

# CURSO STRUTS FRAMEWORK

# INTEGRACIÓN DE FRAMEWORKS

# STRUTS + SPRING + HIBERNATE

# CRUD



Por el experto: Ing. Ubaldo Acosta



Experiencia y Conocimiento para tu vida




**CURSO STRUTS FRAMEWORK**

[www.globalmentoring.com.mx](http://www.globalmentoring.com.mx)

# OBJETIVO DEL EJERCICIO

Crear una aplicación para poner en práctica la integración de los frameworks Struts + Spring + Hibernate aplicando las operaciones CRUD (Cread-Read-Update-Delete) sobre la tabla de persona. Al finalizar deberemos observar lo siguiente:



**Personas con Struts 2**

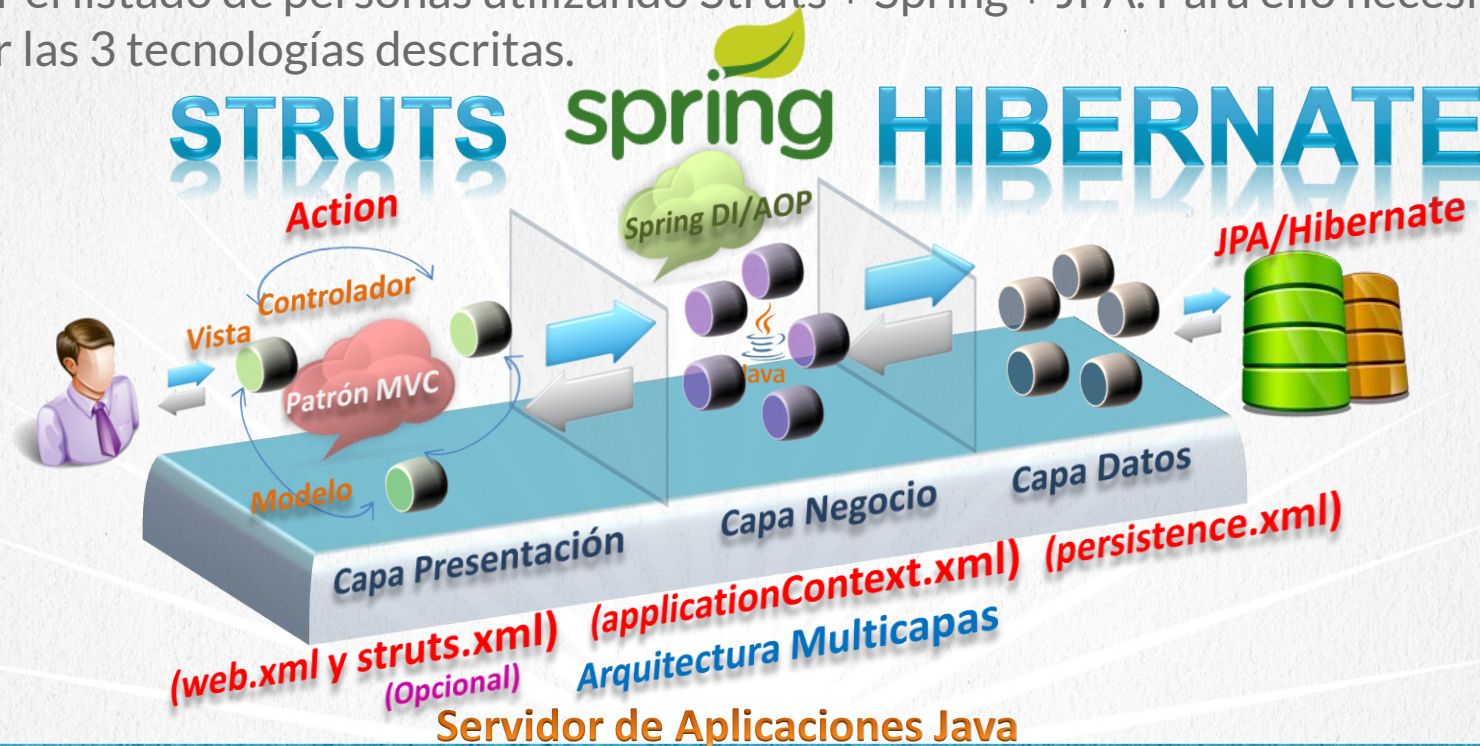
[Agregar Persona](#)

idPersona	Nombre	Apellido Paterno	Apellido Materno	Email	Editar	Eliminar
1	Administrador	Esparza	Sistemas	mesparza@mail.com	<a href="#">Editar</a>	<a href="#">Eliminar</a>
2	Carlos	Romero	Esparza	carlos.romero@gmail.com	<a href="#">Editar</a>	<a href="#">Eliminar</a>
3	Karla	Lopez	Perez	klopez@mail.com	<a href="#">Editar</a>	<a href="#">Eliminar</a>
7	Maria	Toledo	Ramírez	mtoledo@mail.com	<a href="#">Editar</a>	<a href="#">Eliminar</a>



# REQUERIMIENTO DEL EJERCICIO

Mostrar el listado de personas utilizando Struts + Spring + JPA. Para ello necesitamos integrar las 3 tecnologías descritas.



**CURSO STRUTS FRAMEWORK**

[www.globalmentoring.com.mx](http://www.globalmentoring.com.mx)

# 1. CREAR PROYECTO

Vamos a utilizar Maven para crear el proyecto Java Web. El proyecto se llamará StrutsSpringHibernateCRUD. Este proyecto integrará las 3 tecnologías: Struts + Spring + Hibernate.

La tecnología que usaremos para la capa de datos será JPA pero utilizando la implementación de Hibernate Framework. Debemos recordar que JPA es el estándar y Hibernate aunque se puede utilizar de manera aislada, es más recomendable utilizarlo como una implementación de JPA para mantenernos dentro del estándar de Java EE.

Empecemos con nuestro ejercicio:



Experiencia y Conocimiento para tu vida

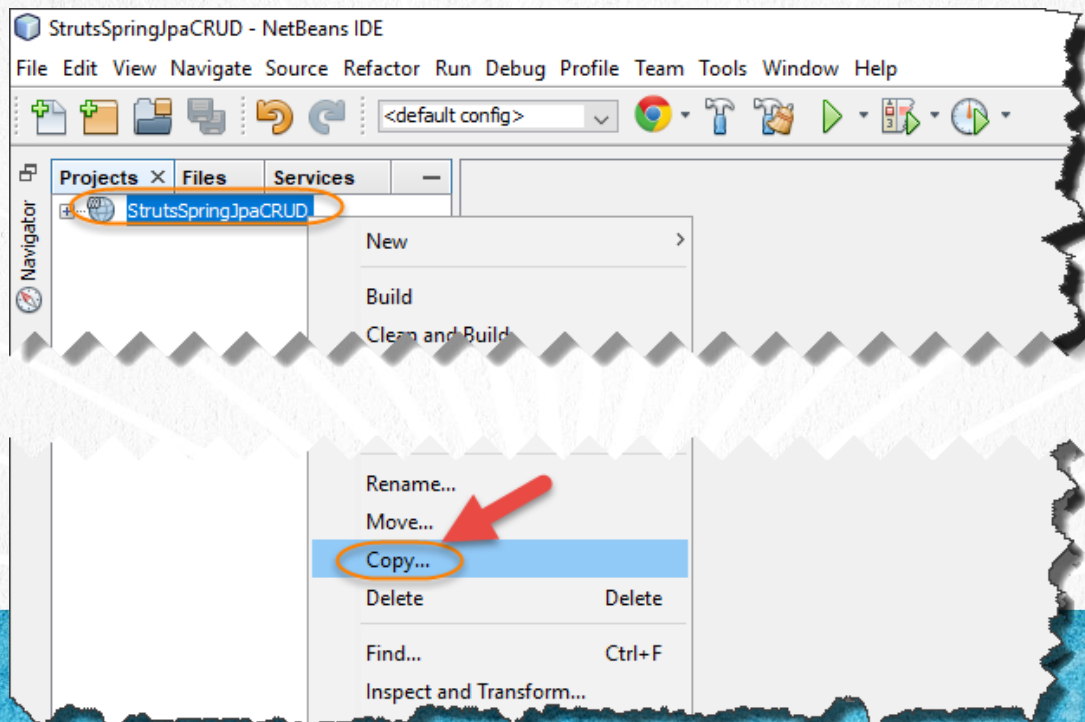
**CURSO STRUTS FRAMEWORK**

[www.globalmentoring.com.mx](http://www.globalmentoring.com.mx)



