

CURSO DE JAVASERVER FACES

EJERCICIO

HOLAMUNDO CON JAVASERVER FACES



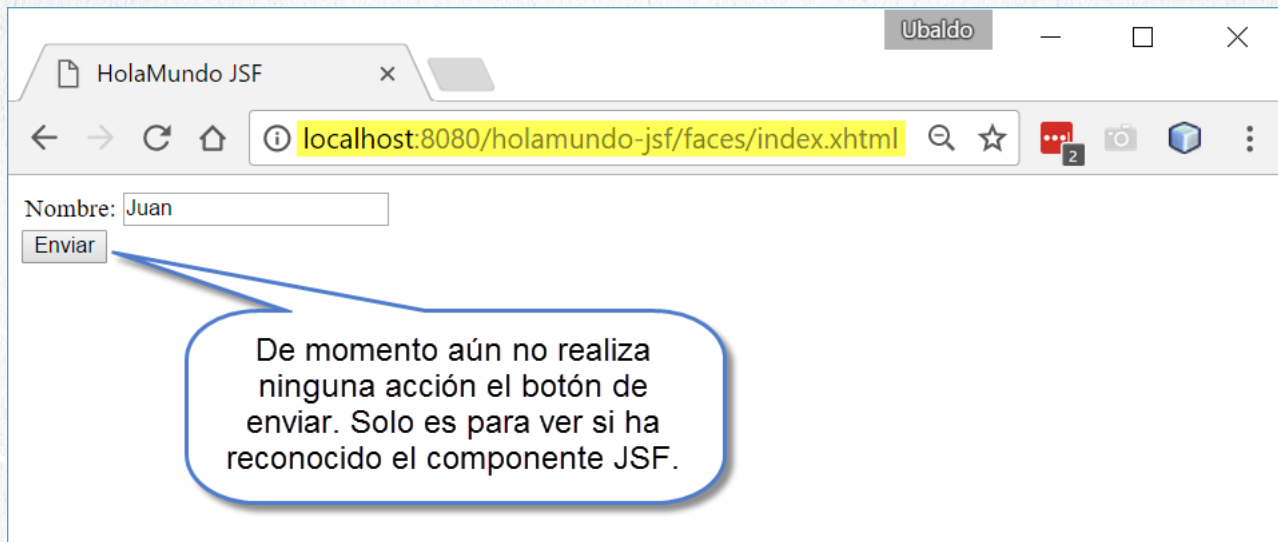
Experiencia y Conocimiento para tu vida

CURSO DE JAVASERVER FACES

www.globalmentoring.com.mx

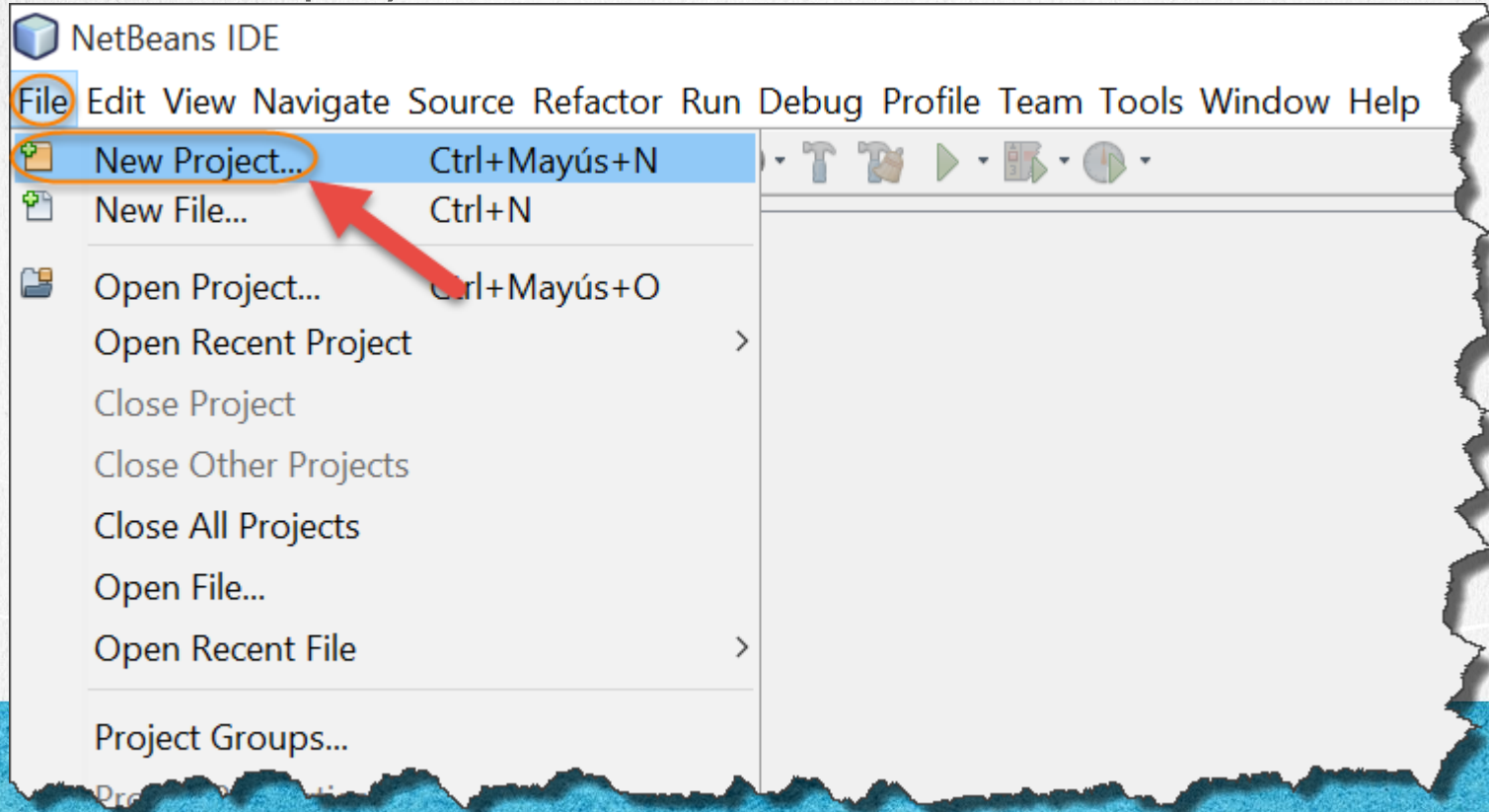
OBJETIVO DEL EJERCICIO

- El objetivo del ejercicio es configurar un proyecto HolaMundo con JSF. Nos apoyaremos de Maven para la creación del proyecto. El resultado debe ser similar al siguiente:



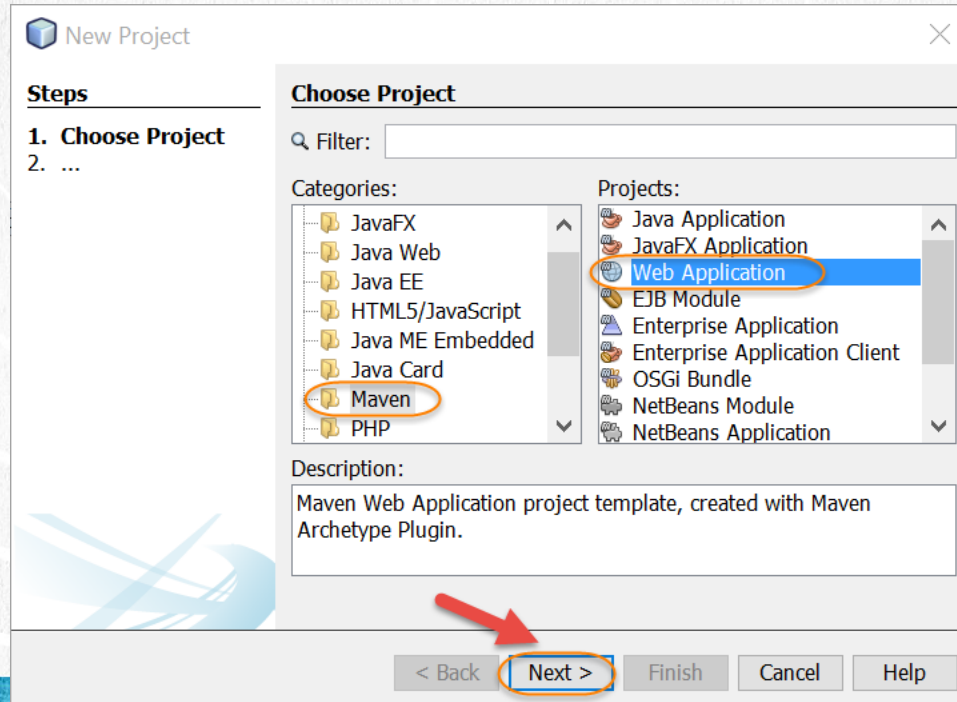
PASO 1. CREAR UN PROYECTO MAVEN

Creamos un nuevo proyecto Maven:



PASO 1. CREAR UN PROYECTO MAVEN

Seleccionando un nuevo proyecto Maven:



CURSO DE JAVASERVER FACES

www.globalmentoring.com.mx

PASO 1. CREAR UN PROYECTO MAVEN

Escribimos los valores por siguientes:

New Web Application

Steps

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

Name and Location

Project Name: holamundo-jsf

Project Location: C:\Cursos\JSF\Leccion01 Browse...

