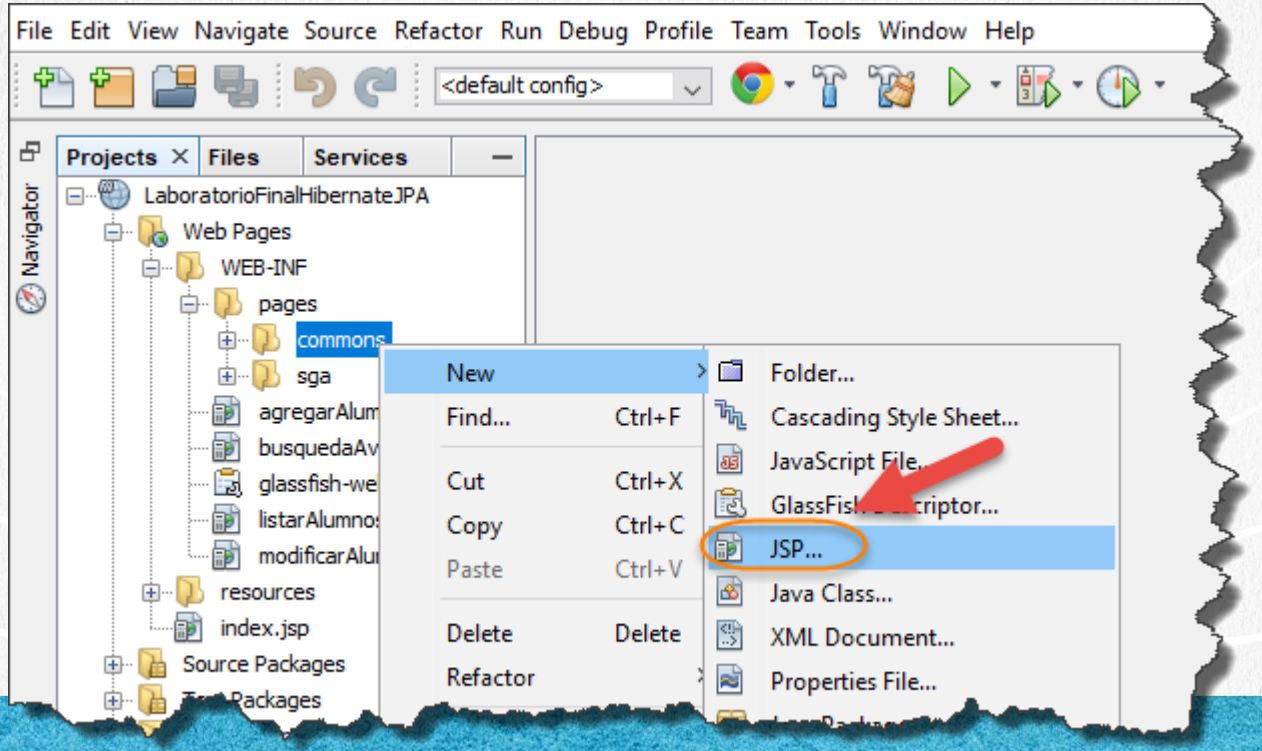
PASO 12. CREAMOS UN ARCHIVO JSP

Crear un archivo menu.jsp:



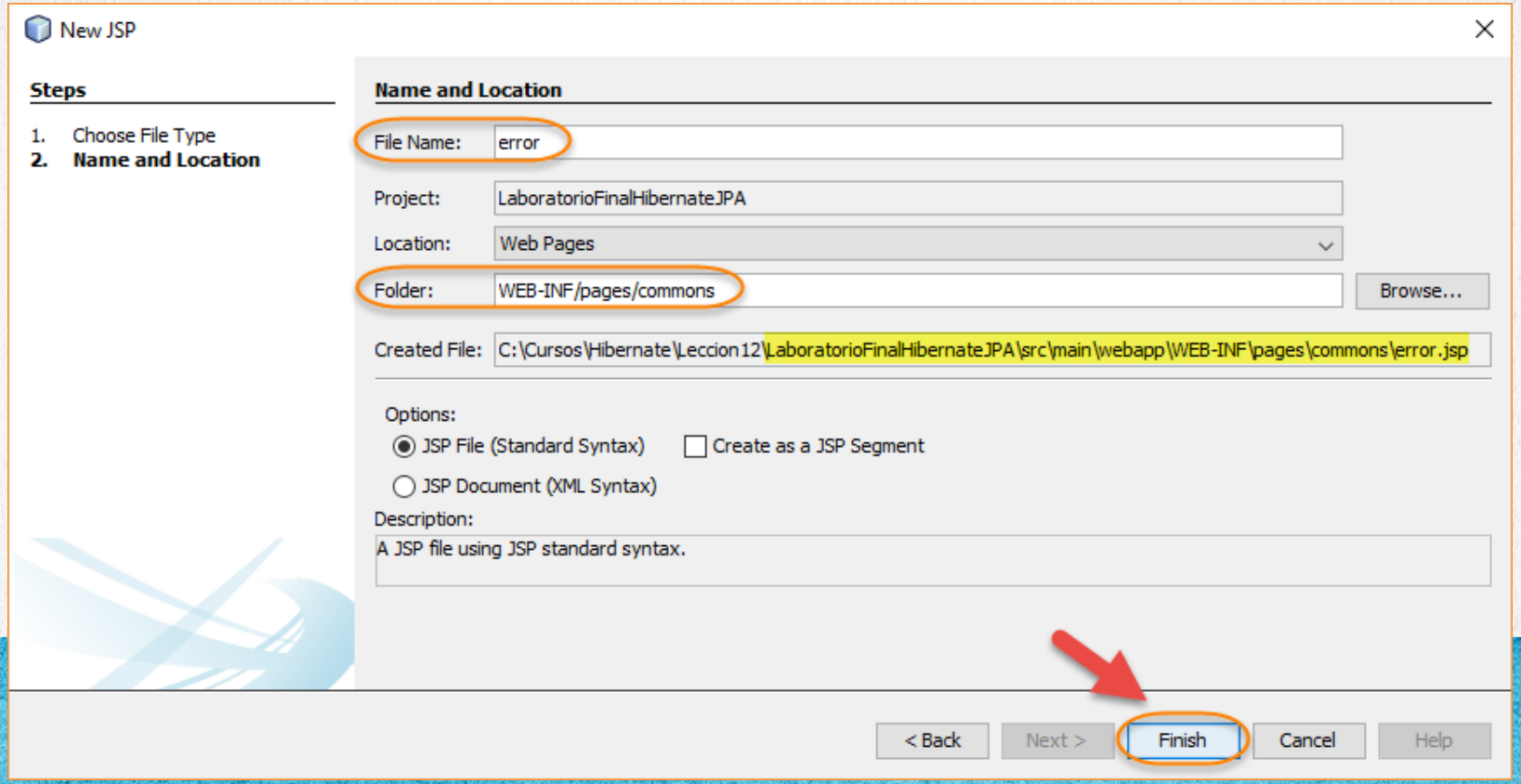
PASO 13. CREAMOS UN ARCHIVO JSP

Crear un archivo error.jsp:



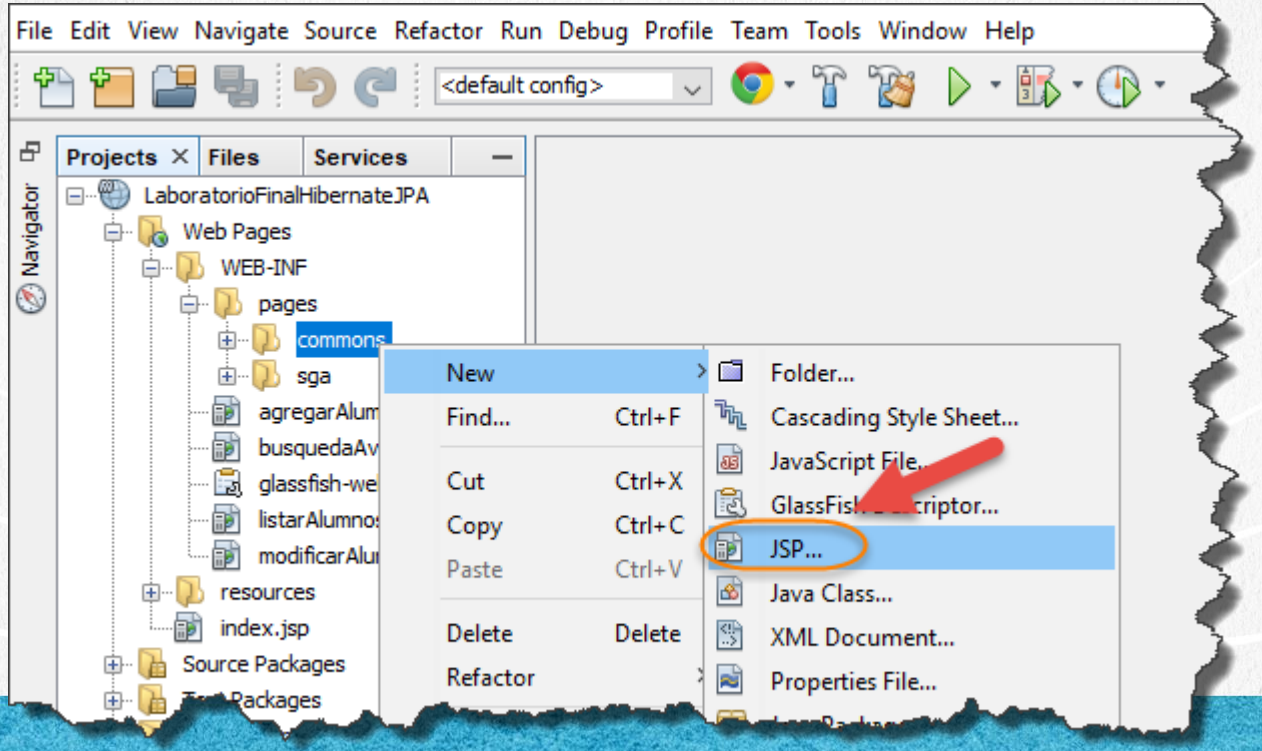
PASO 13. CREAMOS UN ARCHIVO JSP

Crear un archivo error.jsp:



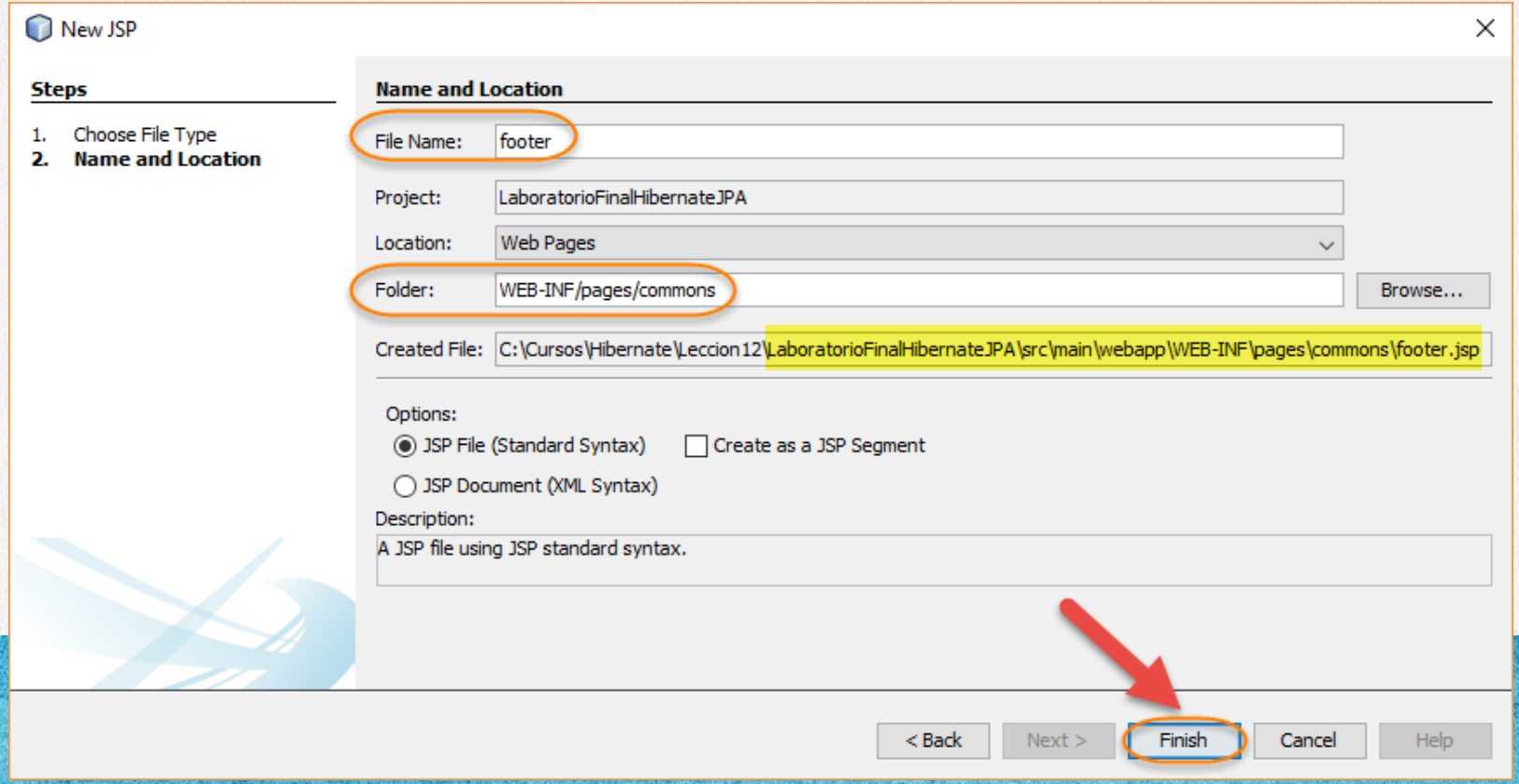
PASO 14. CREAMOS UN ARCHIVO JSP

Crear un archivo footer.jsp:



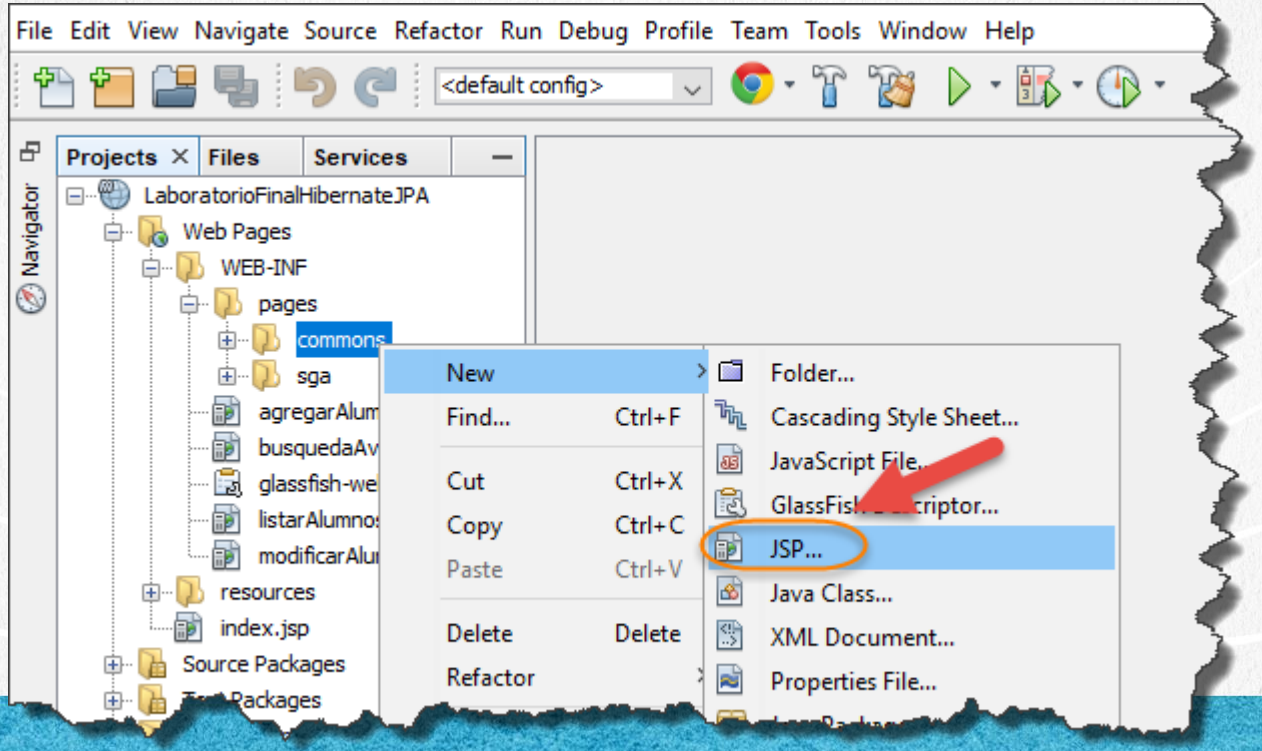
PASO 14. CREAMOS UN ARCHIVO JSP

Crear un archivo footer.jsp:



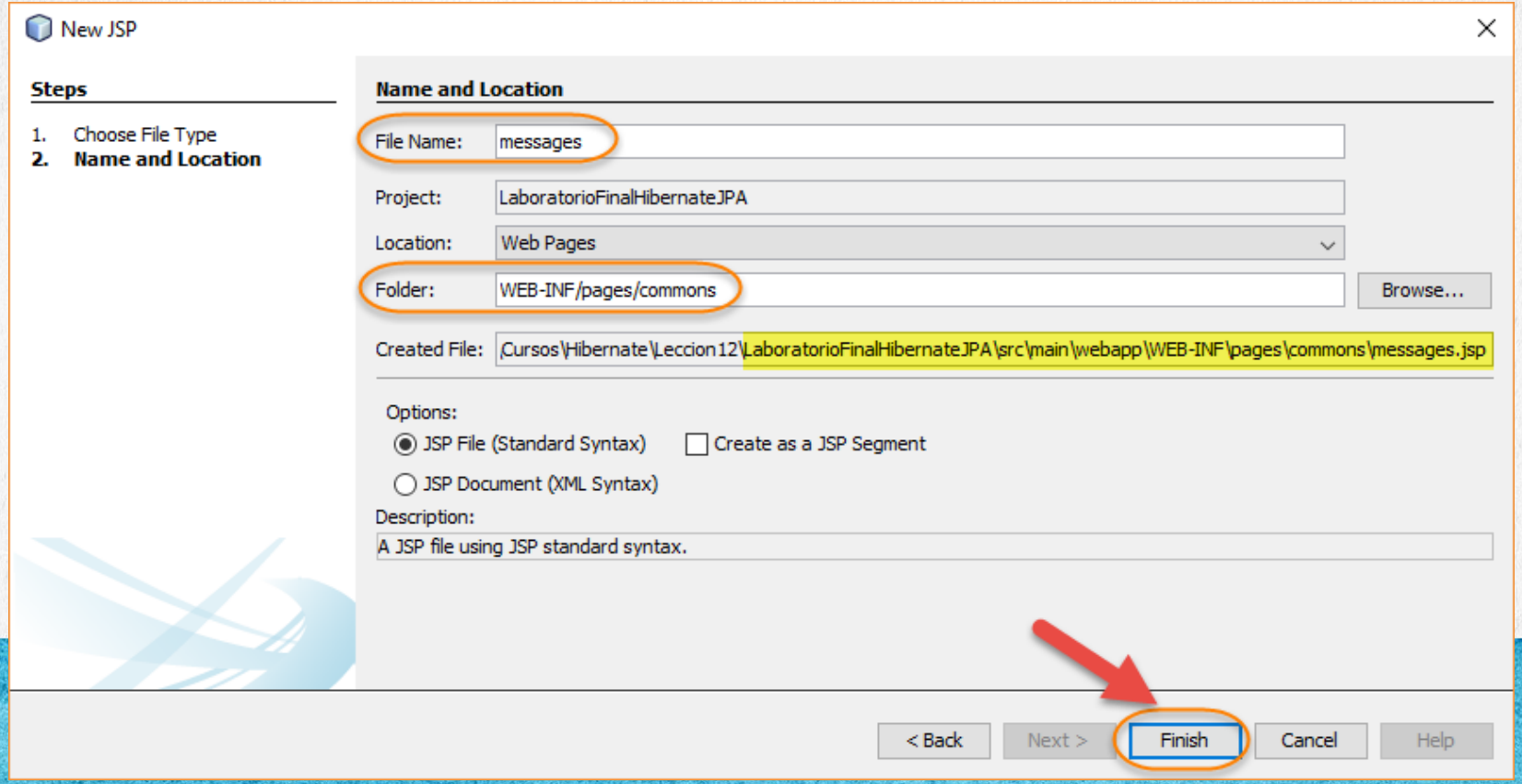
PASO 15. CREAMOS UN ARCHIVO JSP

Crear un archivo messages.jsp:



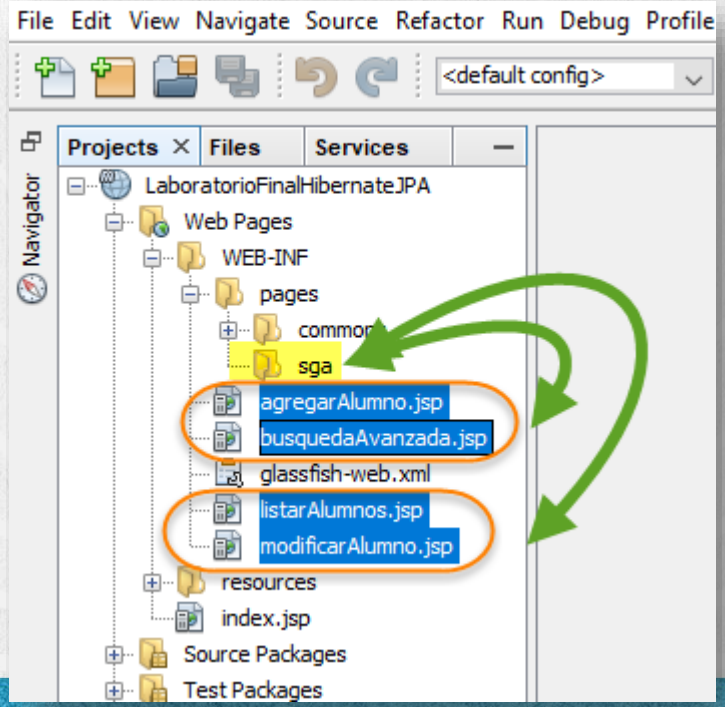
PASO 15. CREAMOS UN ARCHIVO JSP

Crear un archivo messages.jsp:



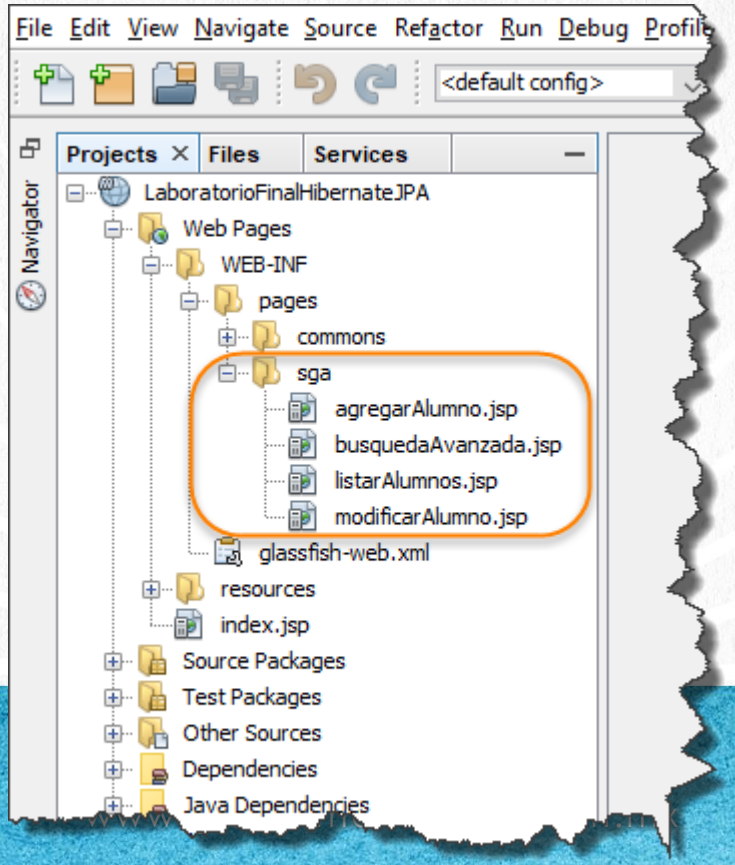
PASO 16. MOVEMOS UNOS ARCHIVOS JSP

Movemos los archivos jsp siguientes a la carpeta de sga (los arrastramos):



PASO 16. MOVEMOS UNOS ARCHIVOS JSP

El resultado de mover los archivos queda así:



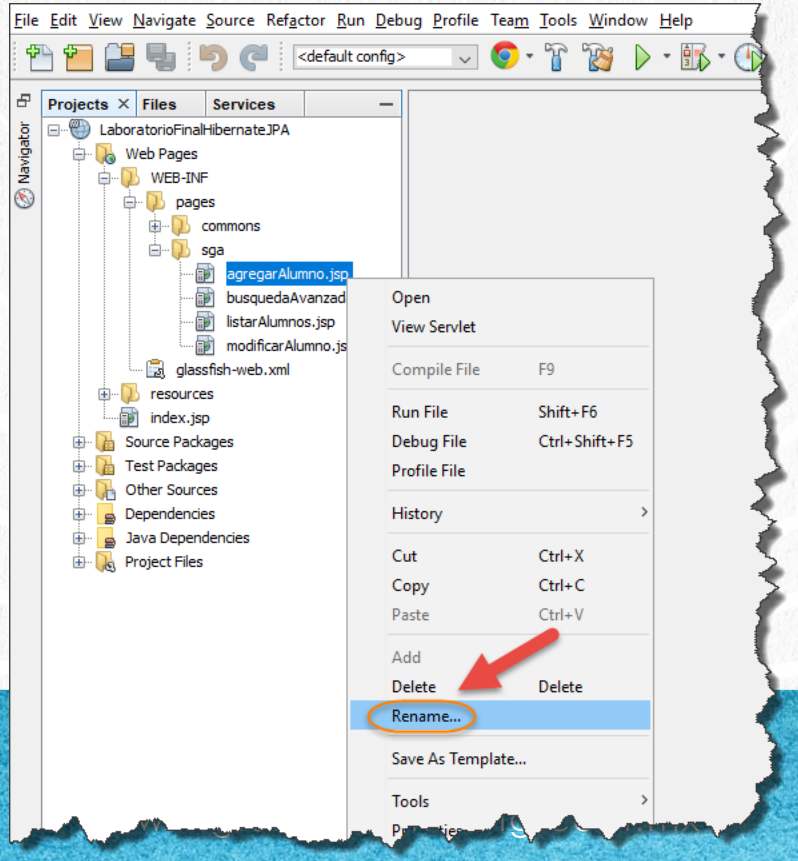
PASO 17. RENOMBRAMOS ALGUNOS ARCHIVOS JSP

Renombramos los siguientes archivos jsp de la carpeta sga:

- listarAlumnos.jsp (queda igual, no se va a renombrar)
- agregarAlumno.jsp lo renombramos a: detalleAlumno.jsp
- busquedaAvanzada.jsp lo renombramos a: buscarAlumnos.jsp
- modificarAlumno.jsp lo eliminamos (click derecho -> delete).

PASO 17. RENOMBRAMOS ALGUNOS ARCHIVOS JSP

Renombramos el archivo agregarAlumno.jsp a detalleAlumno.jsp:

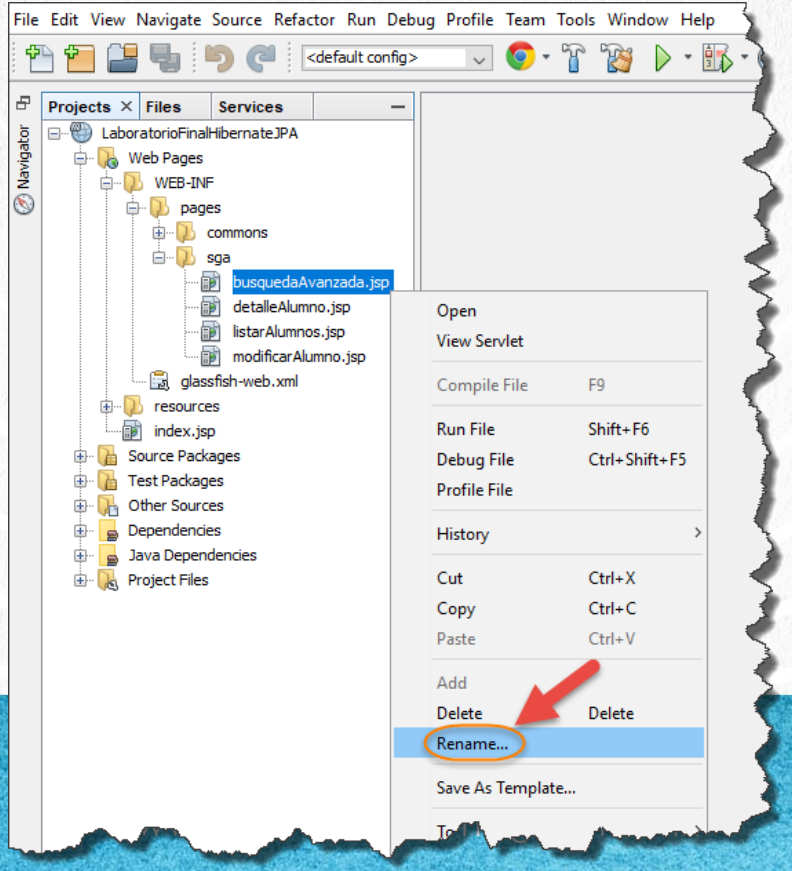


PASO 17. RENOMBRAMOS ALGUNOS ARCHIVOS JSP

Renombramos el archivo agregarAlumno.jsp a detalleAlumno.jsp:

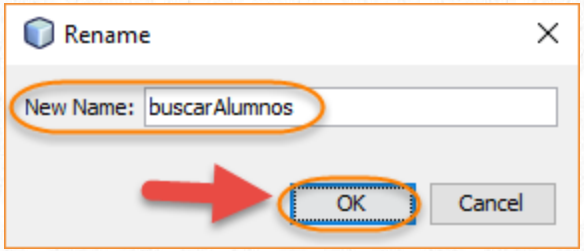
PASO 17. RENOMBRAMOS ALGUNOS ARCHIVOS JSP

Renombramos el archivo busquedaAvanzada.jsp a buscarAlumnos.jsp:



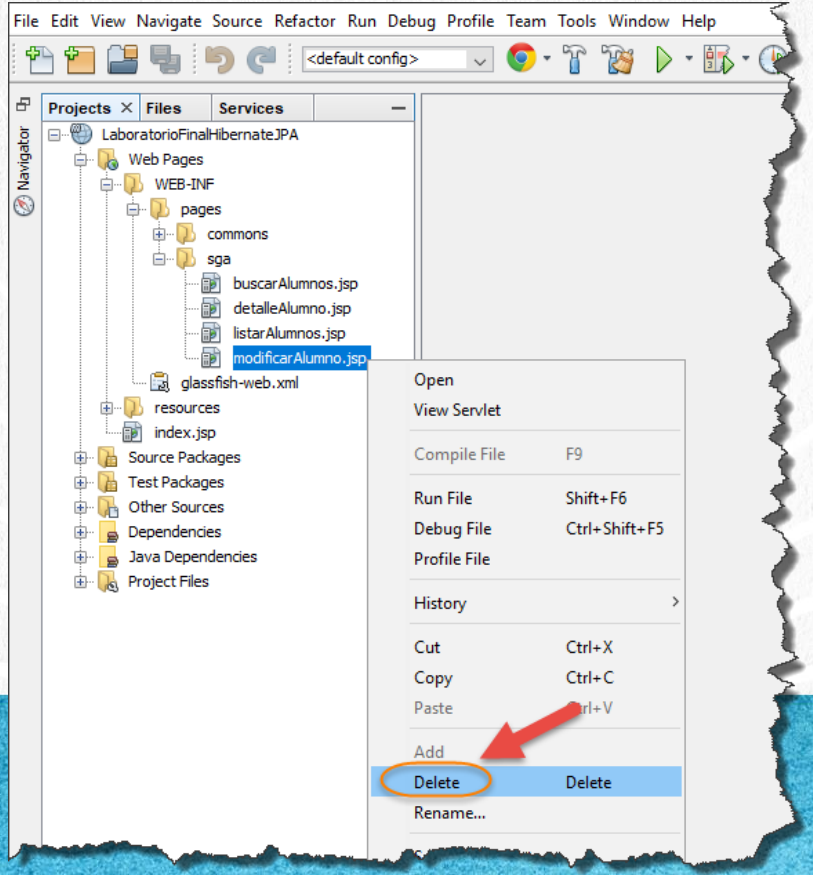
PASO 17. RENOMBRAMOS ALGUNOS ARCHIVOS JSP

Renombramos el archivo busquedaAvanzada.jsp a buscarAlumnos.jsp:



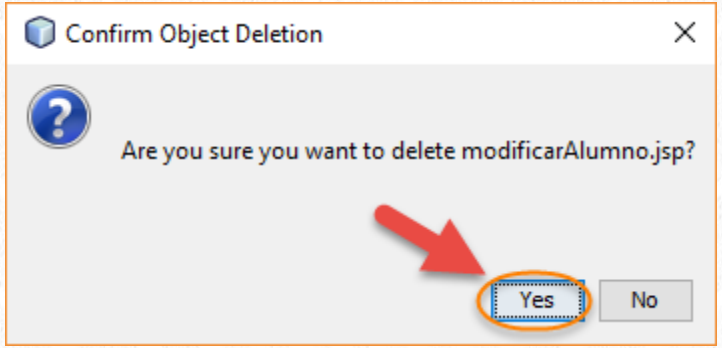
PASO 17. RENOMBRAMOS ALGUNOS ARCHIVOS JSP

Eliminamos el archivo modificarAlumno.jsp:



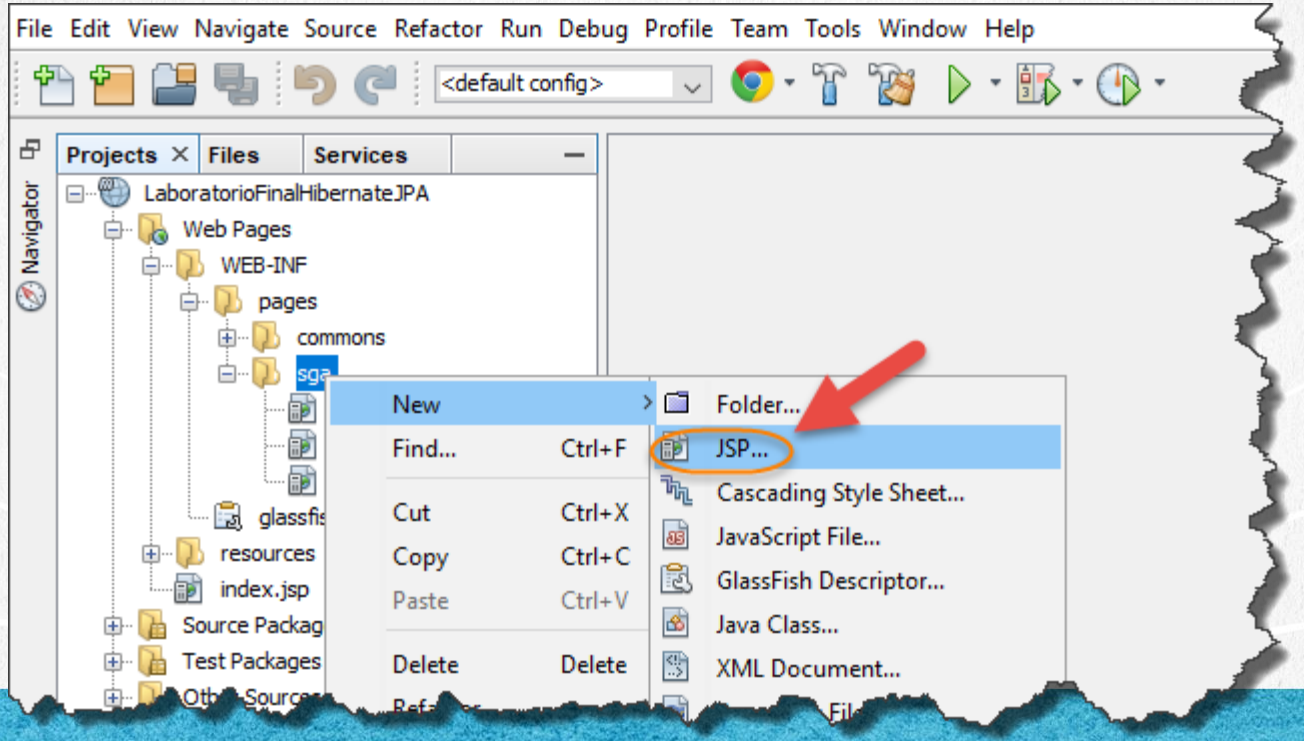
PASO 17. RENOMBRAMOS ALGUNOS ARCHIVOS JSP

Eliminamos el archivo modificarAlumno.jsp:



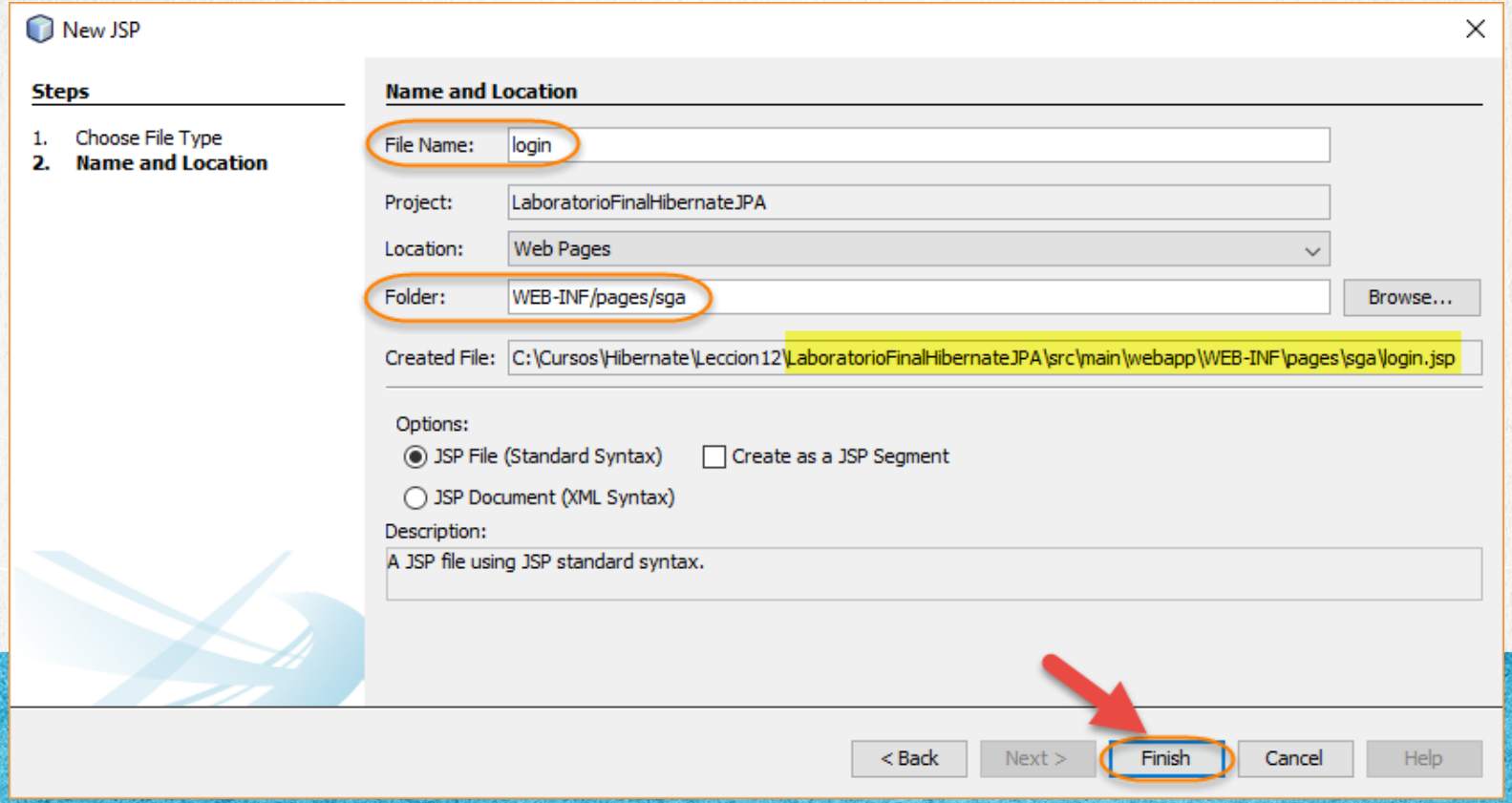
PASO 18. AGREGAMOS UN JSP

Agregamos el archivo login.jsp a la carpeta de sga:



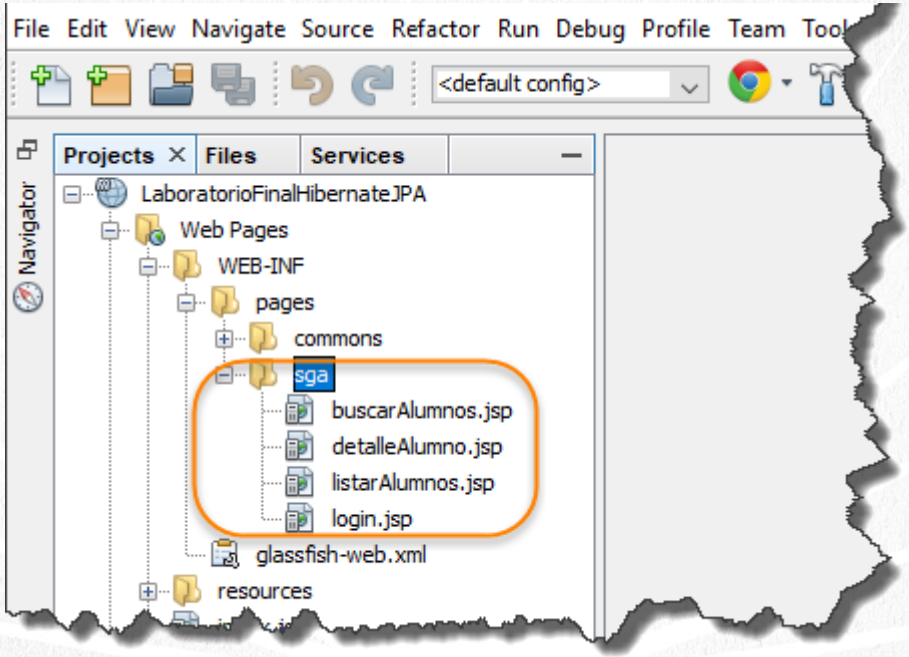
PASO 18. AGREGAMOS UN JSP

Agregamos el archivo login.jsp a la carpeta de sga:



PASO 18. AGREGAMOS UN JSP

Así debe quedar la carpeta sga:



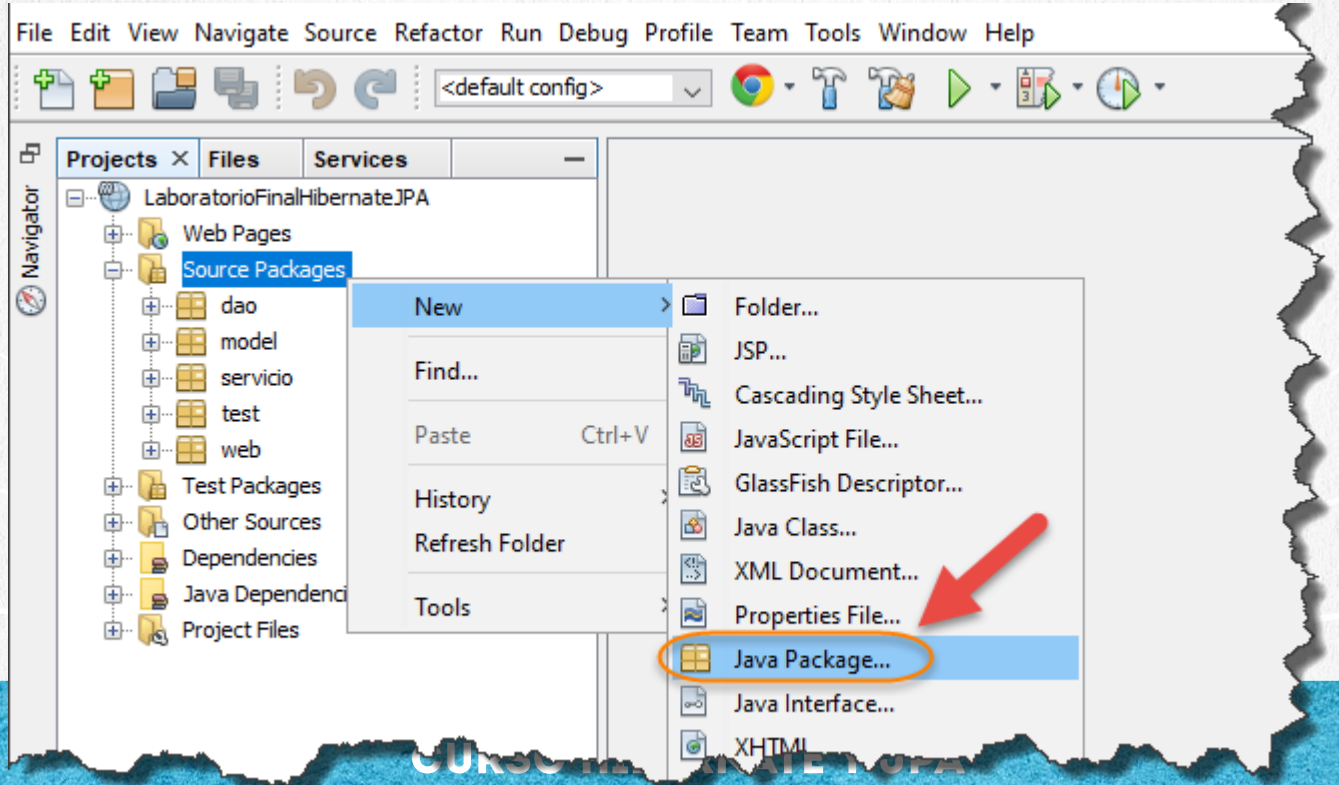
PASO 19. CREAMOS PAQUETES JAVA

Creamos los siguientes paquetes Java:

- com.gm.sga.capaweb
- com.gm.sga.capaservicio
- com.gm.sga.capadatos.dto
- com.gm.sga.capadatos.dao

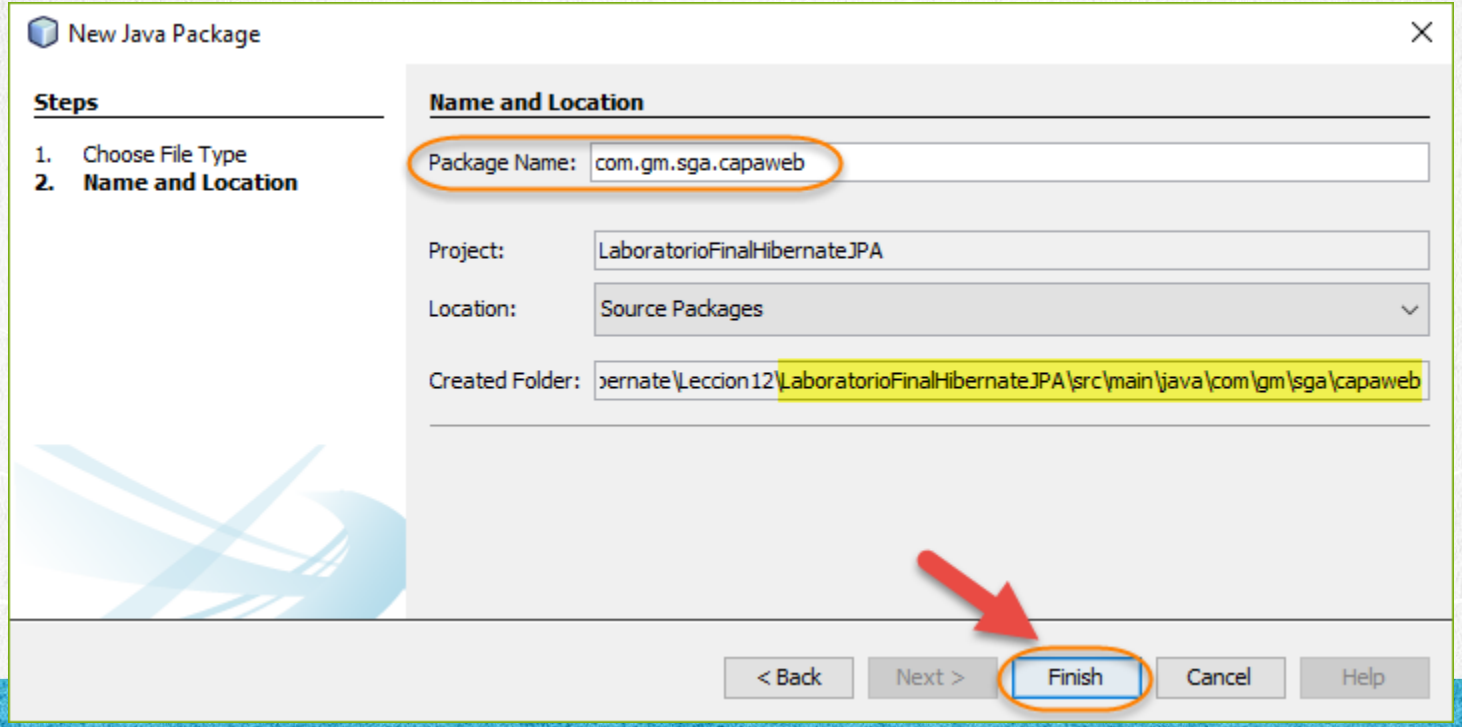
PASO 19. CREAMOS PAQUETES JAVA

Creamos el paquete Java com.gm.sga.capaweb:



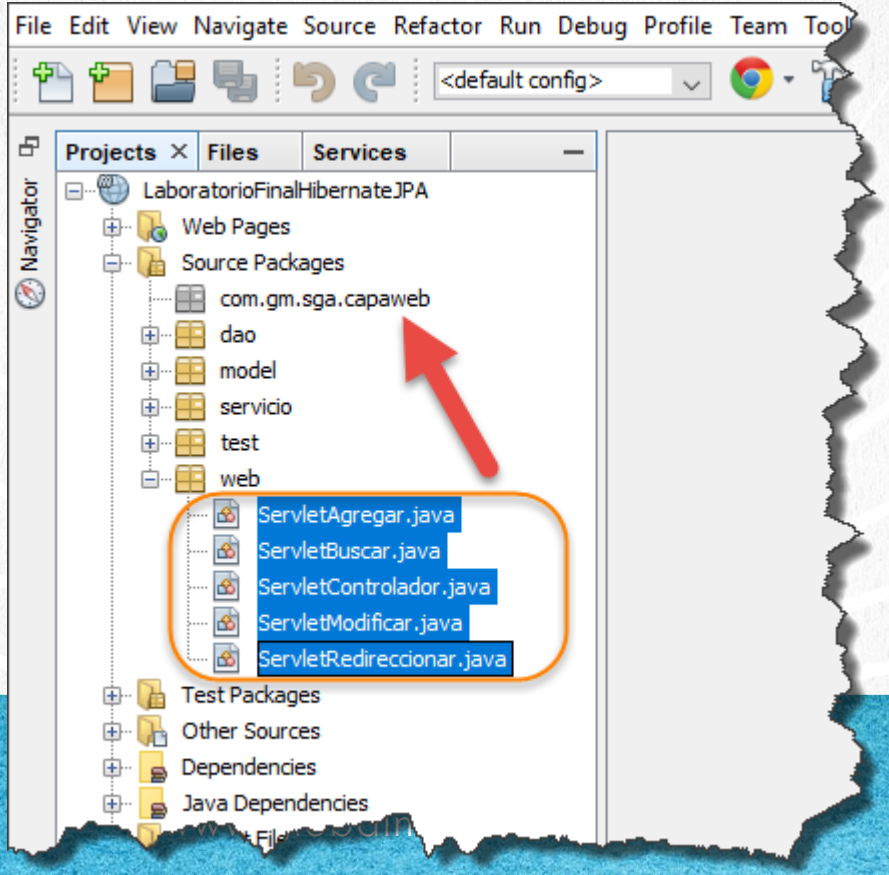
PASO 19. CREAMOS PAQUETES JAVA

Creamos el paquete Java com.gm.sga.capaweb:



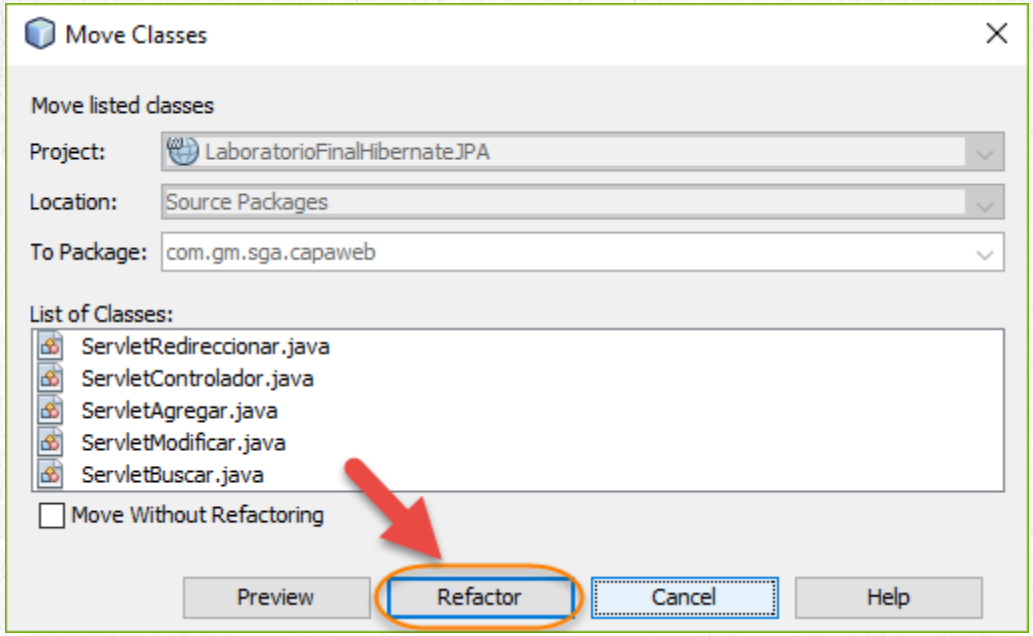
PASO 20. MOVEMOS LAS CLASES JAVA

Movemos las clases del paquete web al paquete com.gm.sga.capaweb



PASO 20. MOVEMOS LAS CLASES JAVA

Movemos las clases del paquete web al paquete com.gm.sga.capaweb



PASO 21. ELIMINAMOS UNA CLASE JAVA

Eliminamos la clase de ServletAgregar.java

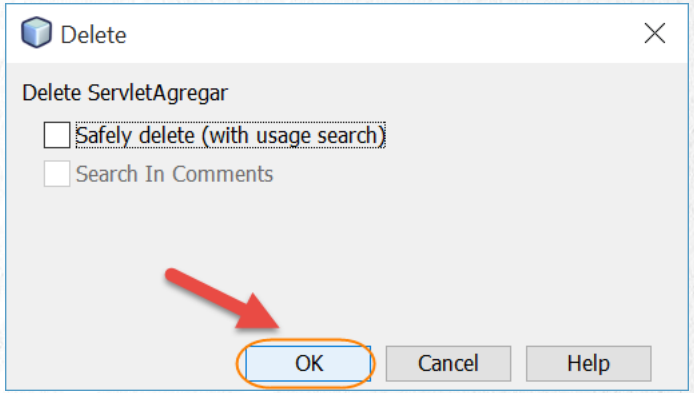
LaboratorioFinalHibernateJPA - NetBeans IDE

File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help

Projects × Files Services

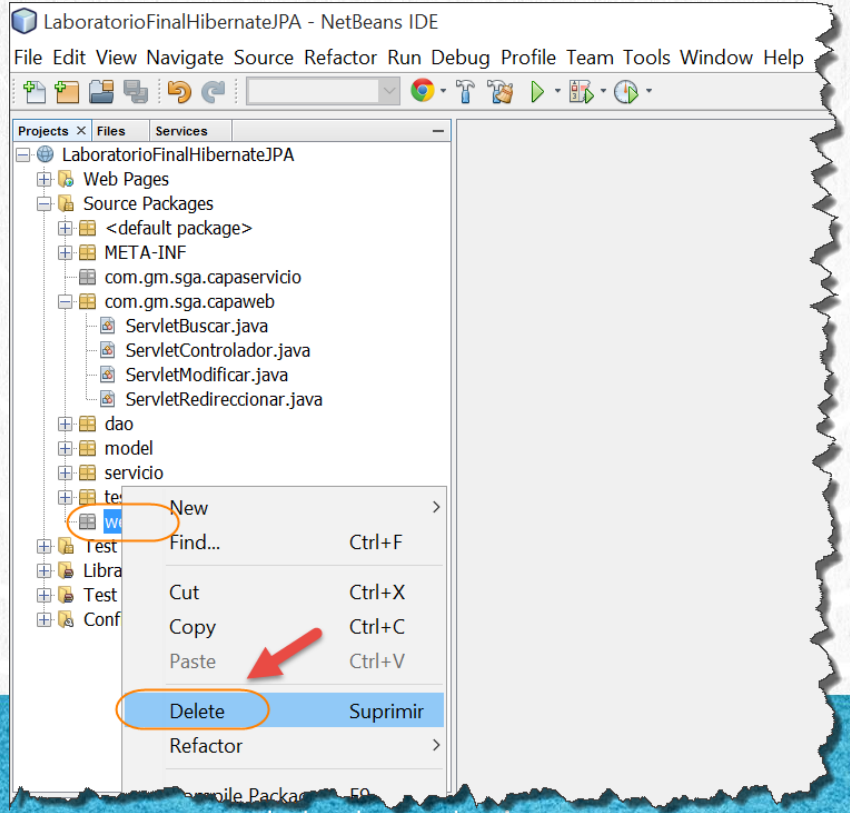
- LaboratorioFinalHiberna
  - Web Pages
  - Source Packages
    - <default package>
    - META-INF
    - com.gm.sga.capav
      - ServletAgregar.
      - ServletBuscar.ja
      - ServletControla
      - ServletModificar
      - ServletRedirecci
    - dao
    - model
    - servicio
    - test
    - web
  - Test Packages
  - Libraries
  - Test Libraries
  - Configuration Files

Open	
Cut	Ctrl+X
Copy	Ctrl+C
Paste	Ctrl+V
Compile File	F9
Run File	Mayús+F6
Debug File	Ctrl+Mayús+F5
Profile File	
Test File	Ctrl+F6
Debug Test File	Ctrl+Mayús+F6
Profile Test File	
Add	
Delete	Suprimir
Save As Template...	
Find Usages	Alt+F7
Refactor	›



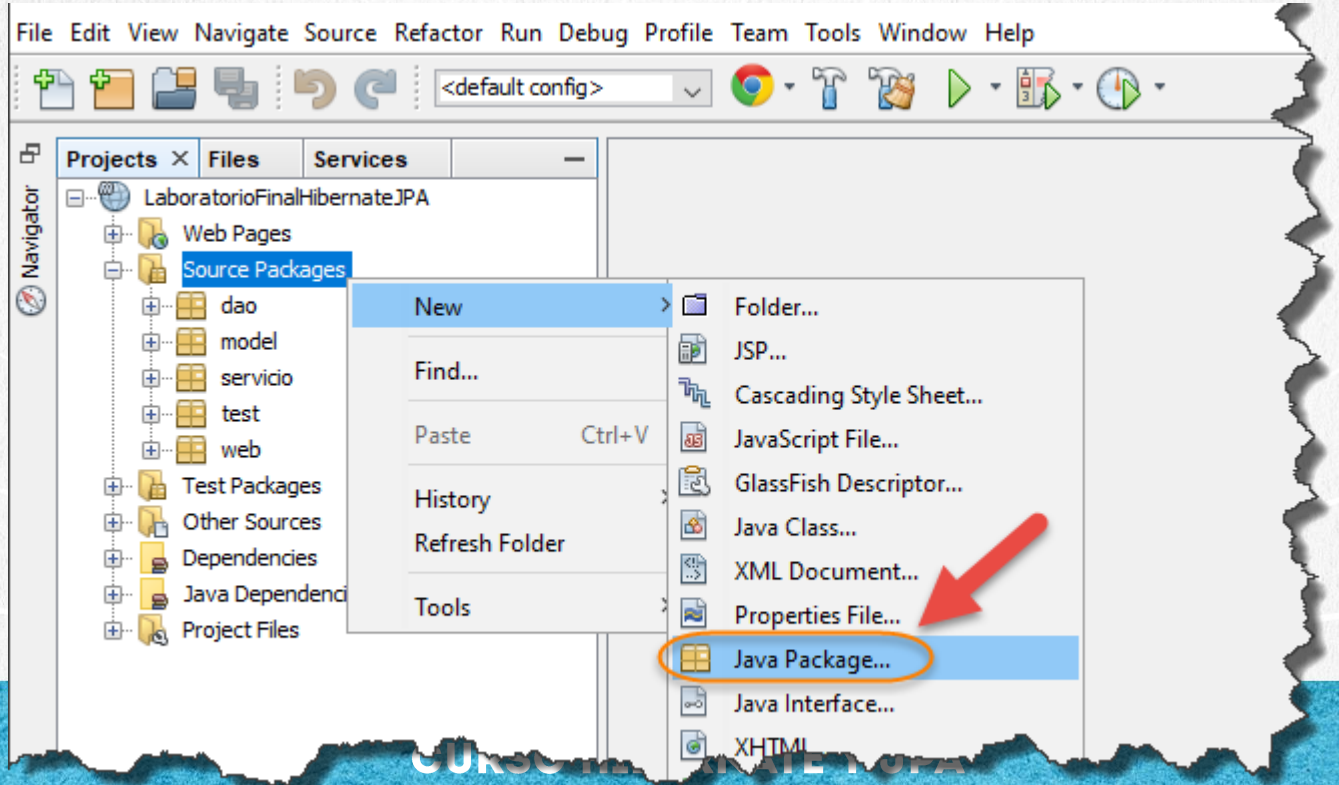
PASO 22. ELIMINAMOS UN PAQUETE

Eliminamos el paquete web:



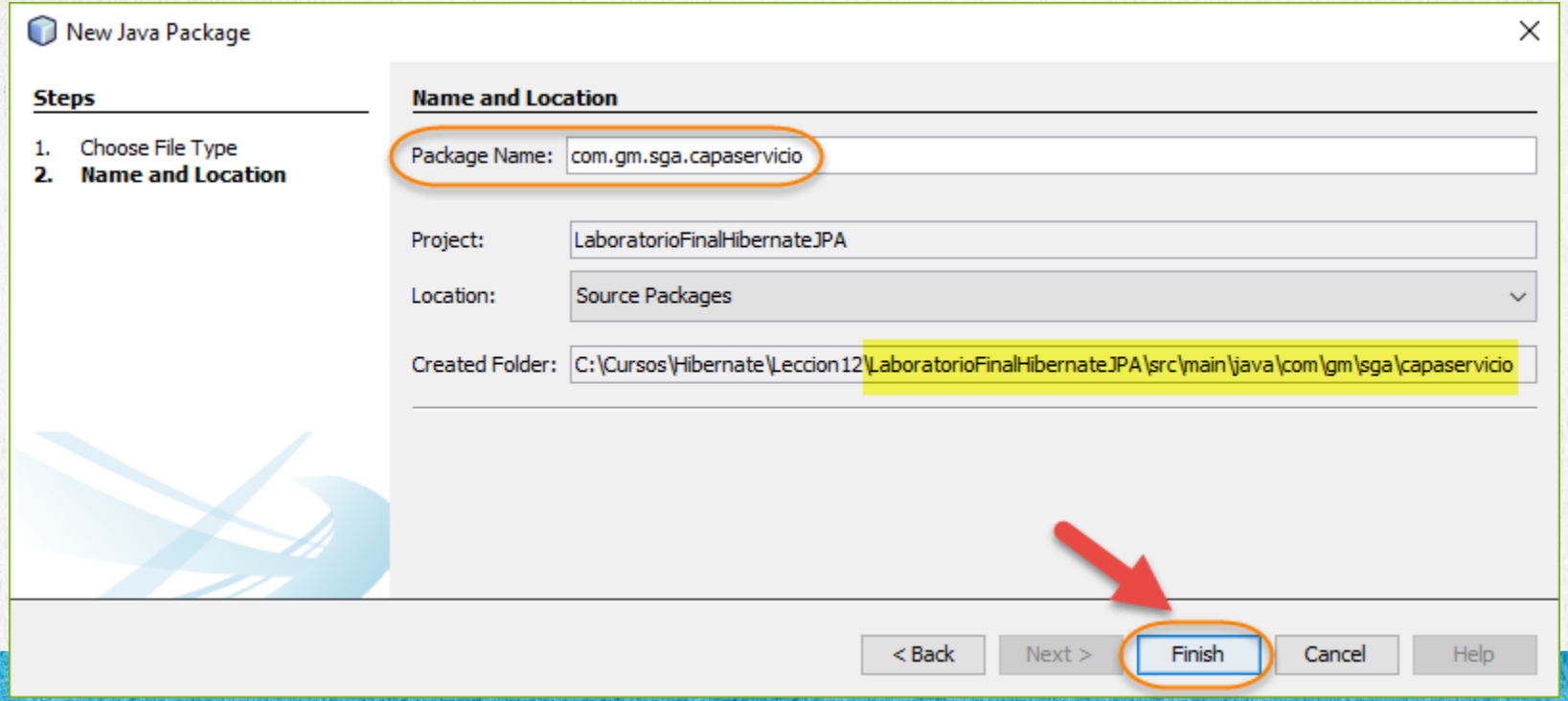
PASO 23. CREAMOS UN PAQUETE JAVA

Creamos el paquete com.gm.sga.capaservicio:



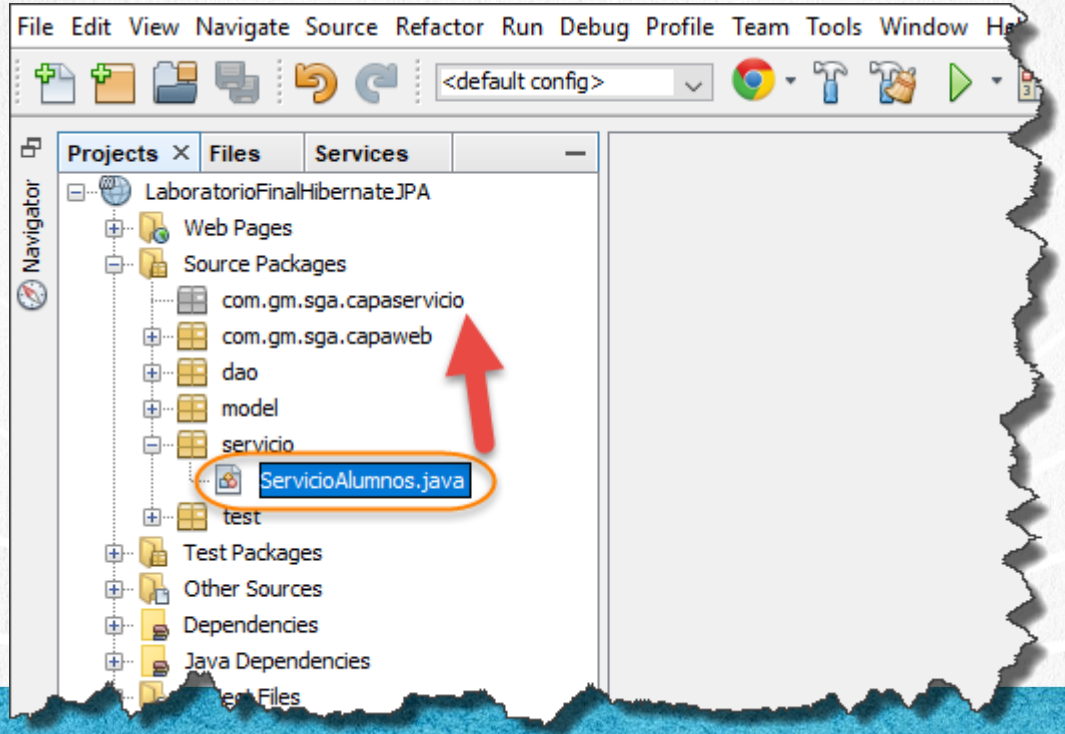
PASO 23. CREAMOS UN PAQUETE JAVA

Creamos el paquete com.gm.sga.capaservicio:



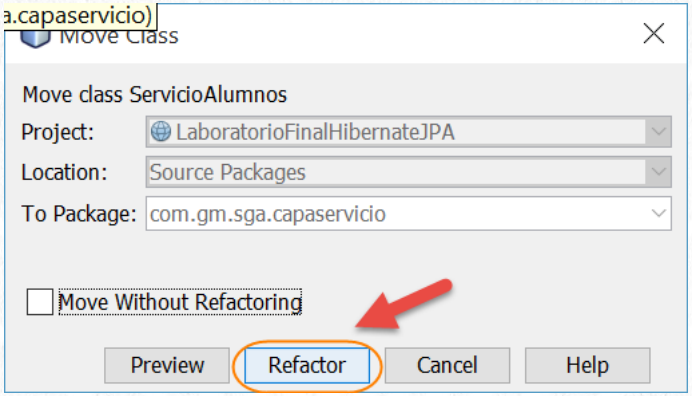
PASO 24. MOVEMOS UNA CLASE JAVA

Movemos la clase del paquete de servicio a com.gm.sga.capaservicio:



PASO 24. MOVEMOS UNA CLASE JAVA

Movemos la clase del paquete de servicio a com.gm.sga.capaservicio:



PASO 25. CREAMOS UNA CLASE JAVA

Creamos la clase ServicioUsuarios.java:

File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help

<default config>

Projects Files Services

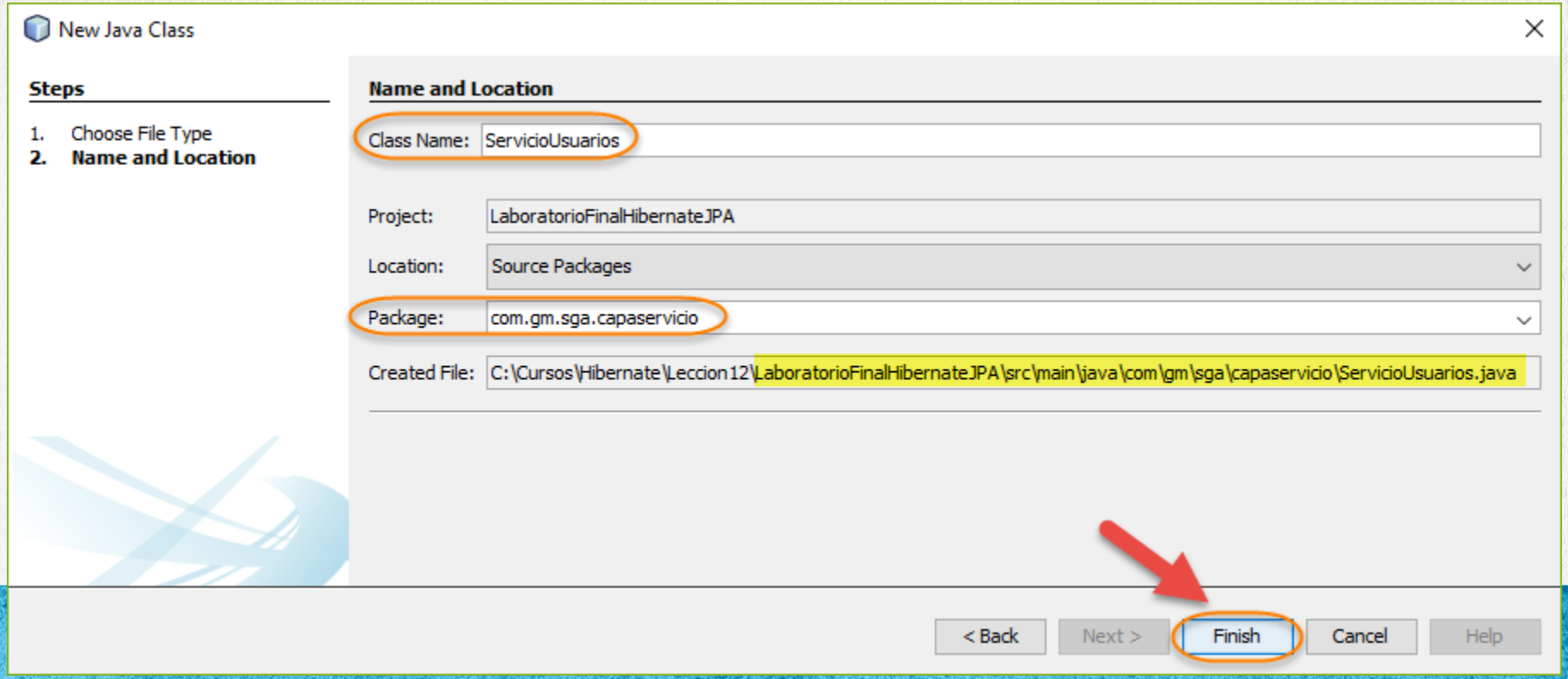
LaboratorioFinalHibernateJPA
- Web Pages
- Source Packages
  - com.gm.sga.capaservicio
  - com.gm.sga.capaweb
  - dao
  - model
  - servicio
  - test
- Test Packages
- Other Sources
- Dependencies
- Java Dependencies
- Project Files

New	
Find...	Ctrl+F
Cut	Ctrl+X
Copy	Ctrl+C
Paste	Ctrl+V
Delete	Delete
Refactor	
Compile Package	F9
Test Package	Ctrl+F6

- Folder...
- Java Package...
- JSP...
- Cascading Style Sheet...
- JavaScript File...
- GlassFish Descript
- Java Class...
- XML Document...
- Properties File...
- Java Interface...

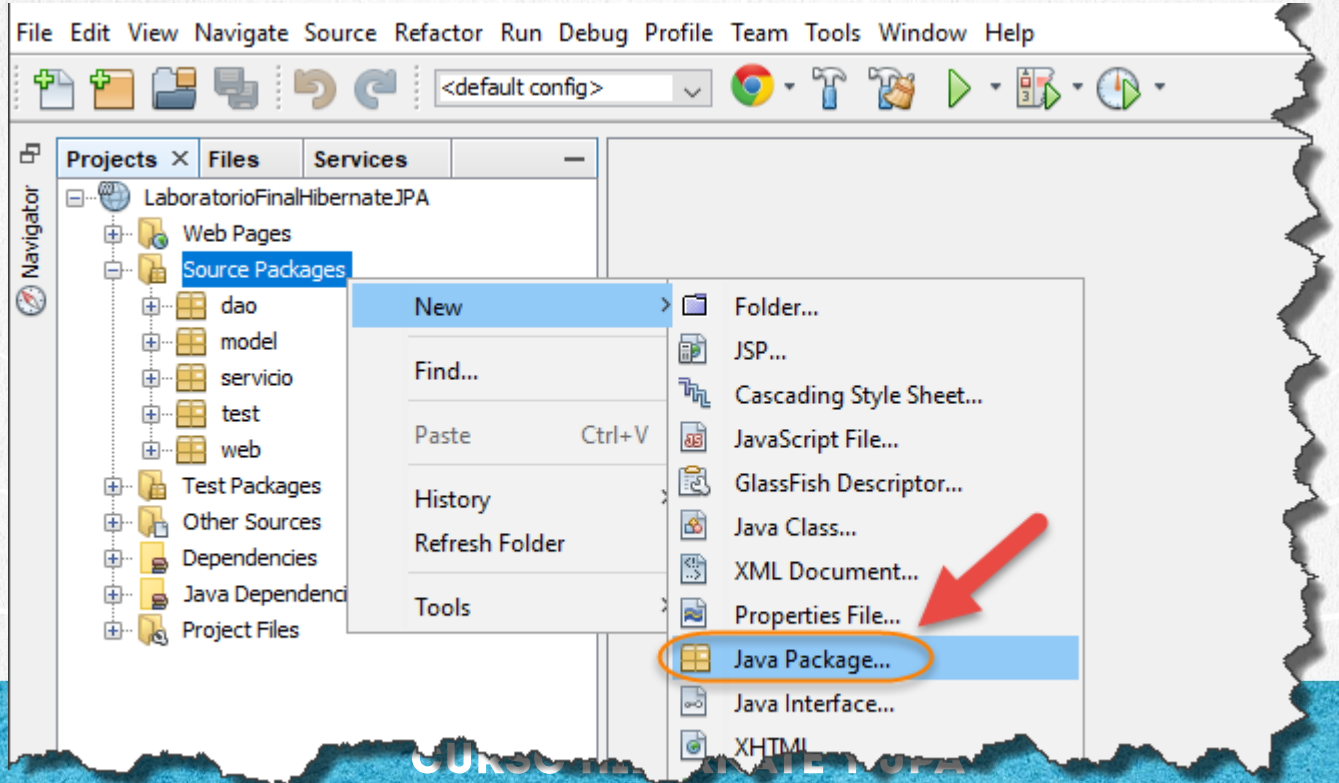
PASO 25. CREAMOS UNA CLASE JAVA

Creamos la clase ServicioUsuarios.java:



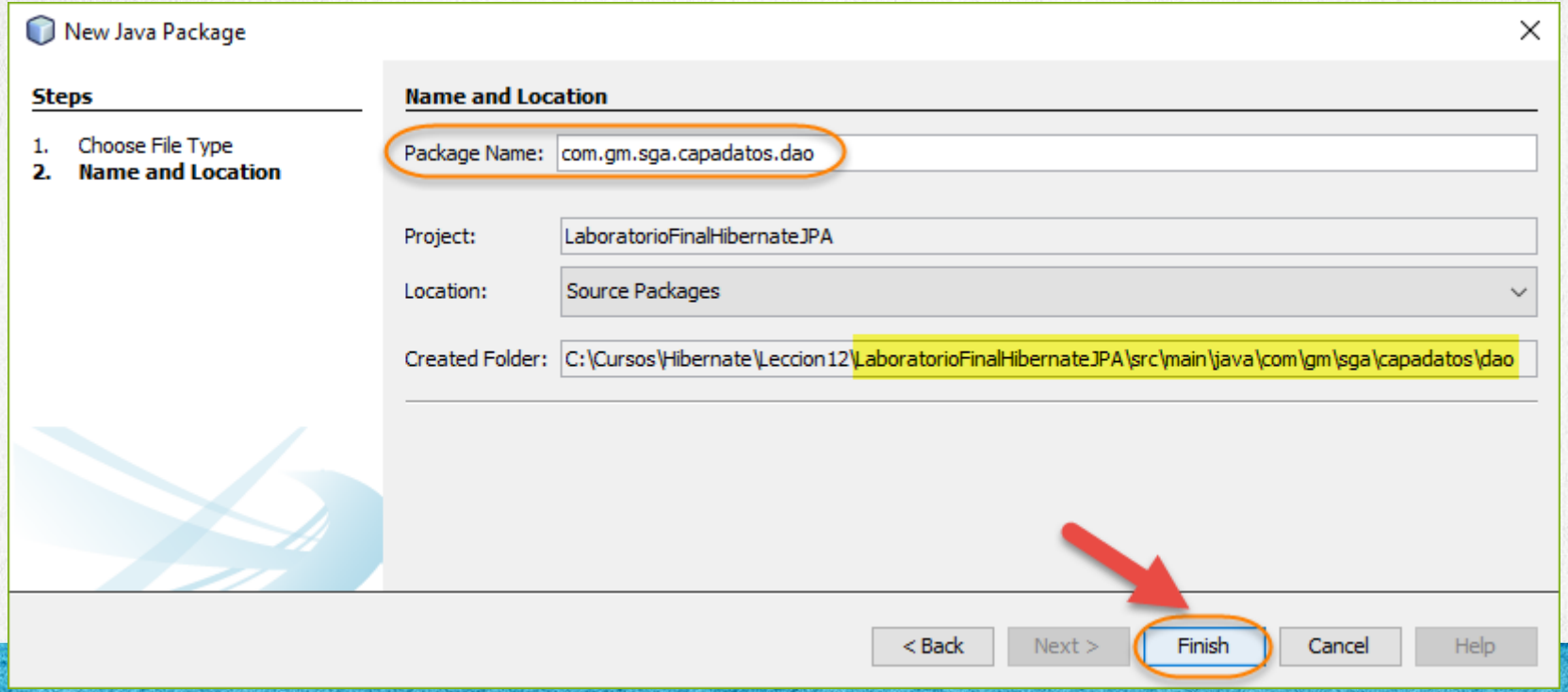
PASO 26. CREAMOS UN PAQUETE DE JAVA

Creamos el paquete com.gm.sga.capadatos.dao:



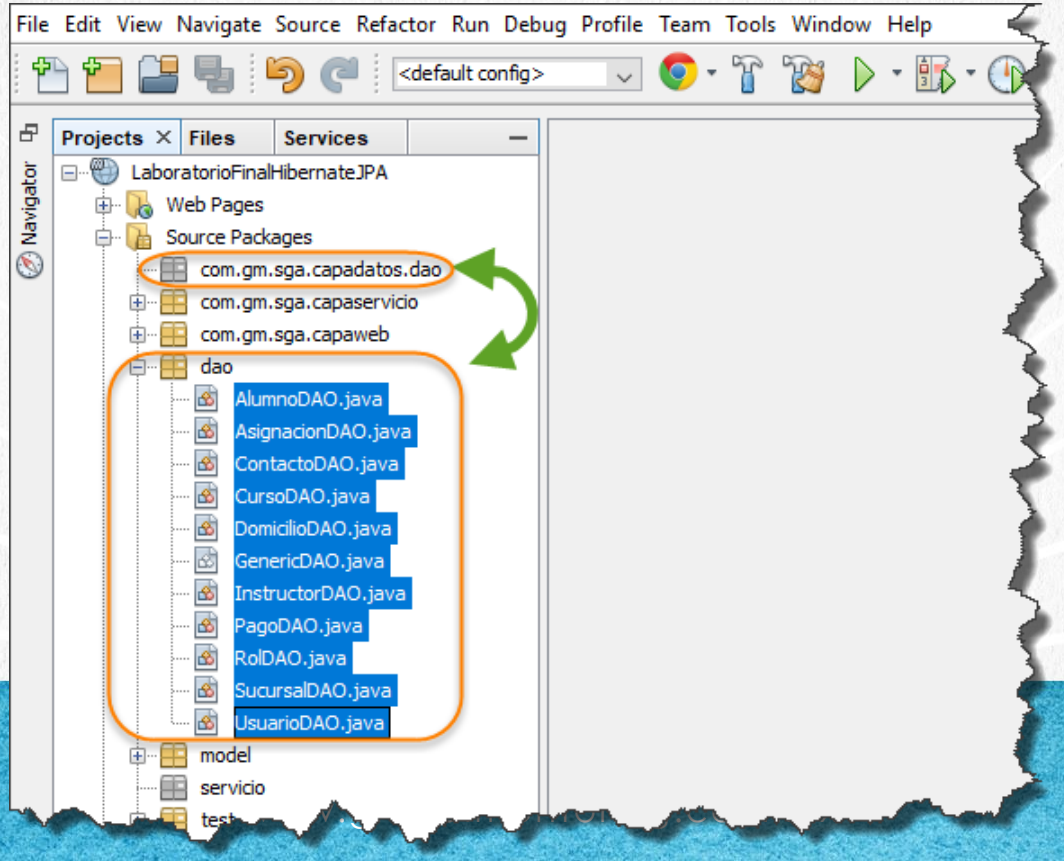
PASO 26. CREAMOS UN PAQUETE DE JAVA

Creamos el paquete com.gm.sga.capadatos.dao:



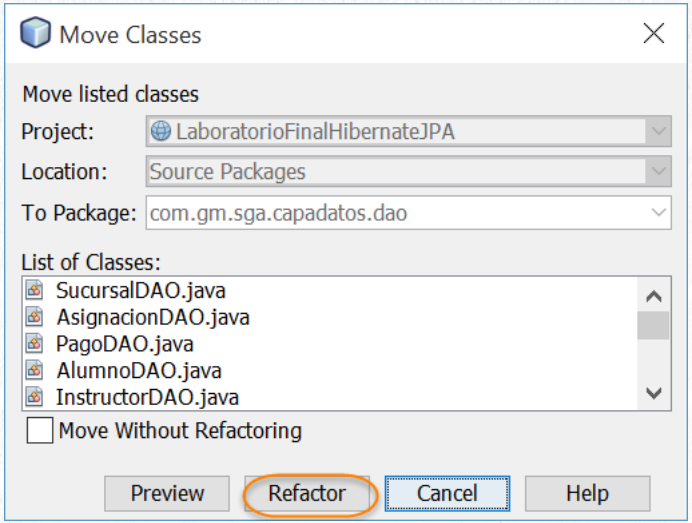
PASO 26. MOVEMOS LAS CLASES JAVA

Movemos las clases Java del paquete dao a com.gm.sga.capadatos.dao:



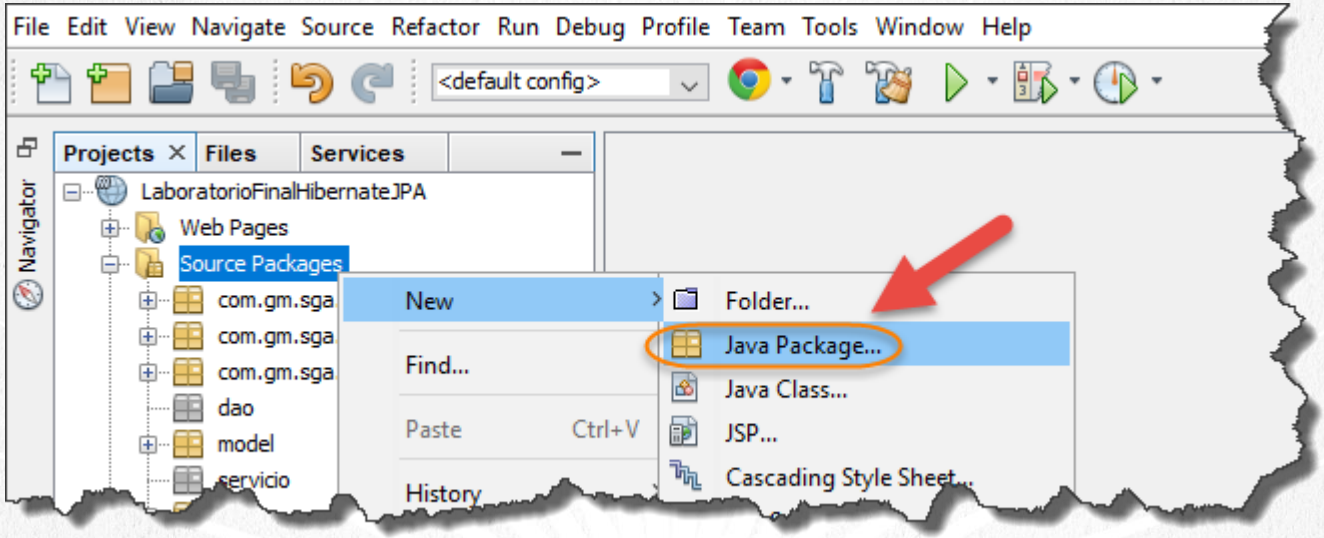
PASO 26. MOVEMOS LAS CLASES JAVA

Movemos las clases Java del paquete dao a com.gm.sga.capadatos.dao:



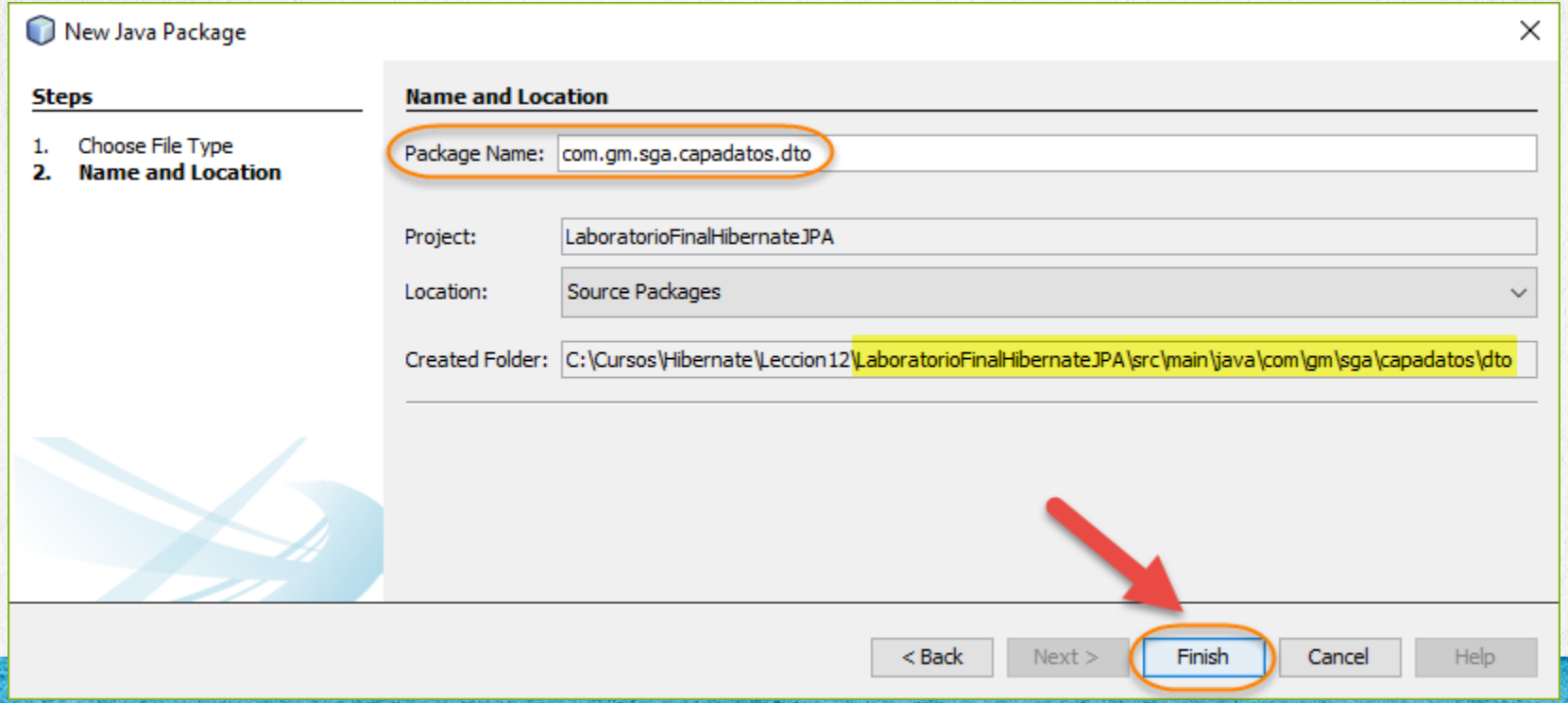
PASO 27. CREAMOS UN PAQUETE DE JAVA

Creamos el paquete com.gm.sga.capadatos.dto:



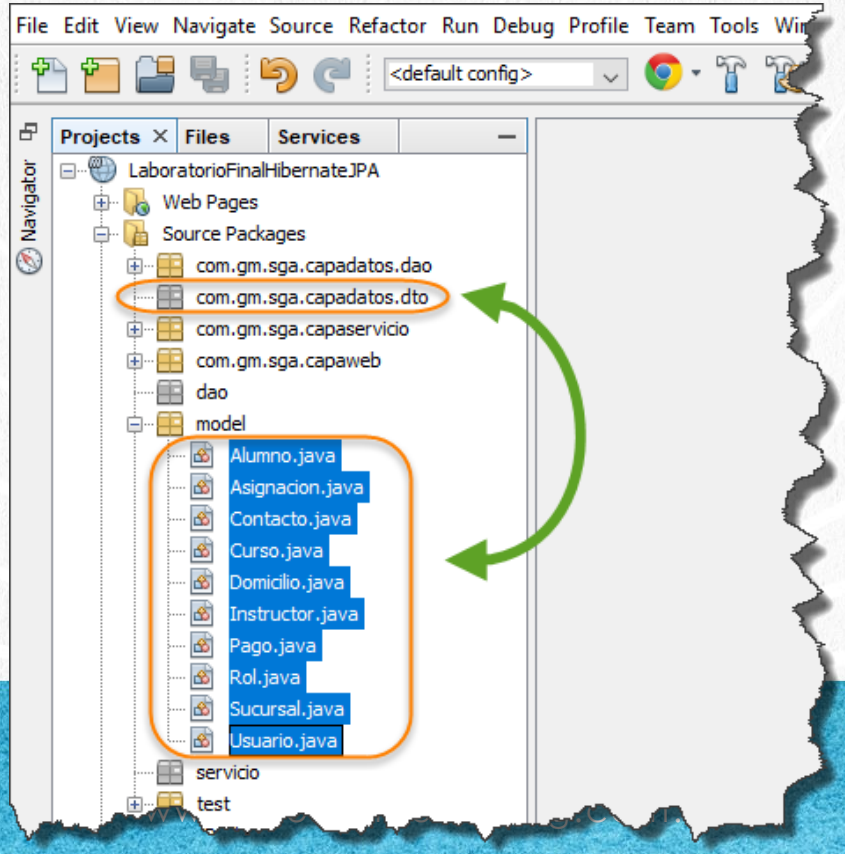
PASO 27. CREAMOS UN PAQUETE DE JAVA

Creamos el paquete com.gm.sga.capadatos.dto:



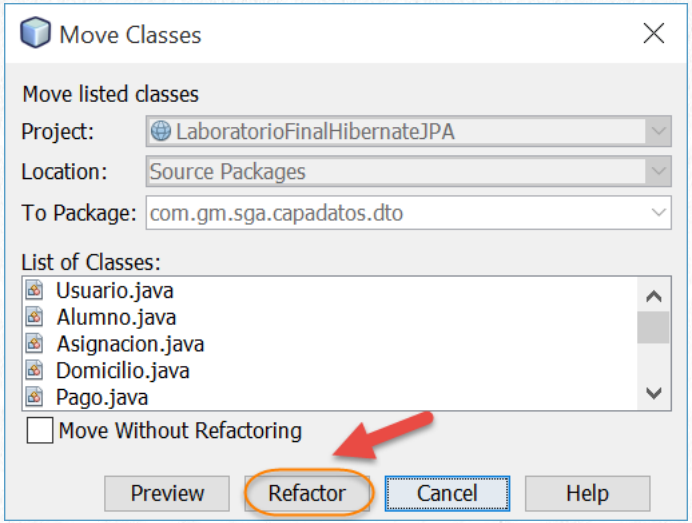
PASO 27. MOVEMOS LAS CLASES JAVA

Movemos las clases Java del paquete model a com.gm.sga.capadatos.dto:



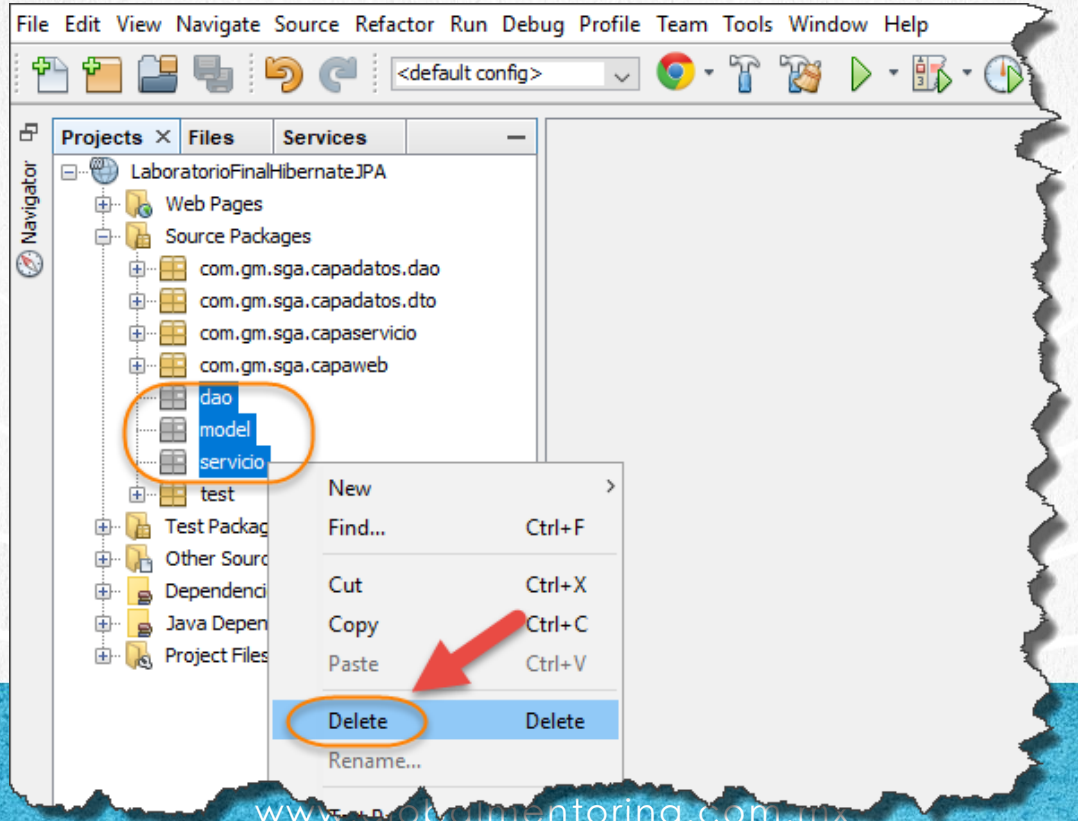
PASO 27. MOVEMOS LAS CLASES JAVA

Movemos las clases Java del paquete model a com.gm.sga.capadatos.dto:



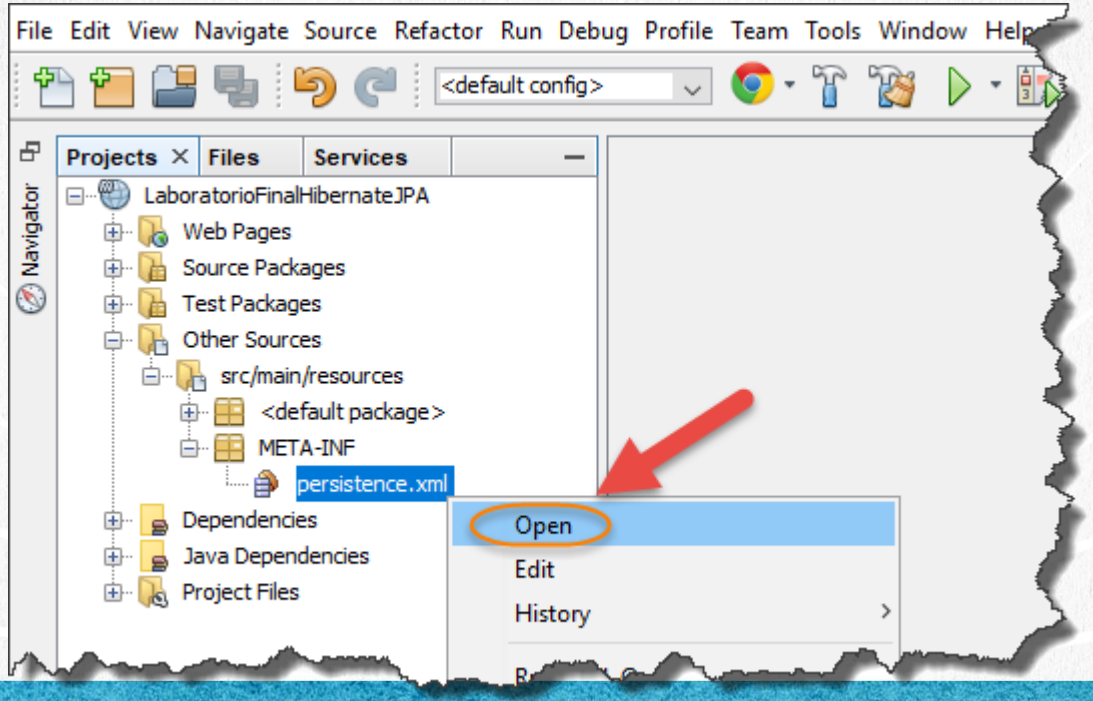
PASO 28. ELIMINAMOS LOS PAQUETES JAVA

Eliminamos los paquetes que ya no usamos, que son: dao, model y servicio:



PASO 29. MODIFICAMOS EL ARCHIVO PERSISTENCE.XML

Modificamos el archivo persistence.xml:



PASO 29. MODIFICAMOS EL CÓDIGO

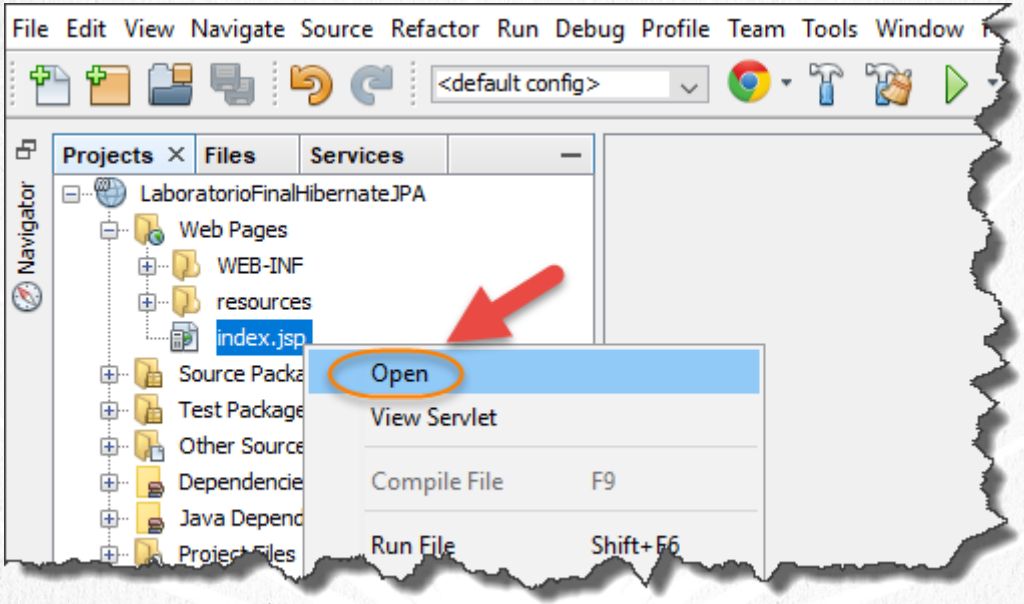
Archivo persistence.xml:

Clic para descargar código

```xml
<?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="HibernateJpaPU" transaction-type="RESOURCE_LOCAL">
    <provider>org.hibernate.jpa.HibernatePersistenceProvider</provider>
    <class>com.gm.sga.dto.Rol</class>
    <class>com.gm.sga.dto.Sucursal</class>
    <class>com.gm.sga.dto.Curso</class>
    <class>com.gm.sga.dto.Alumno</class>
    <class>com.gm.sga.dto.Usuario</class>
    <class>com.gm.sga.dto.Contacto</class>
    <class>com.gm.sga.dto.Pago</class>
    <class>com.gm.sga.dto.Domicilio</class>
    <class>com.gm.sga.dto.Asignacion</class>
    <class>com.gm.sga.dto.Instructor</class>
    <properties>
      <property name="javax.persistence.jdbc.url" value="jdbc:mysql://localhost:3306/sga_db"/>
      <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.format_sql" value="true" />
    </properties>
  </persistence-unit>
</persistence>
```

PASO 30. MODIFICAMOS EL ARCHIVO JSP

Modificamos el archivo index.jsp:



PASO 30. MODIFICAMOS EL CÓDIGO

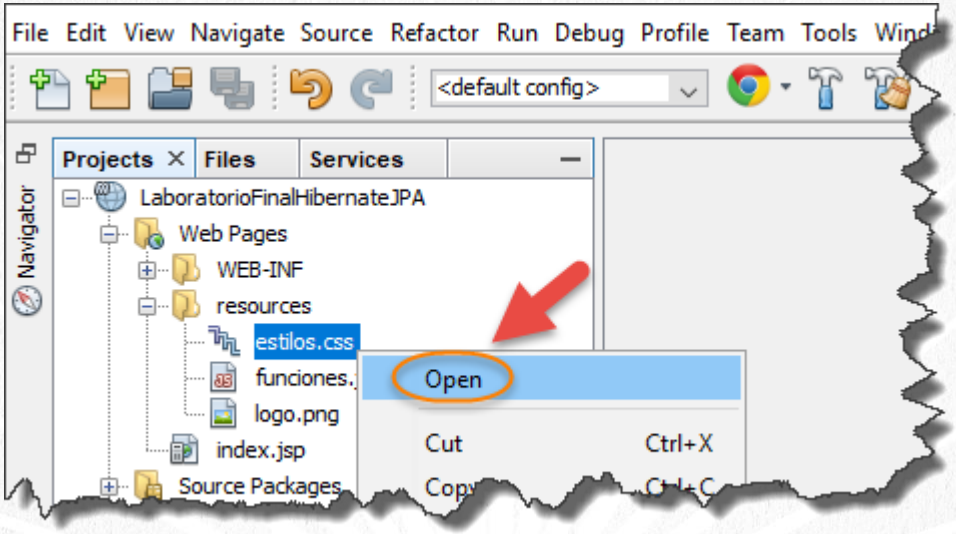
Archivo index.jsp:

Clic para descargar código

```
<!DOCTYPE html>
<html>
  <head>
    <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
    <title>Sistema SGA</title>
  </head>
  <body>
    <div id="wrap">
      <div id="header">
        <jsp:include page="/WEB-INF/pages/commons/header.jsp"/>
      </div>
      <div id="nav" >
        <jsp:include page="/WEB-INF/pages/commons/menu.jsp"/>
      </div>
      <div id="main">
        <div class="subtitulo">
          <a href="${pageContext.request.contextPath}/ServletControlador?accion=listarAlumnos">
            Entrar al Sistema
          </a>
        </div>
      </div>
      <div id="footer">
        <jsp:include page="/WEB-INF/pages/commons/footer.jsp"/>
      </div>
    </div>
  </body>
</html>
```

PASO 30. MODIFICAMOS EL ARCHIVO CSS

Modificamos el archivo estilos.css:



PASO 31. MODIFICAMOS EL CÓDIGO

Archivo estilos.css:

Clic para descargar código

```
html, body {
  margin: 0;
  padding: 0;
}

body {
  font-family:  "Georgia", "Garamond", "Times", "serif";
  font-size:10px;
}

#wrap {
  width:100%;
}

#header {
  color: black;
  background:url('logo.png') no-repeat right;
  margin-right: 15px;
}

//Dar click en el link para descargar el código completo
```

PASO 32. MODIFICAMOS EL CÓDIGO

Archivo funciones.js:

Clic para descargar código

```
var oculto = true;

function verDetalle(){
  if(oculto == true){
    mostrarDetalle();
  }
  else{
    ocultarDetalle();
  }
}

function mostrarDetalle(){
  document.getElementById("detalle").style.display="inline";
  document.getElementById("btnDetalle").value = "Ocultar detalle";
  oculto = false;
}

function ocultarDetalle(){
  document.getElementById("detalle").style.display="none";
  document.getElementById("btnDetalle").value = "Ver detalle";
  oculto = true;
}

//Dar click en el link para descargar el código completo
```

PASO 33. MODIFICAMOS EL CÓDIGO

Archivo login.jsp:

Clic para descargar código

```
<!DOCTYPE html>
<html>
  <head>
    <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
    <title>Login Sistema Web AP</title>
  </head>
  <body onload="sf();">
    <div id="wrap">
      <div id="header">
        <jsp:include page="/WEB-INF/pages/commons/header.jsp"/>
      </div>

      <div id="nav">
        <jsp:include page="/WEB-INF/pages/commons/menu.jsp"/>
      </div>

      <div id="msg">
        <jsp:include page="/WEB-INF/pages/commons/messages.jsp"/>
      </div>
      <div id="main">
        <div class="subtitulo">
          Login Sistema SGA
        </div>

//Dar click en el link para descargar el código completo
```

PASO 34. MODIFICAMOS EL CÓDIGO

Archivo listarAlumnos.jsp:

Clic para descargar código

```
<%@taglib uri="http://java.sun.com/jsp/jstl/core" prefix="c"%>
<!DOCTYPE html>
<html>
    <head>
        <title>Listado de Alumnos</title>
    </head>
    <body>
        <div id="wrap">
            <div id="header">
                <jsp:include page="/WEB-INF/pages/commons/header.jsp" />
            </div>

            <div id="nav">
                <jsp:include page="/WEB-INF/pages/commons/menu.jsp" />
            </div>

            <div id="msg">
                <jsp:include page="/WEB-INF/pages/commons/messages.jsp" />
            </div>

            <div id="main">

//Dar click en el link para descargar el código completo
```

PASO 35. MODIFICAMOS EL CÓDIGO

Archivo detalleAlumno.jsp:

Clic para descargar código

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

<html>
  <head>
    <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
    <title>Detalle del Alumno</title>
  </head>
  <body>
    <div id="wrap">
      <div id="header">
        <jsp:include page="/WEB-INF/pages/commons/header.jsp"/>
      </div>

      <div id="nav">
        <jsp:include page="/WEB-INF/pages/commons/menu.jsp"/>
      </div>

      <div id="msg">
        <jsp:include page="/WEB-INF/pages/commons/messages.jsp"/>
      </div>

//Dar click en el link para descargar el código completo
```

PASO 36. MODIFICAMOS EL CÓDIGO

Archivo buscarAlumnos.jsp:

Clic para descargar código

```
<%@taglib uri="http://java.sun.com/jsp/jstl/core" prefix="c"%>
<!DOCTYPE html>
<html>
    <head>
        <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
        <title>Busqueda Alumnos</title>
    </head>
    <body>
        <div id="wrap">
            <div id="header">
                <jsp:include page="/WEB-INF/pages/commons/header.jsp" />
            </div>

            <div id="nav">
                <jsp:include page="/WEB-INF/pages/commons/menu.jsp" />
            </div>

            <div id="msg">
                <jsp:include page="/WEB-INF/pages/commons/messages.jsp" />
            </div>
//Dar click en el link para descargar el código completo
```

PASO 37. MODIFICAMOS EL CÓDIGO

Archivo error.jsp:

Clic para descargar código

```
<%@page isErrorPage="true" %>
<%
    String param = (String) request.getAttribute("mensaje");
    String mensaje = "Sin mensaje";
    if (param != null && param.length() > 0) {
        mensaje = param;
    }
%>
<!DOCTYPE html>
<html>
    <head>
        <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
        <title>Login Sistema Web AP</title>

    </head>
    <body  onload="ocultarDetalle();">
        <div id="wrap">
            <div id="header">
                <jsp:include page="/WEB-INF/pages/commons/header.jsp"/>
            </div>

            <div id="nav">
                <jsp:include page="/WEB-INF/pages/commons/menu.jsp"/>
            </div>
//Dar click en el link para descargar el código completo
```

PASO 38. MODIFICAMOS EL CÓDIGO

Archivo footer.jsp:

Clic para descargar código

```
<a href="http://www.globalmentoring.com.mx" target="_blank">
  GlobalMentoring.com.mx
</a>
    -  
<a href="http://www.universidadjava.com" target="_blank">
  UniversidadJava.com
</a>
    -  
  Todos los Derechos Reservados
```

PASO 39. MODIFICAMOS EL CÓDIGO

Archivo header.jsp:

Clic para descargar código

```
<%-- Prevencion del cache --%>
<%
    response.setHeader("Cache-Control", "no-cache"); //HTTP 1.1
    response.setHeader("Pragma", "no-cache"); //HTTP 1.0
    response.setDateHeader("Expires", 0); //prevents caching at the proxy server
    response.setHeader("Cache-Control", "no-store");
%>

<link rel="stylesheet" type="text/css" href="${pageContext.request.contextPath}/resources/estilos.css"/>

<script language="javascript" src="${pageContext.request.contextPath}/resources/funciones.js"></script>
<!--Incluimos el logo de global mentoring -->

<div class="titulo">
  Sistema de Gesti&oacute;n
  <br> de Alumnos (SGA)
</div>
```

PASO 40. MODIFICAMOS EL CÓDIGO

Archivo menu.jsp:

Clic para descargar código

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

<div class="der">
  ${usuario!=null ? "Bienvenido " : ""}
</div>

<div class="izq">
  <a href="${pageContext.request.contextPath}/index.jsp">
    Inicio
  </a>
   

  <c:if test="${usuario!=null}" >
     
    <a href="${pageContext.request.contextPath}/ServletControlador?accion=salir">
      Salir
    </a>
  </c:if>
</div>
```

PASO 41. MODIFICAMOS EL CÓDIGO

Archivo messages.jsp:

Clic para descargar código

```
<div class="error"> ${mensaje} </div>
```

PASO 42. MODIFICAMOS EL CÓDIGO

Archivo Alumno.java:

Clic para descargar código

```
@Entity
@Table(name = "alumno")
@NamedQueries({
    @NamedQuery(name = "Alumno.findAll", query = "SELECT a FROM Alumno a")})
public class Alumno implements Serializable {

    private static final long serialVersionUID = 1L;
    @Id
    @GeneratedValue(strategy = GenerationType.IDENTITY)
    @Basic(optional = false)
    @Column(name = "id_alumno")
    private Integer idAlumno;

    @Basic(optional = false)
    @Column(name = "nombre")
    private String nombre;

    @Basic(optional = false)
    @Column(name = "apellido_paterno")
    private String apellidoPaterno;

    @Column(name = "apellido_materno")
    private String apellidoMaterno;

//Dar click en el link para descargar el código completo
```

PASO 43. MODIFICAMOS EL CÓDIGO

Archivo AlumnoDAO.java:

Clic para descargar código

```
public class AlumnoDAO extends GenericDAO {

    Logger log = LogManager.getRootLogger();

    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();
    }

//Dar click en el link para descargar el código completo
```

PASO 44. MODIFICAMOS EL CÓDIGO

Archivo UsuarioDAO.java:

Clic para descargar código

```
public class UsuarioDAO extends GenericDAO {

    Logger log = LogManager.getRootLogger();

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


//Dar click en el link para descargar el código completo
```

PASO 45. MODIFICAMOS EL CÓDIGO

Archivo ServicioAlumnos.java:

Clic para descargar código

```
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();
    }

    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
    }

//Dar click en el link para descargar el código completo
```

PASO 46. MODIFICAMOS EL CÓDIGO

Archivo ServicioUsuarios.java:

Clic para descargar código

```
public class ServicioUsuarios {
    // El Dao se asocia con la clase

    UsuarioDAO usuarioDao = new UsuarioDAO();

    // Esta clase se comunica con los DAO's que necesite
    public List<Usuario> listarUsuarios() {
        return usuarioDao.listarUsuarios();
    }

    public boolean guardarUsuario(Usuario usuario) {
        if (usuario != null && usuario.getIdUsuario() == null) {
            //insert
            usuarioDao.insertar(usuario);
        } else {
            //update
            usuarioDao.actualizar(usuario);
        }

        return true;// si nada fallo, regresamos verdadero
    }

//Dar click en el link para descargar el código completo
```

PASO 47. MODIFICAMOS EL CÓDIGO

Archivo ServletBuscar.java:

Clic para descargar código

```
@WebServlet(name = "ServletBuscar", urlPatterns = {"/ServletBuscar"})
public class ServletBuscar extends HttpServlet {
    @Override
    public void doPost(HttpServletRequest request, HttpServletResponse response)
            throws ServletException, IOException {

        //1. Recuperamos los parámetros de búsqueda
        //Recuperamos los datos del alumno
        String nombreAlumno = request.getParameter("nombreAlumno");
        nombreAlumno = "".equals(nombreAlumno) ? null : nombreAlumno.trim();

        String apellidoPaterno = request.getParameter("apellidoPaterno");
        apellidoPaterno = "".equals(apellidoPaterno) ? null : apellidoPaterno.trim();

        String apellidoMaterno = request.getParameter("apellidoMaterno");
        apellidoMaterno = "".equals(apellidoMaterno) ? null : apellidoMaterno.trim();

        Alumno alumnoDTO = null;
        if (nombreAlumno != null || apellidoPaterno != null || apellidoMaterno != null) {
            alumnoDTO = new Alumno();
            alumnoDTO.setNombre(nombreAlumno);
            alumnoDTO.setApellidoPaterno(apellidoPaterno);
            alumnoDTO.setApellidoMaterno(apellidoMaterno);
        }
//Dar click en el link para descargar el código completo
```

PASO 48. MODIFICAMOS EL CÓDIGO

Archivo ServletControlador.java:

Clic para descargar código

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

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

        //Revisamos los casos de uso del sistema SGA
        String accion = request.getParameter("accion");

        if ("listarAlumnos".equals(accion)) {
            this.confirmarUsuarioEnSession(request, response);
        } else if ("validarUsuario".equals(accion)) {
            this.validarUsuario(request, response);
        } else if ("agregarAlumno".equals(accion)) {
            this.agregarAlumno(request, response);
        } else if ("editarAlumno".equals(accion)) {
            this.editarAlumno(request, response);
        } else if ("eliminarAlumno".equals(accion)) {
            this.eliminarAlumno(request, response);
        }

     //Dar click en el link para descargar el código completo
```

PASO 49. MODIFICAMOS EL CÓDIGO

Archivo ServletModificar.java:

Clic para descargar código

```
@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");

        //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);