# 1. COPIAR PROYECTO

Vamos a partir del ejercicio StrutsSpringJpaCRUD para crear nuestro proyecto ya que muchas de los archivos son idénticos, y solo modificaremos los archivos que sean necesarios. Copiamos el proyecto StrutsSpringJpaCRUD y lo renombramos a StrutsSpringHibernateCRUD:

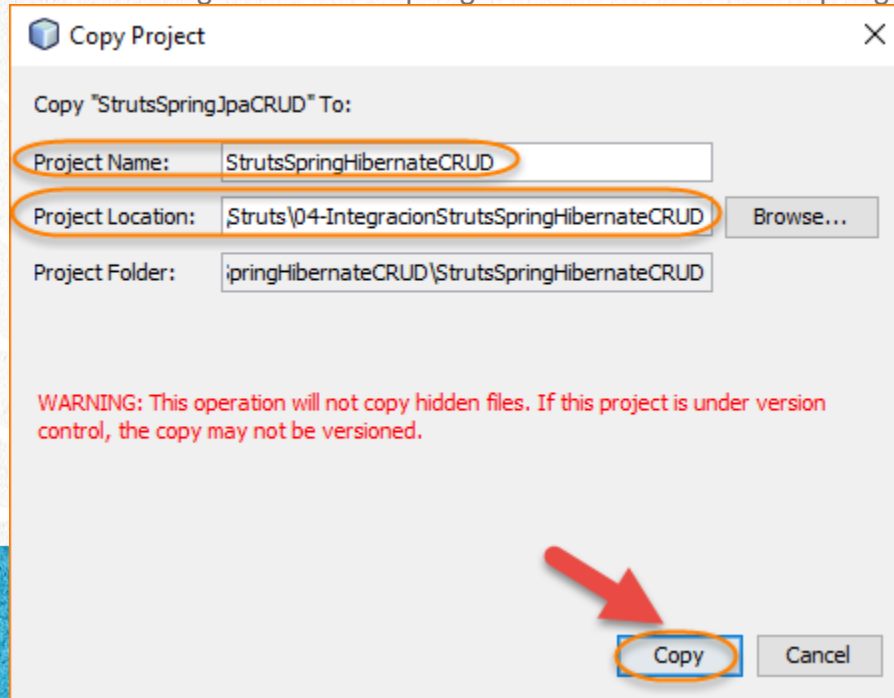


# 1. COPIAR PROYECTO

Renombramos el proyecto a StrutsSpringHibernateCRUD y la ruta puede cambiar, nosotros vamos a utilizar la siguiente ruta:

**Nombre del Proyecto:** StrutsSpringHibernateCRUD

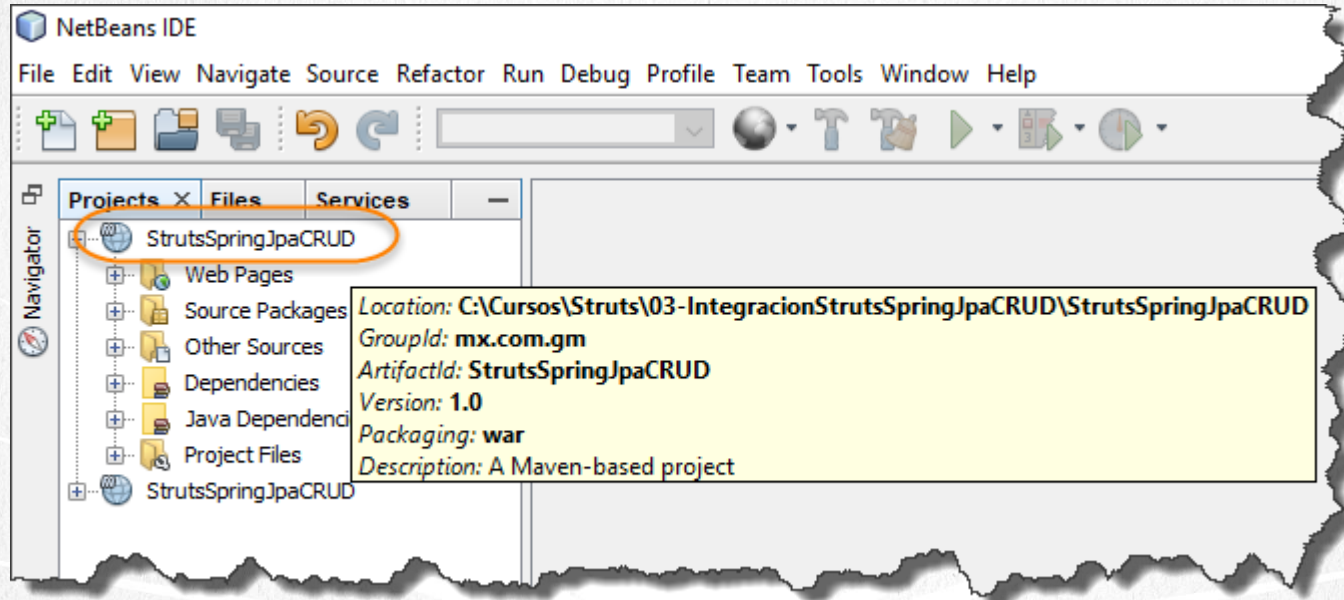
**Project Location:** C:\Cursos\Struts\04-IntegracionStrutsSpringHibernateCRUD\StrutsSpringHibernateCRUD