Project Folder: C:\Cursos\JSF\Leccion01\holamundo-jsf

Artifact Id: holamundo-jsf

Group Id: mx.com.gm

Version: 1.0

Package: (Optional)

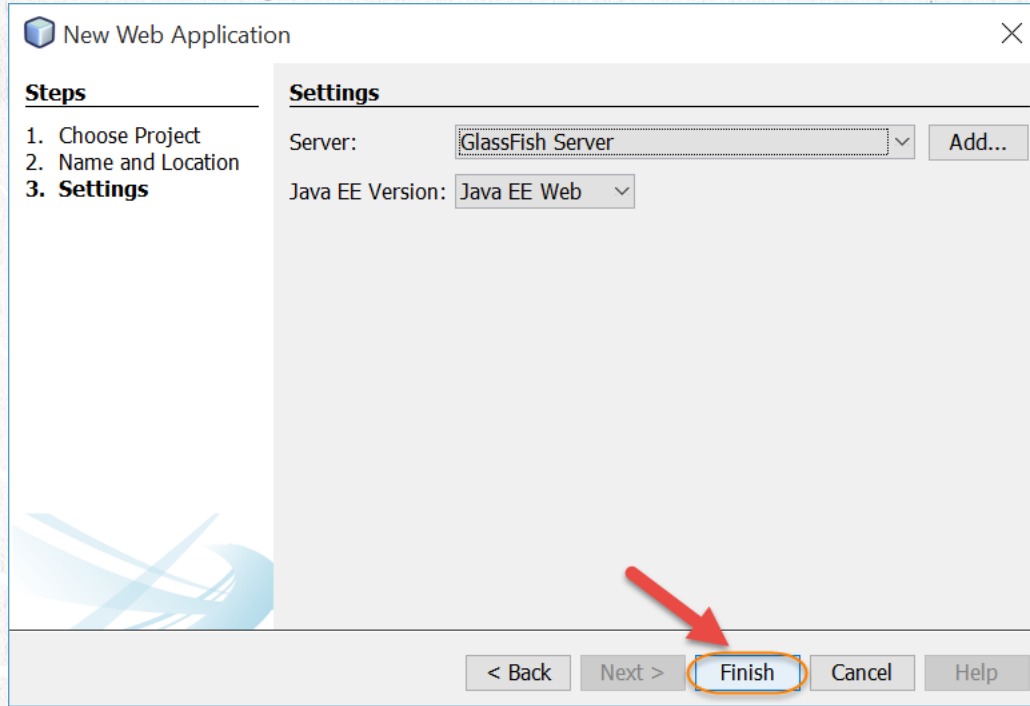
< Back **Next >** Finish Cancel Help

CURSO DE JAVASERVER FACES

www.globalmentoring.com.mx

PASO 1. CREAR UN PROYECTO MAVEN

Escribimos los valores por siguientes:



New Web Application

Steps

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

Settings

Server: GlassFish Server Add...

Java EE Version: Java EE Web

< Back Next > **Finish** Cancel Help

CURSO DE JAVASERVER FACES

www.globalmentoring.com.mx

PASO 2. AGREGAR LIBRERÍAS DE JSF

Vamos a utilizar Maven para administrar las dependencias que vamos a utilizar en el proyecto. La librería de javaee-web ya incluye las dependencias necesarias para con JSF.

Agregamos las siguientes librerías .jar al archivo pom.xml:

- javaee-web
- log4j



Experiencia y Conocimiento para tu vida

CURSO DE JAVASERVER FACES

www.globalmentoring.com.mx

PASO 2. MODIFICAMOS EL CÓDIGO

Archivo pom.xml:

Dar click para ir al código

```
<?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>holamundo-jsf</artifactId>
  <version>1.0</version>
  <packaging>war</packaging>
  <name>holamundo-jsf</name>
  <properties>
    <endorsed.dir>${project.build.directory}/endorsed</endorsed.dir>
    <project.build.sourceEncoding>UTF-8</project.build.sourceEncoding>
    <log4j.version>2.8.2</log4j.version>
  </properties>
  <dependencies>
    <dependency>
      <groupId>javax</groupId>
      <artifactId>javaee-web-api</artifactId>
      <version>7.0</version>
      <scope>provided</scope>
    </dependency>
    <!--log4j2-->
    <dependency>
      <groupId>org.apache.logging.log4j</groupId>
      <artifactId>log4j-api</artifactId>
      <version>${log4j.version}</version>
    </dependency>
```