     //Dar click en el link para descargar el código completo
```

PASO 50. MODIFICAMOS EL CÓDIGO

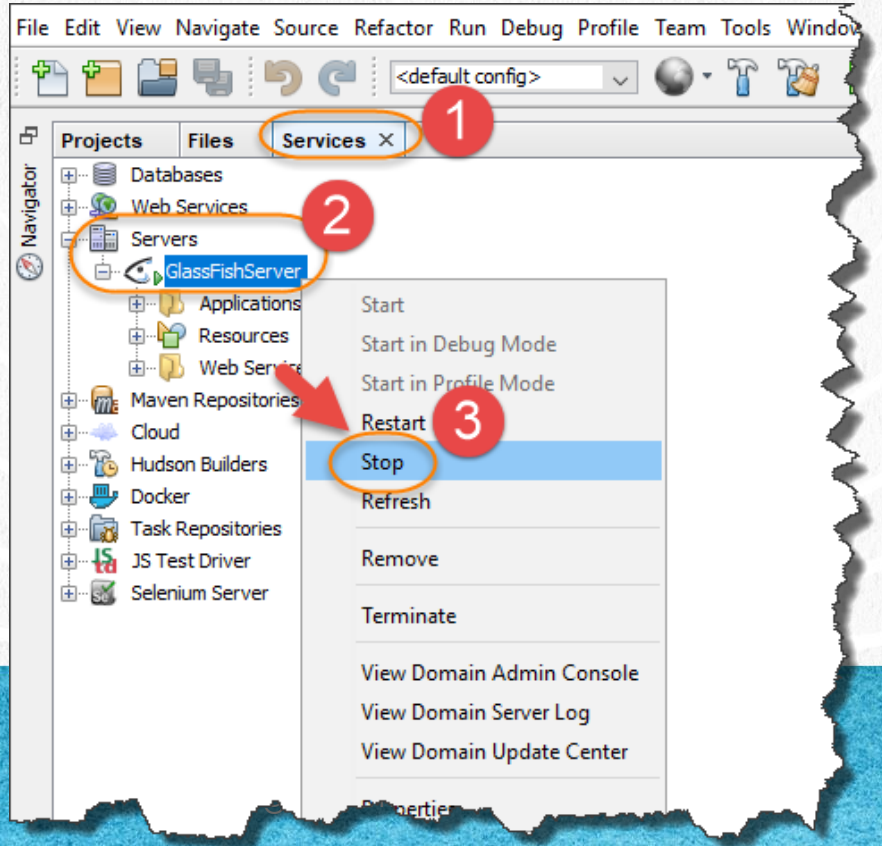
Archivo ServletRedireccionar.java:

Clic para descargar código

```
package com.gm.sga.capaweb;
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 {

        String accion = request.getParameter("accion");
        if ("agregar".equals(accion)) {
            //Redireccionamos a la pagina de Agregar Alumno
            request.getRequestDispatcher("WEB-INF/agregarAlumno.jsp").forward(request, response);
        } else if ("buscar".equals(accion)) {
            //Redireccionamos a la pagina de Busqueda avanzada para usar criteria
            request.getRequestDispatcher("WEB-INF/busquedaAvanzada.jsp").forward(request, response);
        }
    }
}
```

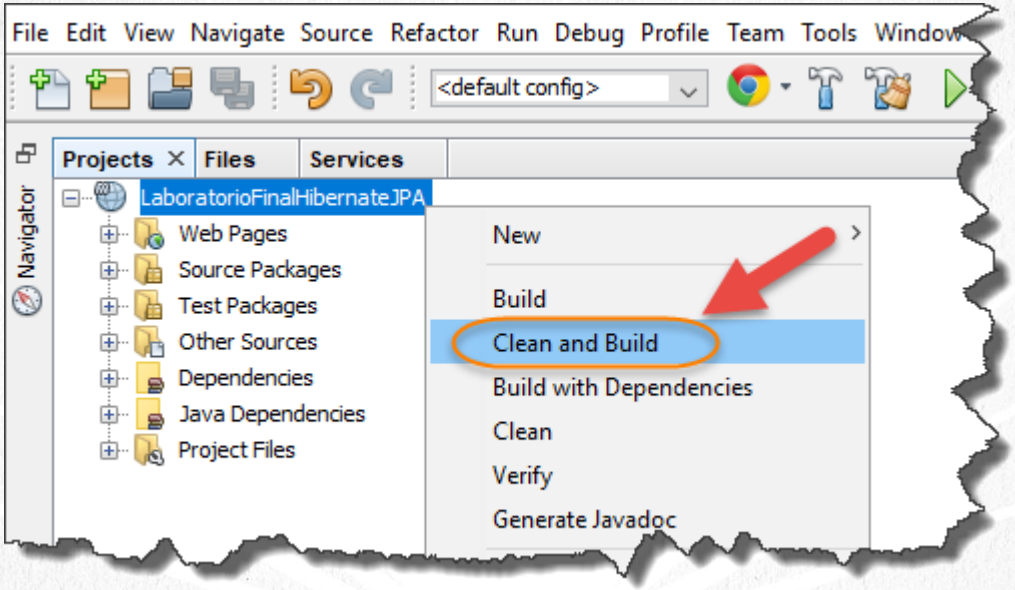
PASO 51. DETENEMOS EL SERVIDOR (OPCIONAL)

Detenemos el servidor de Glassfish sólo si se está ejecutando:



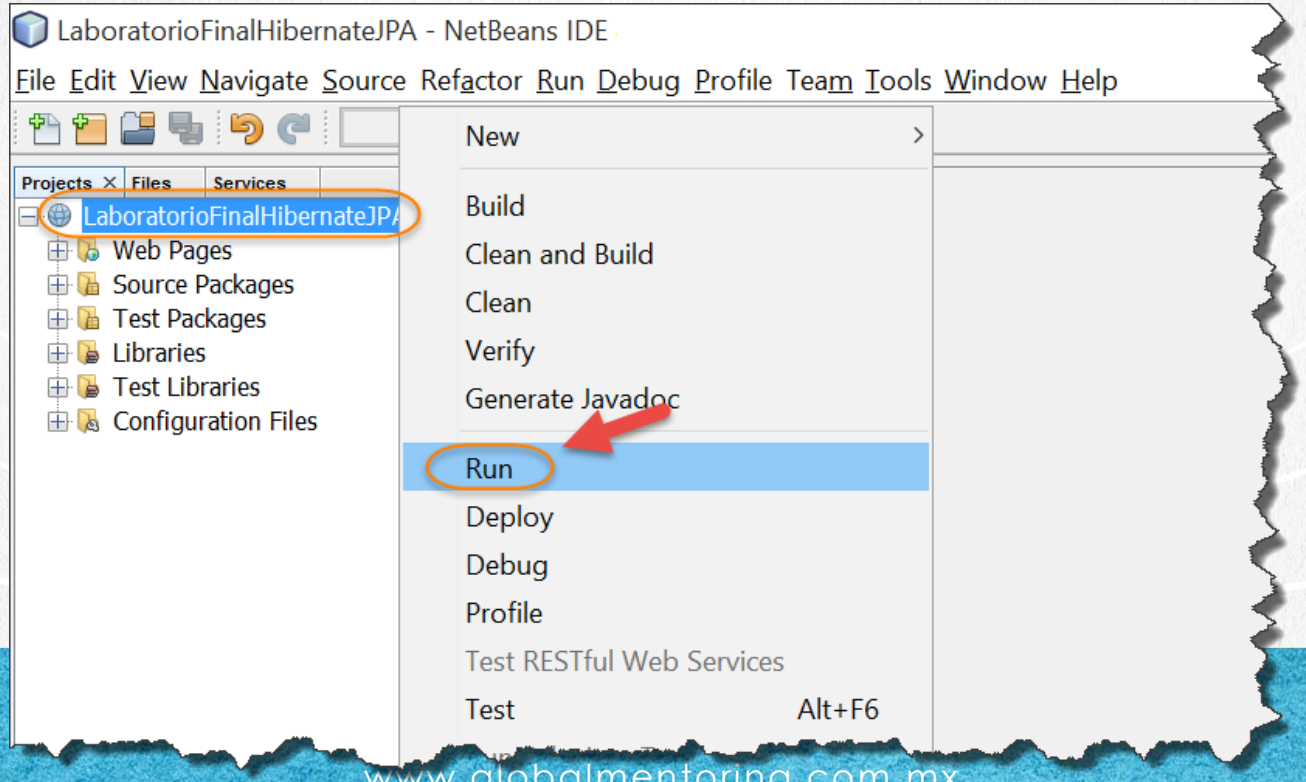
PASO 52. HACEMOS CLEAN & BUILD AL PROYECTO

Hacemos clean & build para obtener la última versión de cada archivo:



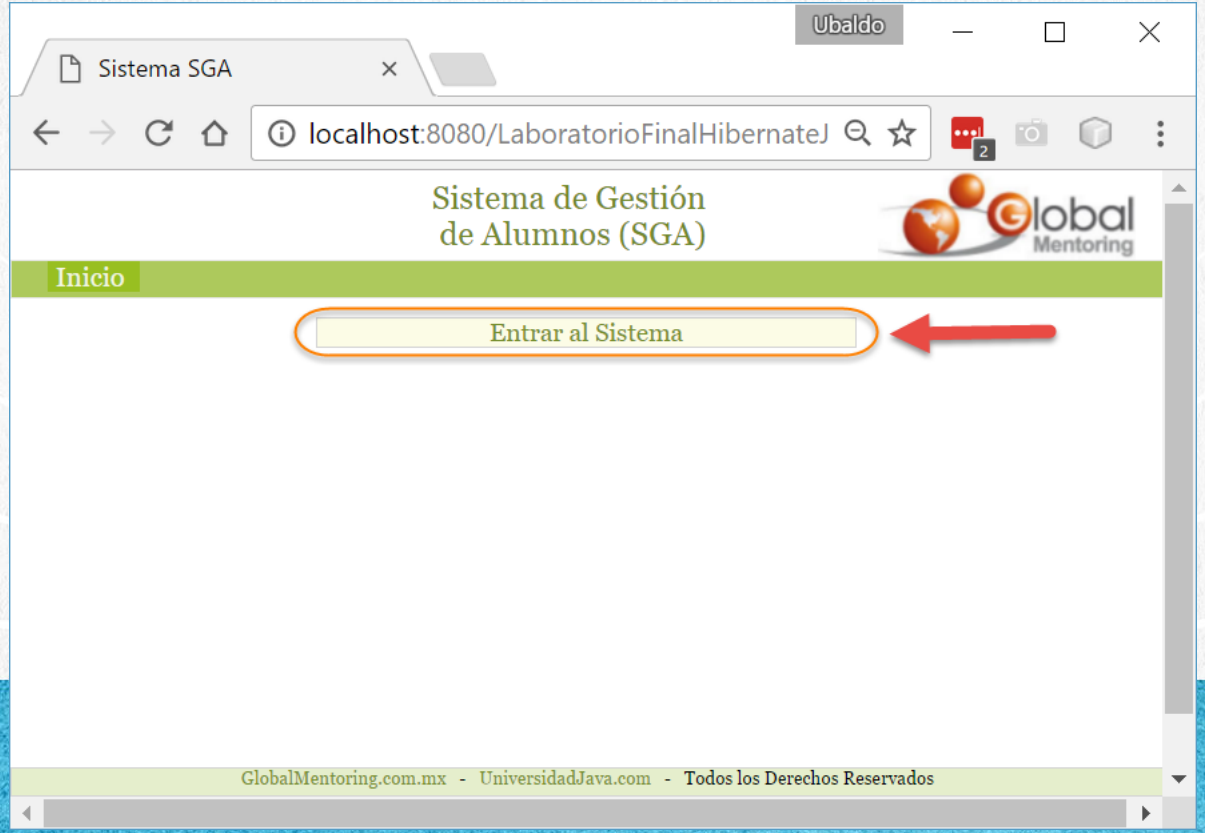
PASO 53. EJECUTAMOS EL PROYECTO

Ejecutamos el proyecto:



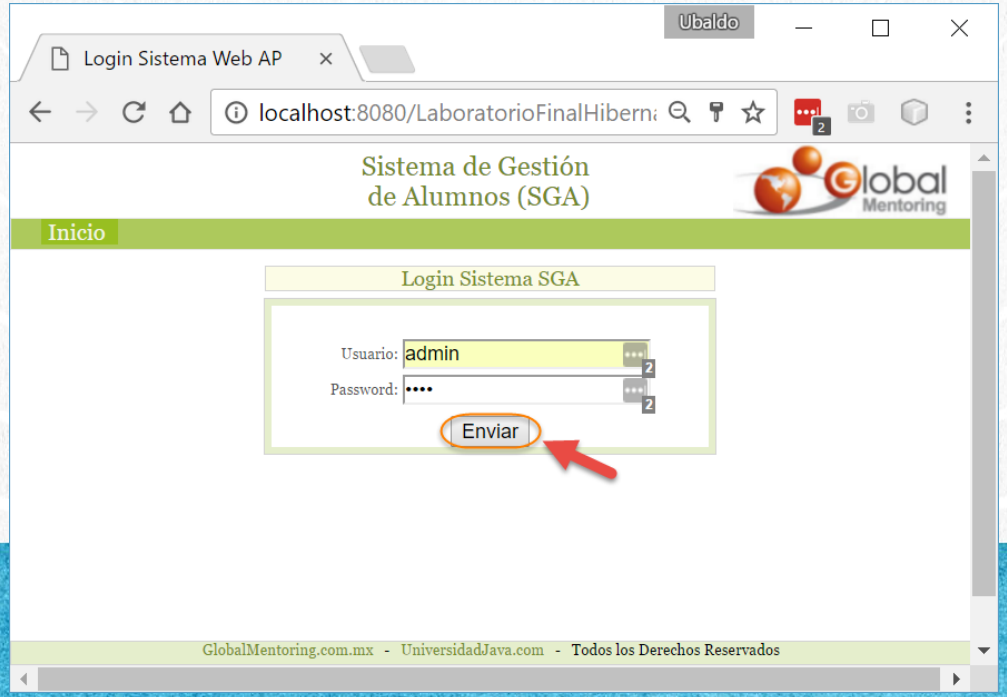
PASO 53. EJECUTAMOS EL PROYECTO

Ejecutamos el proyecto:



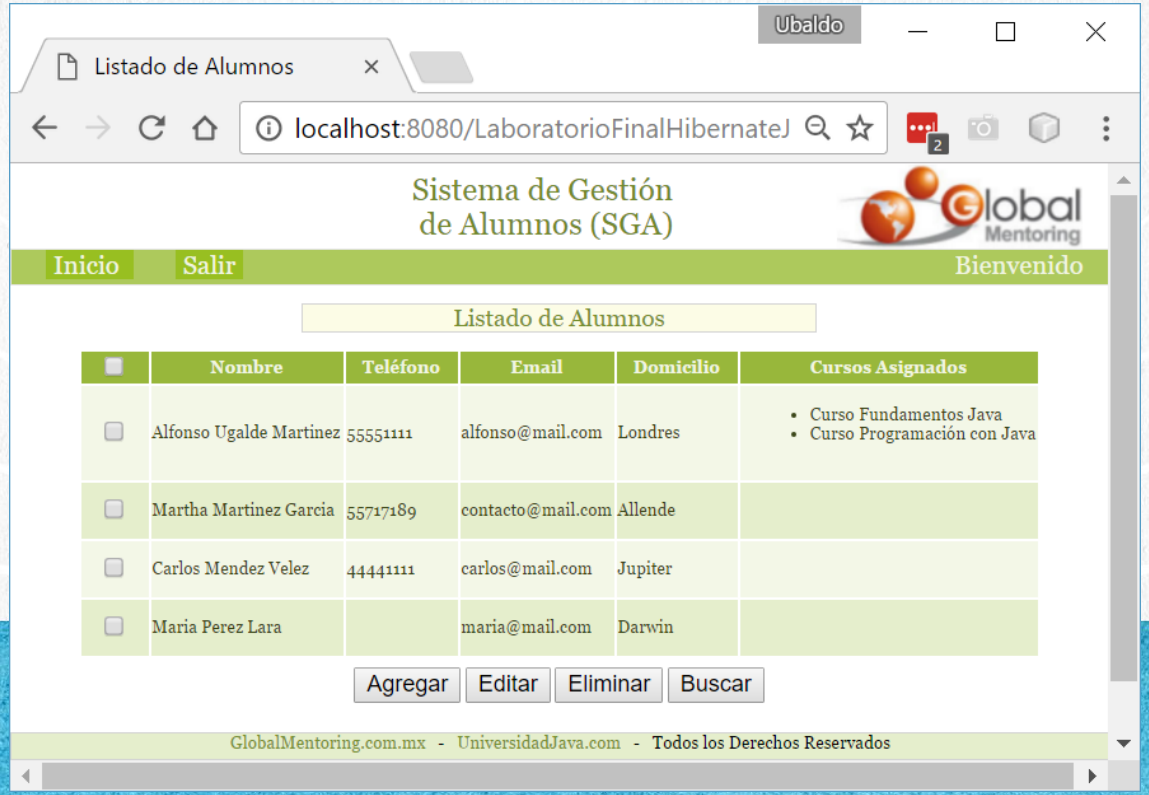
PASO 53. EJECUTAMOS EL PROYECTO

Ejecutamos el proyecto. Los valores de usuario y password son:
Usuario: admin y Password: 123
Estos valores se pueden agregar a la tabla de usuario en la base de datos.



PASO 53. EJECUTAMOS EL PROYECTO

Probamos cada una de las acciones de la aplicación. Listar, agregar, editar, eliminar y buscar:



EN CASO DE PROBLEMAS

En caso de tener problemas, recuerden que primero deben detener Glassfish si es que está arriba.

Posteriormente deben hacer un Clean & Build para tener la última versión de cada archivo.

Y finalmente, vuelven a desplegar la aplicación ejecutando la opción de run.

CONCLUSIÓN DEL EJERCICIO

Con este laboratorio final concluimos el curso de Hibernate y JPA.

Hemos puesto en práctica todos los conceptos vistos en el curso, además lo integramos a una aplicación Web del mundo real, la cual puede ser usada como base para tus propios proyectos.

Te esperamos pronto en más cursos de Global Mentoring y la Universidad Java. ¡Hasta Pronto!

CURSO ONLINE

HIBERNATE & JPA

Por: Ing. Ubaldo Acosta

CURSO HIBERNATE Y JPA

www.globalmentoring.com.mx