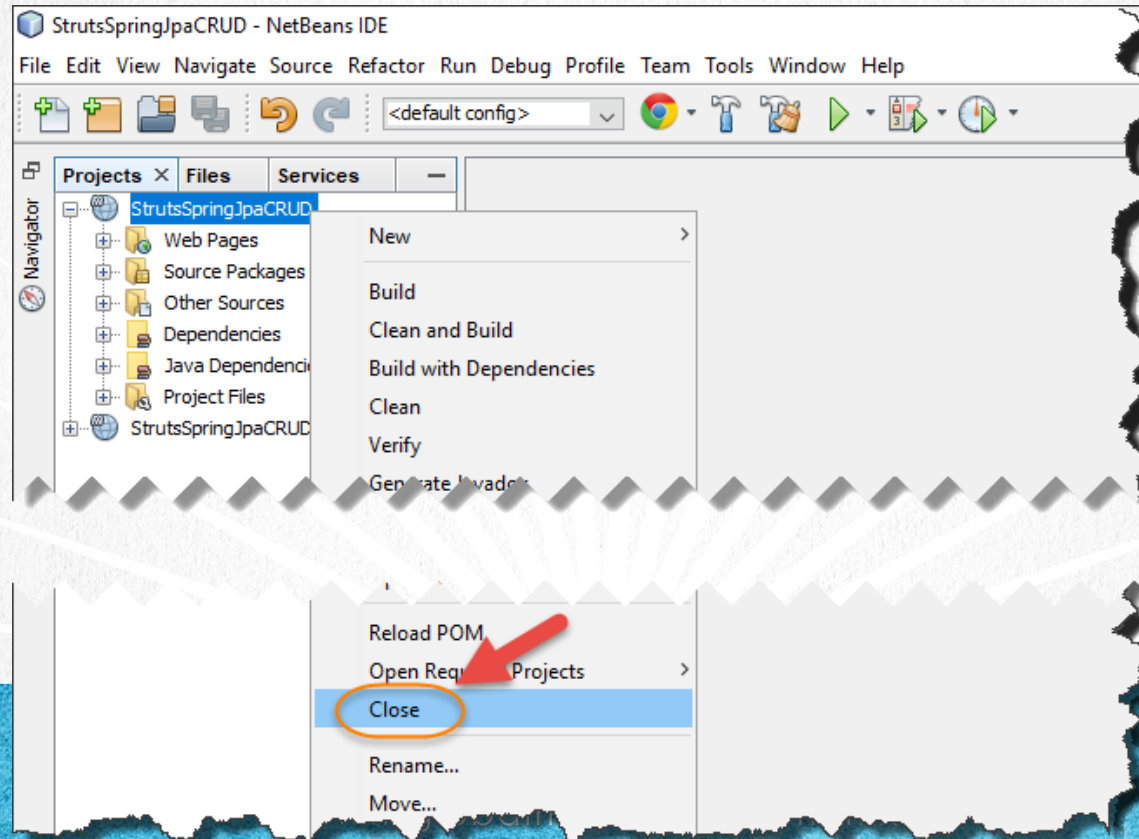
## 2. CERRAR PROYECTO

- Identificamos el proyecto que deseamos cerrar:



## 2. CERRAR PROYECTO

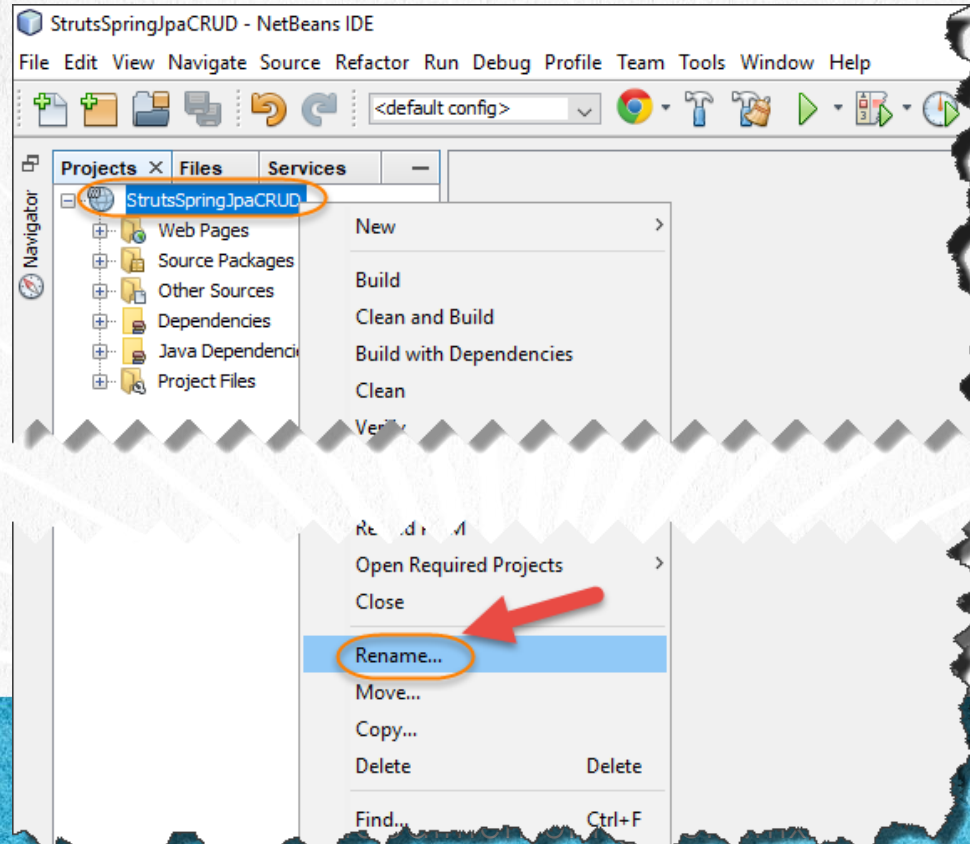
- Y Cerramos el proyecto que ya no vamos a utilizar:





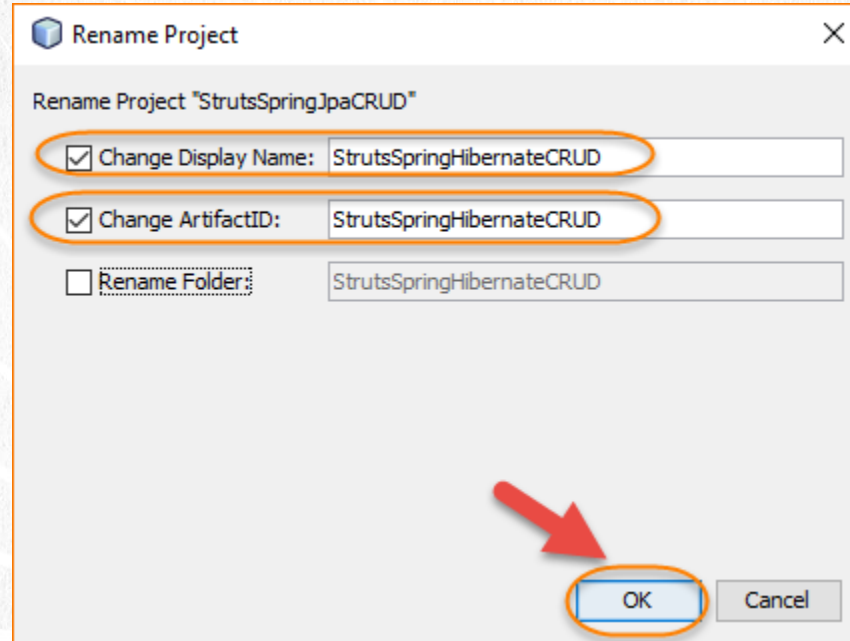
# 3. RENOMBAMOS EL PROYECTO

- Finalmente renombramos el proyecto como sigue:



### 3. RENOMBAMOS EL PROYECTO

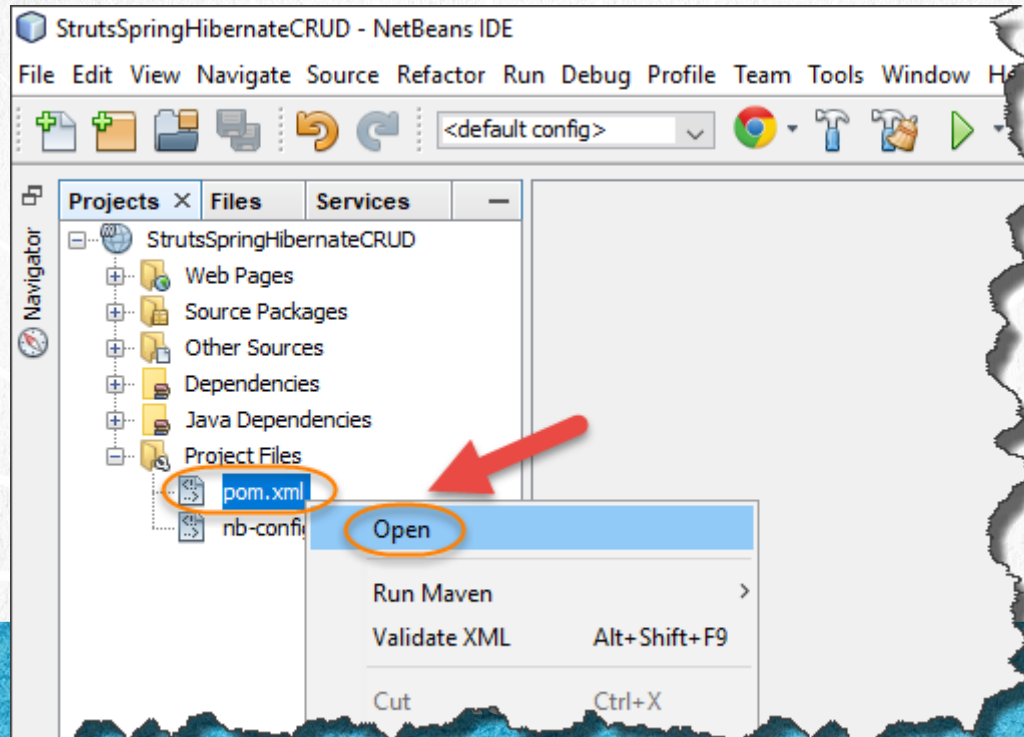
- Finalmente renombramos el proyecto como sigue:





## 4. ABRIMOS EL ARCHIVO POM.XML DE MAVEN

- Abrimos el archivo pom.xml de maven para agregar la librería de Hibernate. Es el único cambio que se agregará al archivo. Veamos como queda:



# PASO 4. 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>StrutsSpringHibernateCRUD</artifactId>
    <version>1.0</version>
    <packaging>war</packaging>

    <name>StrutsSpringHibernateCRUD</name>

    <properties>
        <project.build.sourceEncoding>UTF-8</project.build.sourceEncoding>
        <spring.version>5.0.4.RELEASE</spring.version>
    </properties>

    <dependencies>
        <dependency>
            <groupId>javax</groupId>
            <artifactId>javaee-web-api</artifactId>
            <version>8.0</version>
            <scope>provided</scope>
        </dependency>
```



# PASO 4. MODIFICAMOS EL CÓDIGO

Archivo pom.xml:

Clic para ver el  
archivo

```
<dependency>
  <groupId>org.apache.struts</groupId>
  <artifactId>struts2-core</artifactId>
  <version>2.5.14.1</version>
</dependency>
<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>org.apache.struts</groupId>
  <artifactId>struts2-convention-plugin</artifactId>
  <version>2.5.14.1</version>
</dependency>
<dependency>
  <groupId>org.ow2.asm</groupId>
  <artifactId>asm</artifactId>
  <version>6.0</version>
</dependency>
```

# PASO 4. MODIFICAMOS EL CÓDIGO

## Archivo pom.xml:

Clic para ver el  
archivo

```
<!--Spring-->
<dependency>
  <groupId>org.springframework</groupId>
  <artifactId>spring-core</artifactId>
  <version>${spring.version}</version>
</dependency>
<dependency>
  <groupId>org.springframework</groupId>
  <artifactId>spring-context</artifactId>
  <version>${spring.version}</version>
</dependency>
<dependency>
  <groupId>org.springframework</groupId>
  <artifactId>spring-test</artifactId>
  <version>${spring.version}</version>
</dependency>
<dependency>
  <groupId>org.springframework</groupId>
  <artifactId>spring-orm</artifactId>
  <version>${spring.version}</version>
</dependency>
<dependency>
  <groupId>org.springframework</groupId>
  <artifactId>spring-web</artifactId>
  <version>${spring.version}</version>
</dependency>
```



# PASO 4. MODIFICAMOS EL CÓDIGO

Archivo pom.xml:

Clic para ver el  
archivo

```
<dependency>
  <groupId>org.springframework</groupId>
  <artifactId>spring-webmvc</artifactId>
  <version>${spring.version}</version>
</dependency>
<!--Struts 2 y Spring integracion-->
<dependency>
  <groupId>org.apache.struts</groupId>
  <artifactId>struts2-spring-plugin</artifactId>
  <version>2.5.14.1</version>
</dependency>
<!--Hibernate-->
<dependency>
  <groupId>org.hibernate</groupId>
  <artifactId>hibernate-core</artifactId>
  <version>5.2.15.Final</version>
</dependency>
<!-- MySql -->
<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](http://www.globalmentoring.com.mx)