PASO 2. MODIFICAMOS EL CÓDIGO

Archivo pom.xml:

Dar click para ir al código

```
<dependency>
  <groupId>org.apache.logging.log4j</groupId>
  <artifactId>log4j-core</artifactId>
  <version>${log4j.version}</version>
</dependency>
<dependency>
  <groupId>org.apache.logging.log4j</groupId>
  <artifactId>log4j-jcl</artifactId>
  <version>${log4j.version}</version>
</dependency>
</dependencies>
<build>
  <plugins>
    <plugin>
      <groupId>org.apache.maven.plugins</groupId>
      <artifactId>maven-compiler-plugin</artifactId>
      <version>3.1</version>
      <configuration>
        <source>1.8</source>
        <target>1.8</target>
        <compilerArguments>
          <endorseddirs>${endorsed.dir}</endorseddirs>
        </compilerArguments>
      </configuration>
    </plugin>
```

PASO 2. MODIFICAMOS EL CÓDIGO

Archivo pom.xml:

Dar click para ir al código

```
<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-dependency-plugin</artifactId>
  <version>2.6</version>
  <executions>
    <execution>
      <phase>validate</phase>
      <goals>
        <goal>copy</goal>
      </goals>
    </execution>
  </executions>
</plugin>
```

PASO 2. MODIFICAMOS EL CÓDIGO

Archivo pom.xml:

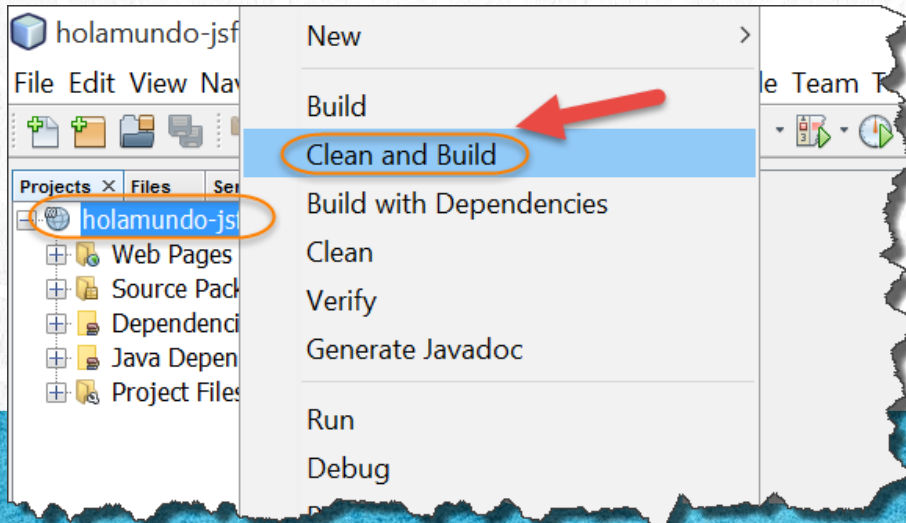
Dar click para ir al código

```
<configuration>
  <outputDirectory>${endorsed.dir}</outputDirectory>
  <silent>true</silent>
  <artifactItems>
    <artifactItem>
      <groupId>javax</groupId>
      <artifactId>javaee-endorsed-api</artifactId>
      <version>7.0</version>
      <type>jar</type>
    </artifactItem>
  </artifactItems>
</configuration>
</execution>
</executions>
</plugin>
</plugins>
</build>
</project>
```


PASO 3. ACTUALIZACIÓN DEL REPOSITORIO MAVEN)

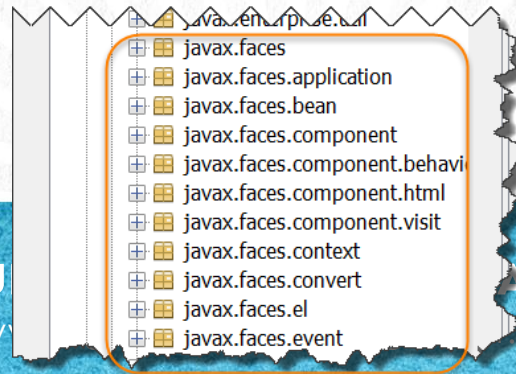
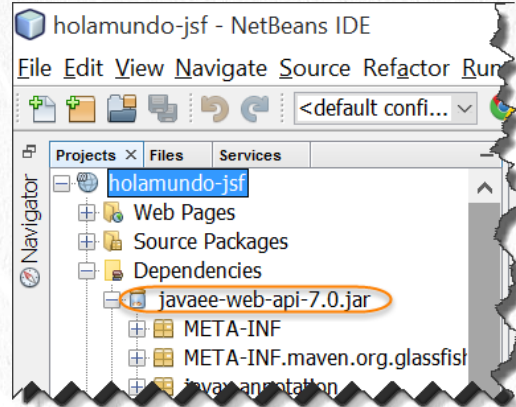
Para actualizar el repositorio de Maven, el cual se encargará de administrar las librerías .jar de nuestro proyecto, basta con hacer clean&build a nuestro proyecto