# PASO 4. MODIFICAMOS EL CÓDIGO

## Archivo pom.xml:

Clic para ver el  
archivo

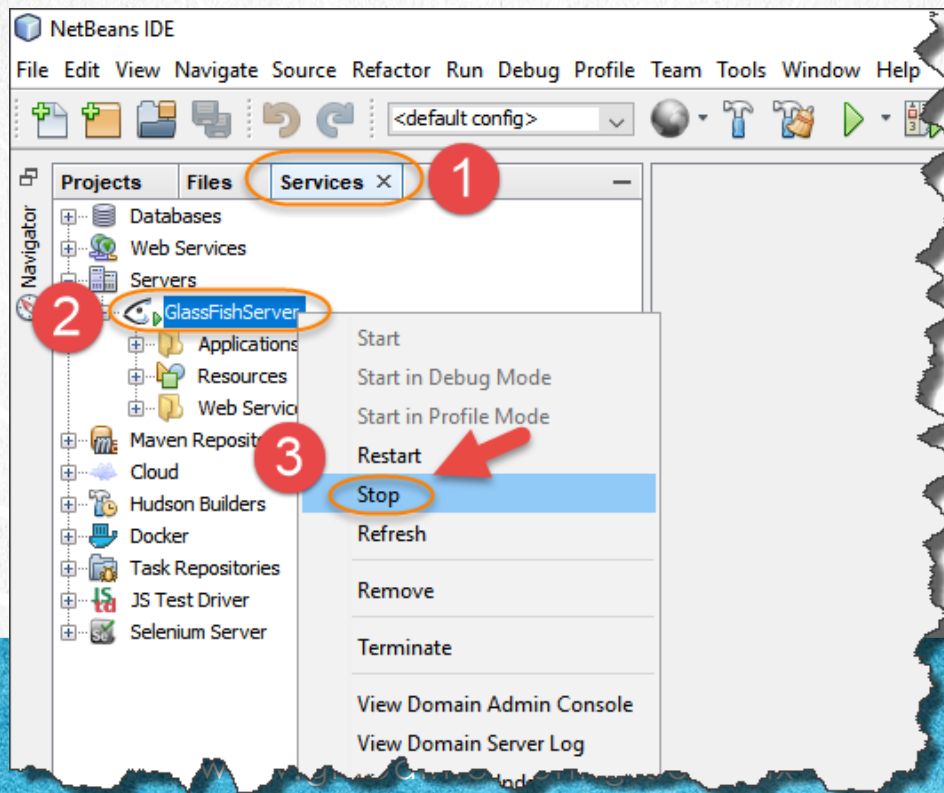
```
<build>
  <plugins>
    <plugin>
      <groupId>org.apache.maven.plugins</groupId>
      <artifactId>maven-war-plugin</artifactId>
      <version>2.3</version>
      <configuration>
        <failOnMissingWebXml>>false</failOnMissingWebXml>
      </configuration>
    </plugin>
    <plugin>
      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



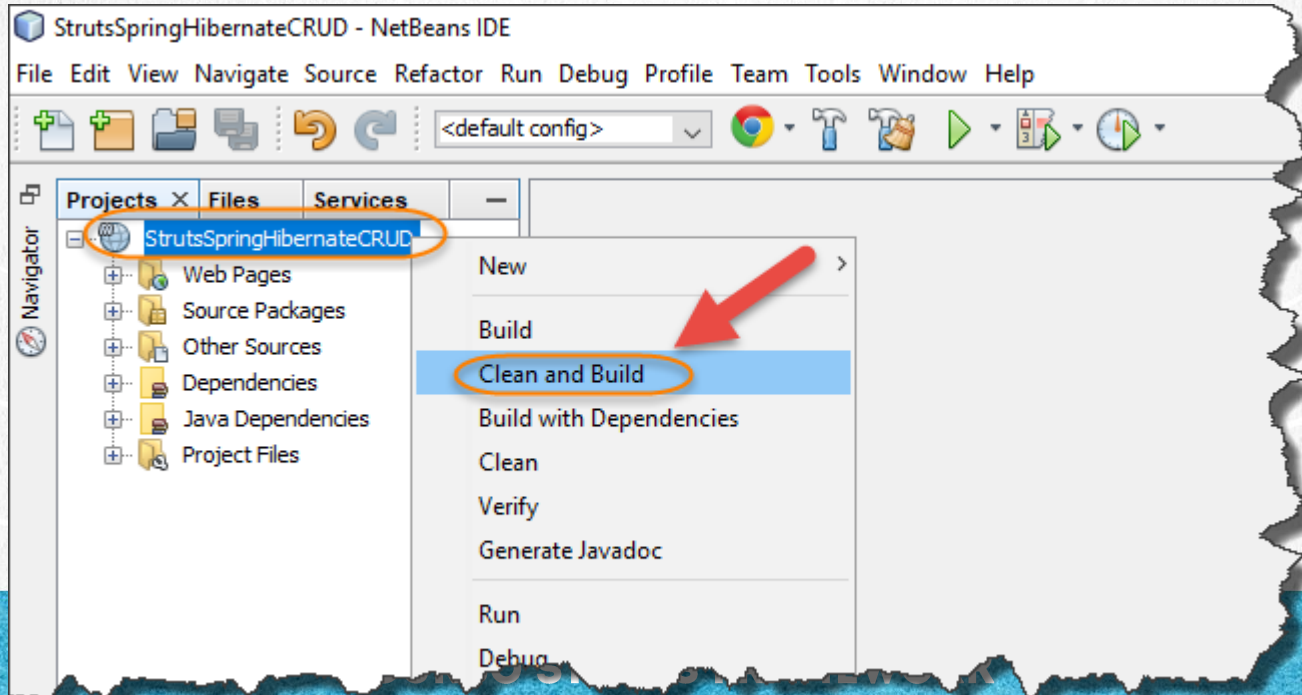
# 5. DETENEMOS GLASSFISH SI ESTUVIERA INICIADO

- Antes de hacer Clean & Build del proyecto para que descargue las librerías si fuera necesario, verificamos que el servidor de Glassfish no esté iniciado ya que puede haber problemas para hacer el proceso de Clean & build si el servidor está iniciado.



## 6. HACEMOS CLEAN & BUILD

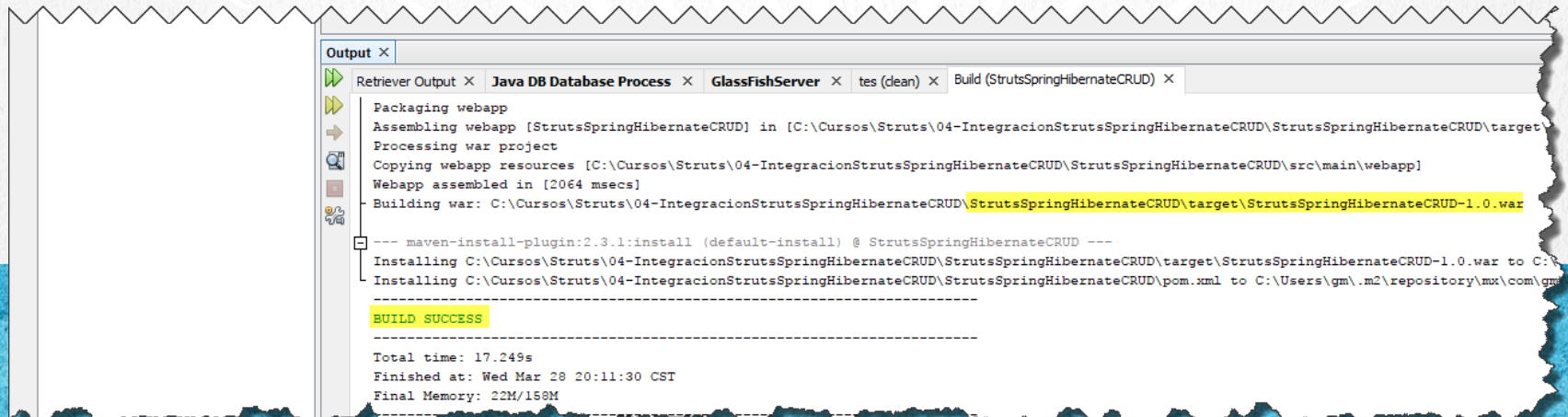
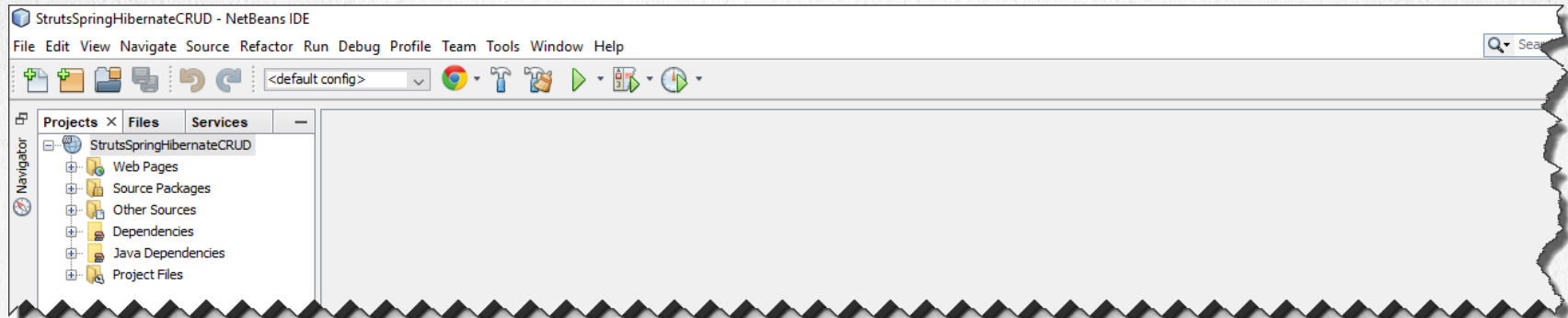
• Para que se descarguen las nuevas librerías si fuera necesario, hacemos Clean & Build al proyecto. Si por alguna razón este proceso falla, se debe desactivar cualquier software como antivirus, Windows defender o firewall durante este proceso para que no se impida la descarga de archivos .jar de Java.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





## 6. HACEMOS CLEAN & BUILD

- Si ya no fue necesario descargar ninguna librería debido a que podría ya tener todas descargadas, el proceso es más rápido. Al final deberemos observar lo siguiente:



# 7. MODIFICAMOS EL ARCHIVO PERSISTENCE.XML

Otro de los cambios que necesitamos realizar es en el archivo persistence.xml, ya que precisamente