Nota: Si por alguna razón no se actualiza el repositorio, deshabilitar el antivirus o verificar si se tiene una configuración de proxy.



PASO 4. REVISIÓN DE LAS LIBRERÍAS

Revisamos que el proyecto tenga la librería de javaee-web y que observemos que cuente con las clases del paquete javax.faces:

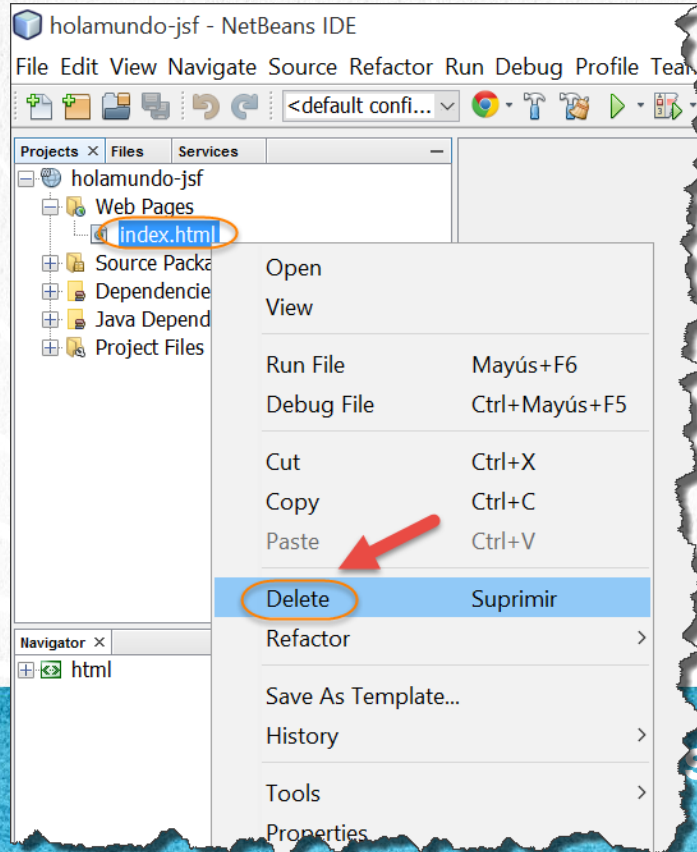


CU
W

FACES
mx

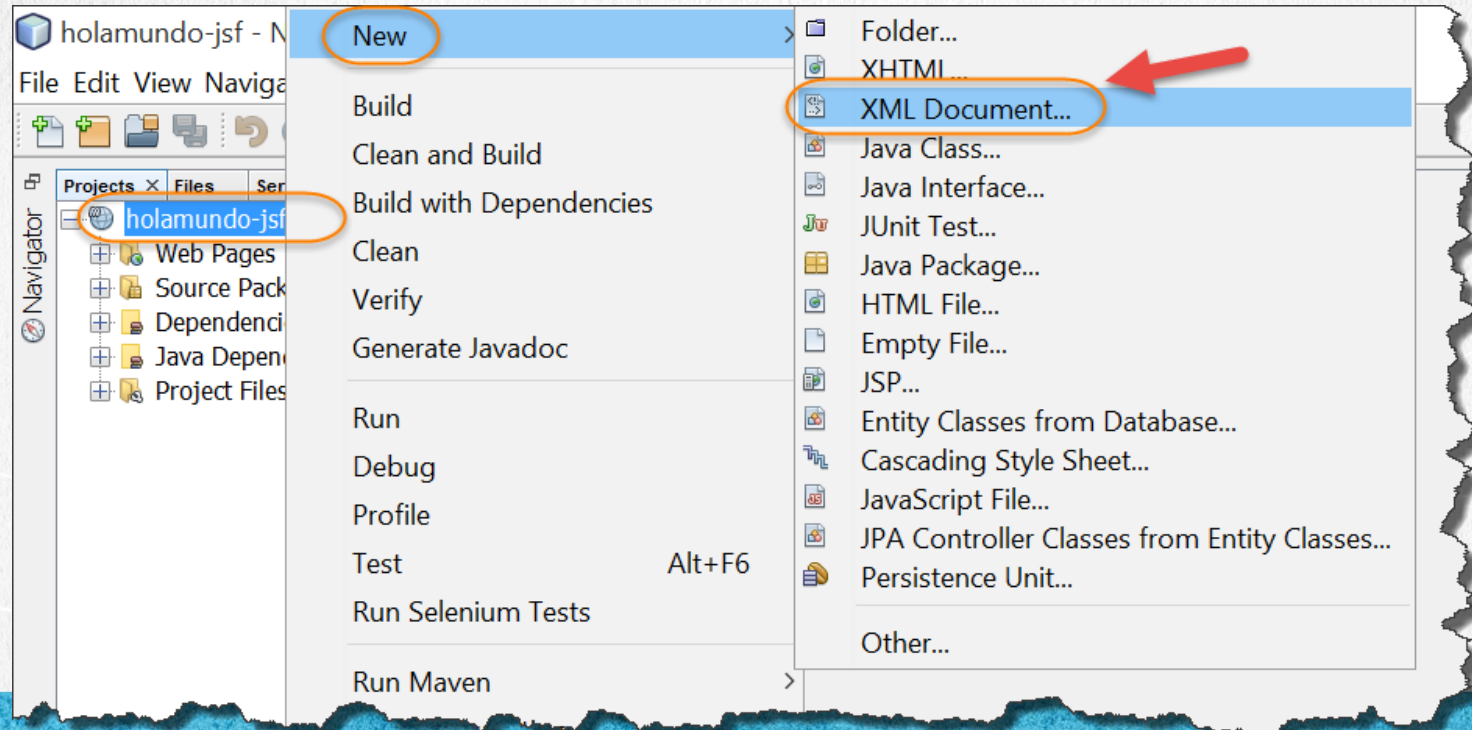
PASO 5. ELIMINAMOS EL ARCHIVO HTML

Si existe, eliminamos el archivo index.html:



PASO 6. CREAMOS UN ARCHIVO XML

Creamos el archivo web.xml:



CURSO DE JAVASERVER FACES

www.globalmentoring.com.mx

PASO 6. CREAMOS UN ARCHIVO XML

Creamos el archivo web.xml dentro de la carpeta WEB-INF según se observa:

New XML Document

Steps

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

Name and Location

File Name:

Project:

Folder:

Created File:

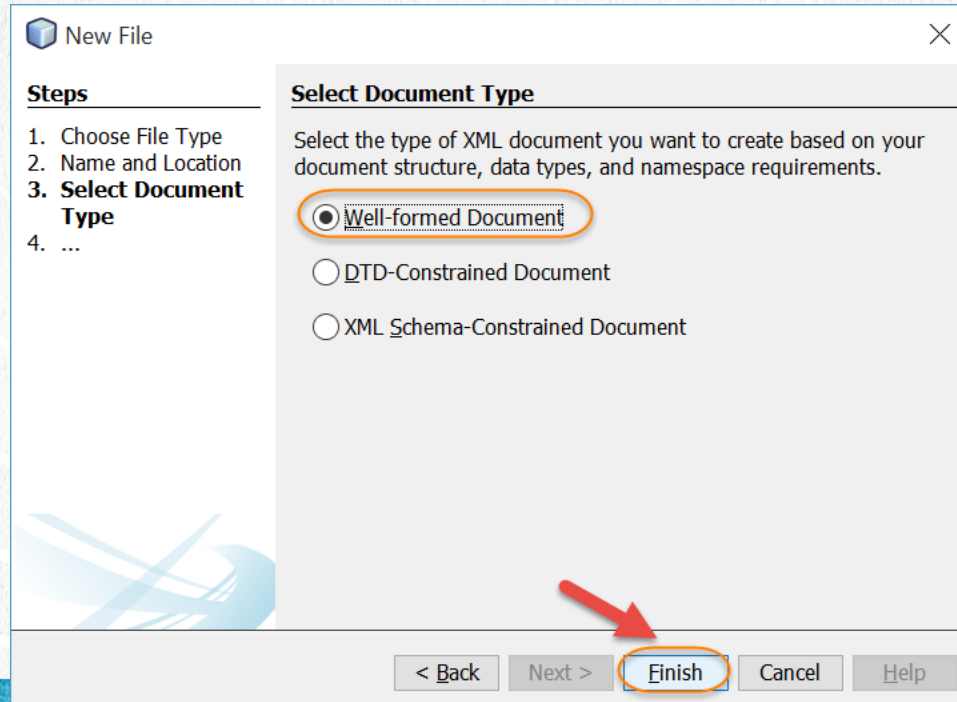
< Back **Next >** Finish Cancel Help

CURSO DE JAVASERVER FACES

www.globalmentoring.com.mx

PASO 6. CREAMOS UN ARCHIVO XML

Creamos el archivo web.xml. Seleccionamos cualquier opción, no es importante ya que sobrescribiremos el contenido del archivo xml:



CURSO DE JAVASERVER FACES

www.globalmentoring.com.mx

PASO 7. MODIFICAMOS EL CÓDIGO

Archivo web.xml:

Dar click para ir al código

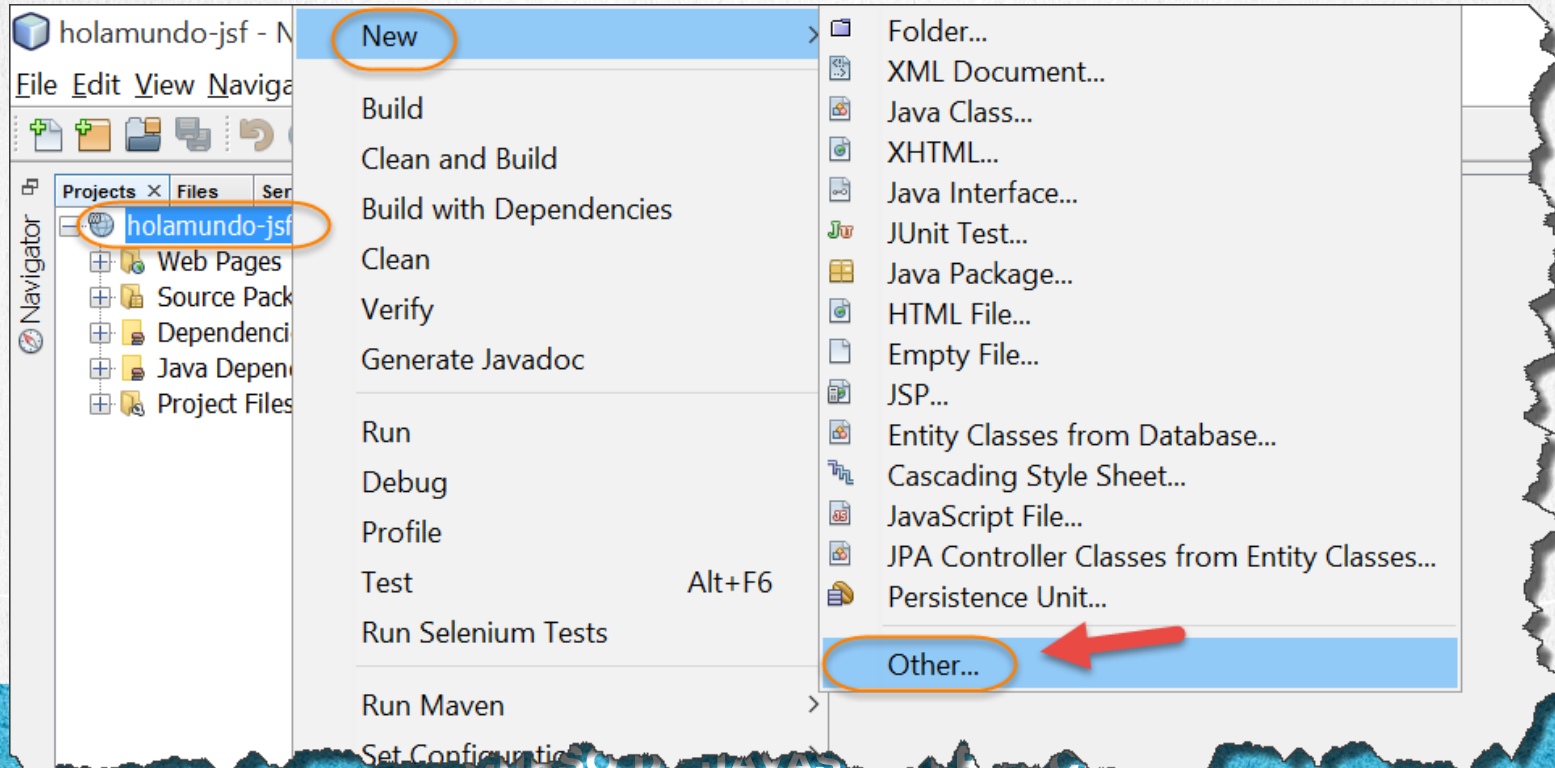
```
<web-app xmlns="http://xmlns.jcp.org/xml/ns/javaee"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://xmlns.jcp.org/xml/ns/javaee
    http://xmlns.jcp.org/xml/ns/javaee/web-app_3_1.xsd" version="3.1">
  <context-param>
    <param-name>javax.faces.PROJECT_STAGE</param-name>
    <param-value>Development</param-value>
  </context-param>
  <servlet>
    <servlet-name>Faces Servlet</servlet-name>
    <servlet-class>javax.faces.webapp.FacesServlet</servlet-class>
    <load-on-startup>1</load-on-startup>
  </servlet>
  <servlet-mapping>
    <servlet-name>Faces Servlet</servlet-name>
    <url-pattern>/faces/*</url-pattern>
  </servlet-mapping>
  <welcome-file-list>
    <welcome-file>faces/index.xhtml</welcome-file>
  </welcome-file-list>
</web-app>
```