aquí es donde vamos a utilizar la implementación de Hibernate, por lo tanto necesitamos especificar que ahora el proveedor de JPA será la implementación de Hibernate.

Veamos como queda nuestro archivo persistence.xml:



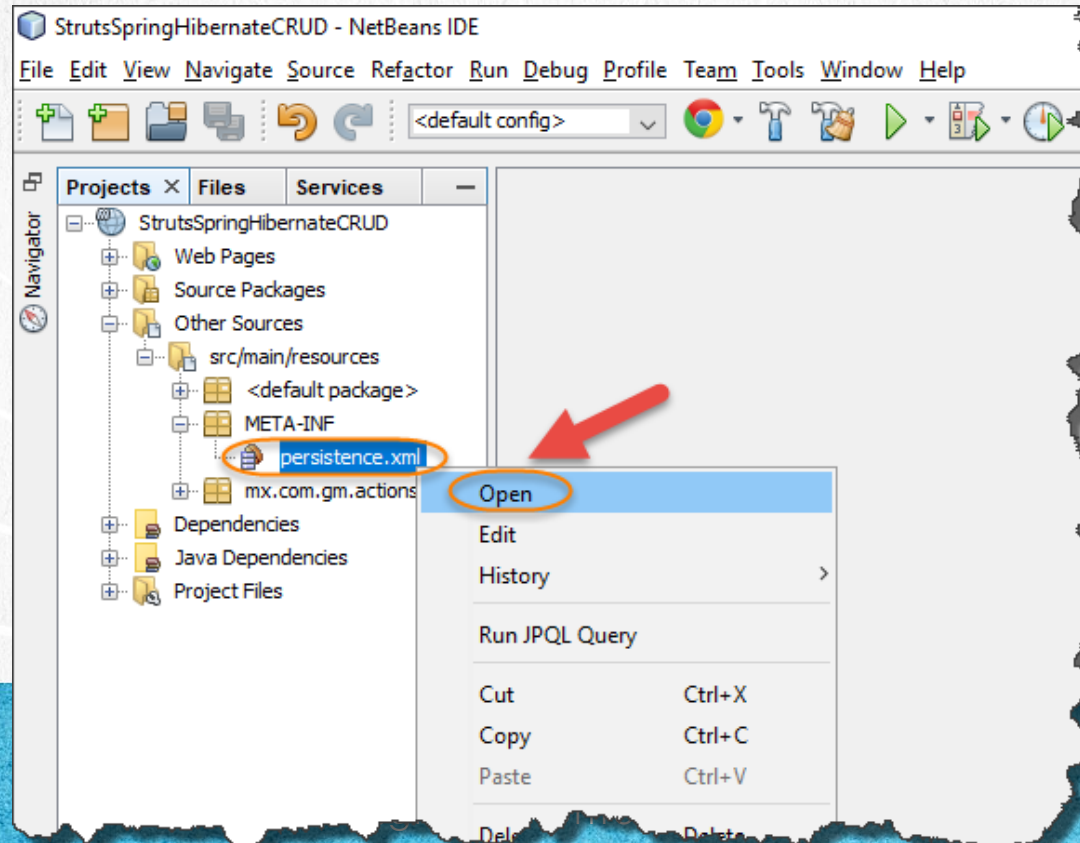
Experiencia y Conocimiento para tu vida

**CURSO STRUTS FRAMEWORK**

[www.globalmentoring.com.mx](http://www.globalmentoring.com.mx)

# 7. MODIFICAMOS EL ARCHIVO PERSISTENCE.XML

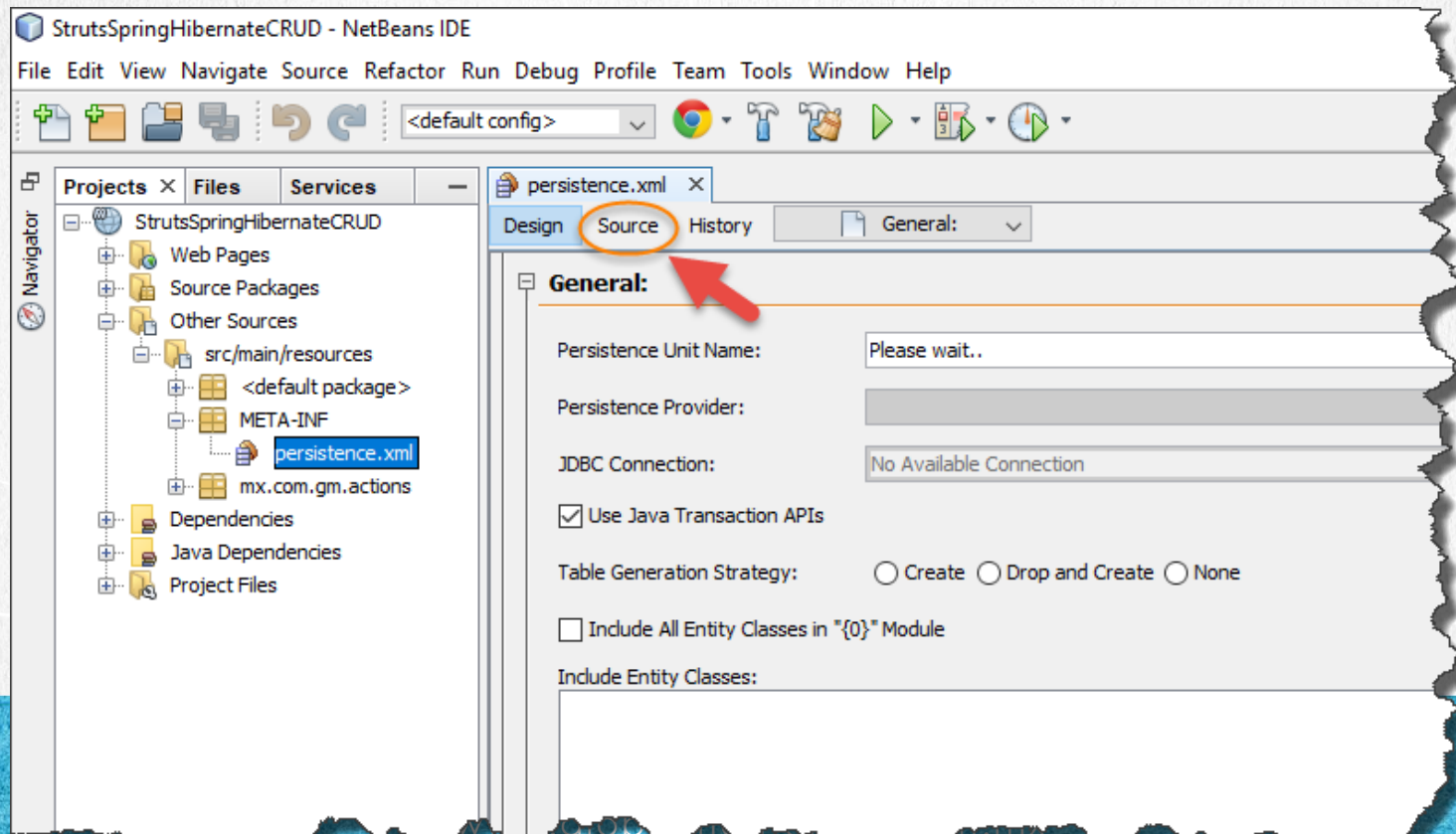
- Abrimos el archivo persistence.xml para modificarlo:





# 7. MODIFICAMOS EL ARCHIVO PERSISTENCE.XML

- Si nos muestra la vista siguiente, damos clic en Source para modificar el código fuente:



# PASO 7. MODIFICAMOS EL CÓDIGO

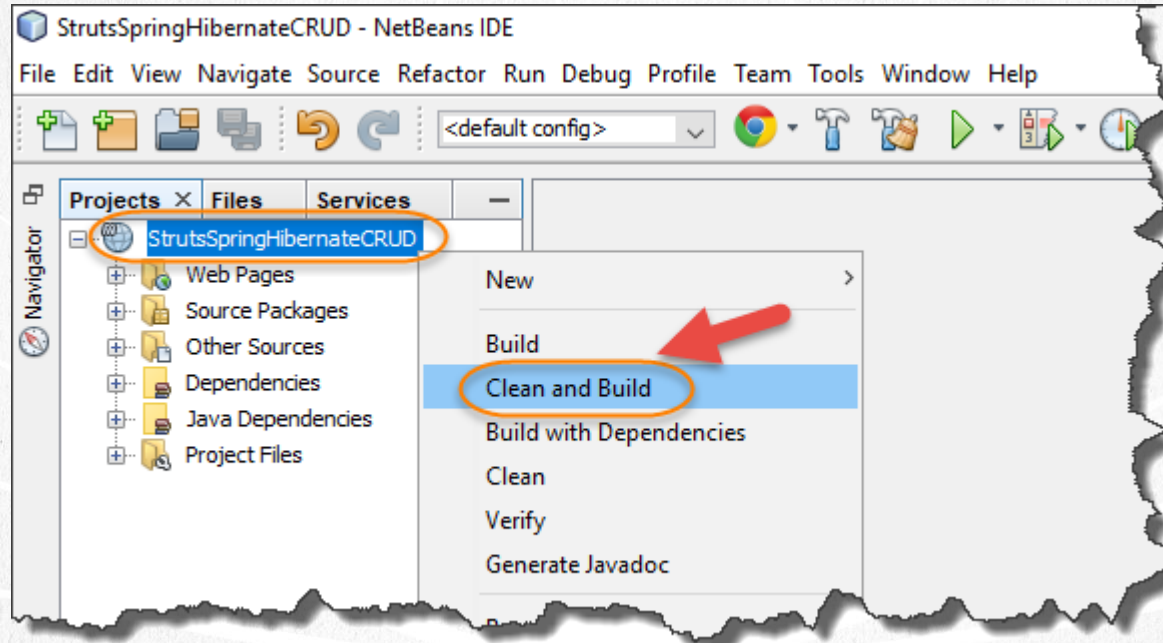
## Archivo persistence.xml:

Clic para ver el  
archivo

```
<?xml version="1.0" encoding="UTF-8"?>
<persistence xmlns="http://xmlns.jcp.org/xml/ns/persistence"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://xmlns.jcp.org/xml/ns/persistence
    http://xmlns.jcp.org/xml/ns/persistence/persistence_2_2.xsd"
  version="2.2">
  <persistence-unit name="PersistenceUnit" transaction-type="JTA">
    <provider>org.hibernate.jpa.HibernatePersistenceProvider</provider>
    <jta-data-source>jdbc/PersonaDb</jta-data-source>
    <properties>
      <property name="hibernate.show_sql" value="true"/>
      <property name="hibernate.transaction.jta.platform"
        value="org.hibernate.service.jta.platform.internal.SunOneJtaPlatform"/>
    </properties>
  </persistence-unit>
</persistence>
```

# PASO 8. HACEMOS CLEAN & BUILD

- Hacemos Clean & Build para asegurarnos que tenemos todo listo:



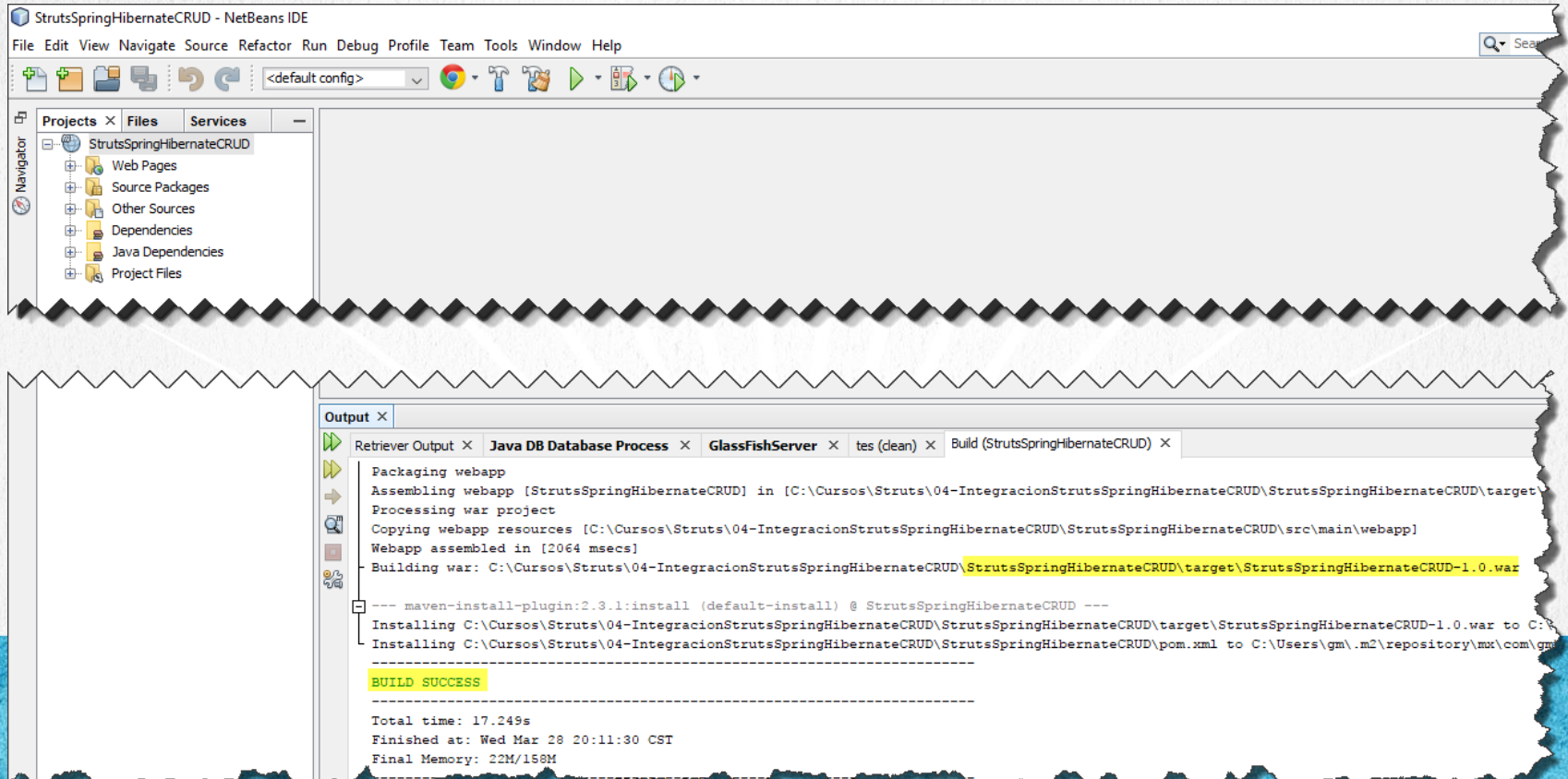
**CURSO STRUTS FRAMEWORK**

[www.globalmentoring.com.mx](http://www.globalmentoring.com.mx)



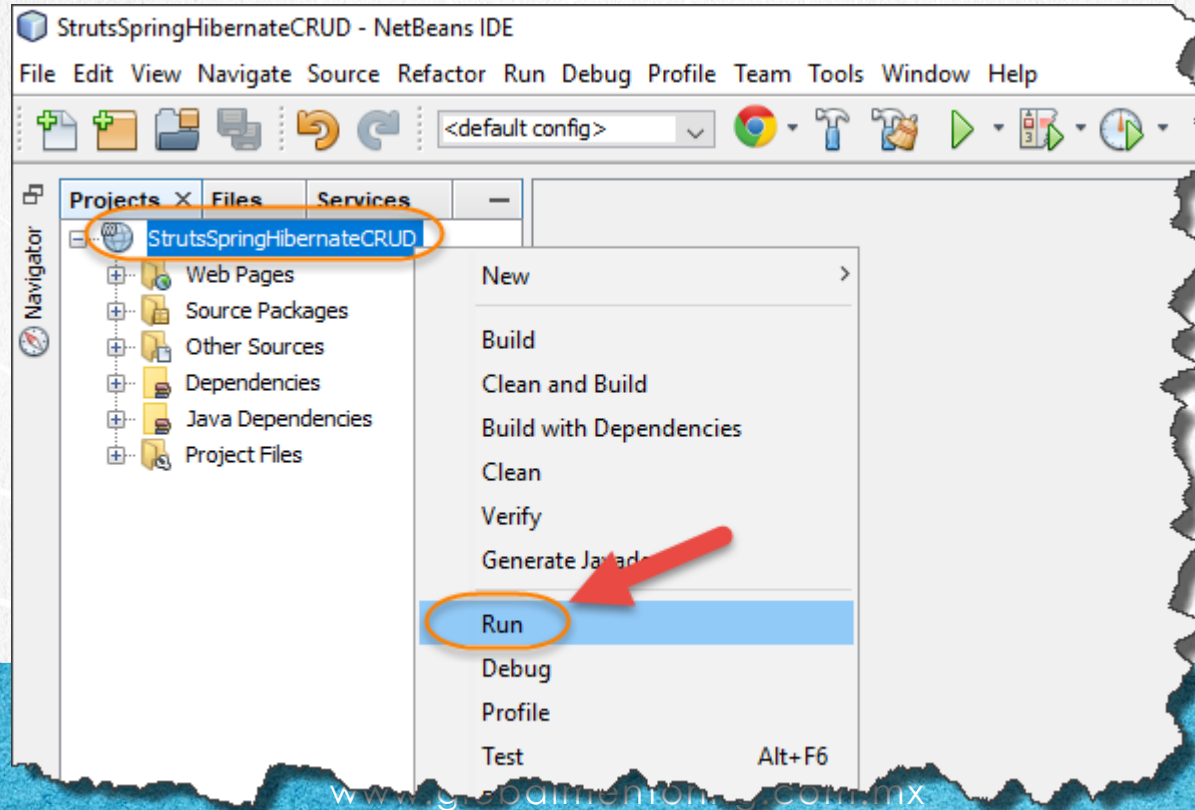
# PASO 8. EJECUTAMOS CLEAN & BUILD

- Observamos que el proceso se haya ejecutado con éxito (Build Success):



# PASO 9. EJECUTAMOS LA APLICACIÓN

- Ejecutamos la aplicación StrutsSpringHibernateCRUD como sigue:



# PASO 9. EJECUTAMOS LA APLICACIÓN

- Observamos el resultado como sigue, y damos clic en el link:





# PASO 9. EJECUTAMOS LA APLICACIÓN

- Observamos que obtenemos la misma información, ya que solo hicimos el cambio de proveedor de la capa de datos de EclipseLink a Hibernate, pero debido a que estamos usando el estándar de JPA los cambios fueron muy sencillos de realizar.



Personas con Struts 2

[Agregar Persona](#)

idPersona	Nombre	Apellido Paterno	Apellido Materno	Email	Editar	Eliminar
1	Administrador	Esparza	Sistemas	mesparza@mail.com	<a href="#">Editar</a>	<a href="#">Eliminar</a>
2	Carlos	Romero	Esparza	carlos.romero@gmail.com	<a href="#">Editar</a>	<a href="#">Eliminar</a>
3	Karla	Lopez	Perez	klopez@mail.com	<a href="#">Editar</a>	<a href="#">Eliminar</a>
7	Maria	Toledo	Ramírez	mtoledo@mail.com	<a href="#">Editar</a>	<a href="#">Eliminar</a>

# RECOMENDACIONES FINALES

Si por alguna razón falla el ejercicio, se pueden hacer varias cosas para corregirlo:

- 1) Detener el servidor de Glassfish
- 2) Hacer un Clean & Build del proyecto para tener la versión más reciente compilada
- 3) Volver a hacer Run del proyecto (desplegar nuevamente el proyecto en el servidor)

Si lo anterior no funciona, pueden probar cargando el proyecto resuelto el cual es funcional 100% y descartar problemas de configuración en su ambiente o cualquier otro error de código.

La configuración por convenciones de Struts 2, es muy sensible, de tal manera que todo debe estar escrito tal como se especificó en el ejercicio, ya que cualquier cambio en los nombres provocará que no se ejecute correctamente el ejercicio.

La integración con otros frameworks y tecnologías como Spring y JPA también es muy propenso a errores, así que puedes apoyarte del proyecto resuelto que te entregamos, el cual es 100% funcional, y así apoyarte en cualquier momento de esta documentación y los proyectos resueltos que te entregamos.

**CURSO STRUTS FRAMEWORK**

[www.globalmentoring.com.mx](http://www.globalmentoring.com.mx)



# CONCLUSIÓN DEL EJERCICIO

Con este ejercicio hemos creado una aplicación que integra las 3 tecnologías como son:

- Struts 2
- Spring Framework
- JPA/Hibernate (Hibernate como implementación del estándar JPA)

En este ejercicio hemos aplicado las operaciones CRUD (Create-Read-Update-Delete) para la tabla de persona.

Se deja como ejercicio probar cada una de las acciones para comprobar que todo sigue funcionando solo con los pocos cambios aplicados al proyecto.



**CURSO ONLINE**

# **STRUTS 2 FRAMEWORK**

---

Por: Ing. Ubaldo Acosta



Experiencia y Conocimiento para tu vida



**CURSO STRUTS FRAMEWORK**

[www.globalmentoring.com.mx](http://www.globalmentoring.com.mx)