CURSO DE JAVASERVER FACES

www.globalmentoring.com.mx

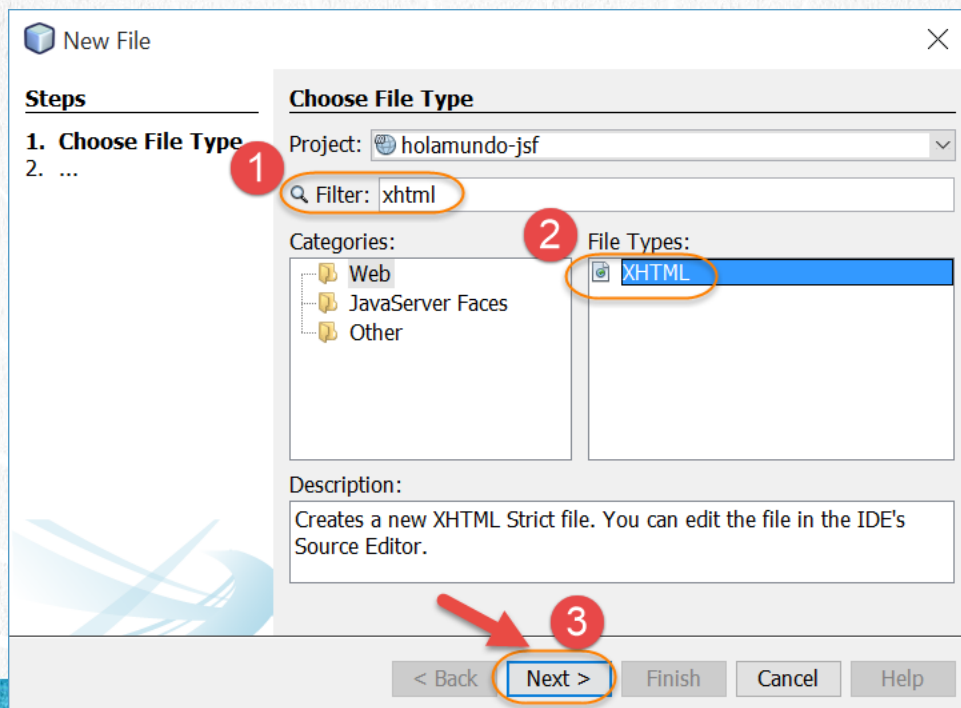
PASO 8. CREAMOS UN ARCHIVO XHTML

Creamos el archivo index.xhtml:



PASO 8. CREAMOS UN ARCHIVO XHTML

Creamos el archivo index.xhtml:



CURSO DE JAVASERVER FACES

www.globalmentoring.com.mx

PASO 8. CREAMOS UN ARCHIVO XHTML

Creamos el archivo index.xhtml:

New XHTML

Steps

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

Name and Location

XHTML File Name:

Project:

Location:

Folder:

Created File:

CURSO DE JAVASERVER FACES

www.globalmentoring.com.mx

PASO 9. MODIFICAMOS EL CÓDIGO

Archivo index.xhtml:

Dar click para ir al código

```
<?xml version='1.0' encoding='UTF-8' ?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml"
      xmlns:h="http://java.sun.com/jsf/html">
  <h:head>
    <title>HolaMundo JSF</title>
  </h:head>
  <h:body>
    <h:form>
      <table>
        <tr>
          <td><h:outputLabel for="nombre" value="Nombre:" /></td>
          <td><h:inputText id="nombre" /></td>
          <td><h:message for="nombre" /></td>
        </tr>
      </table>
      <h:commandButton value="Enviar" />
    </h:form>
  </h:body>
</html>
```

CURSO DE JAVASERVER FACES

www.globalmentoring.com.mx

10. CREAMOS UN ARCHIVO XML

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

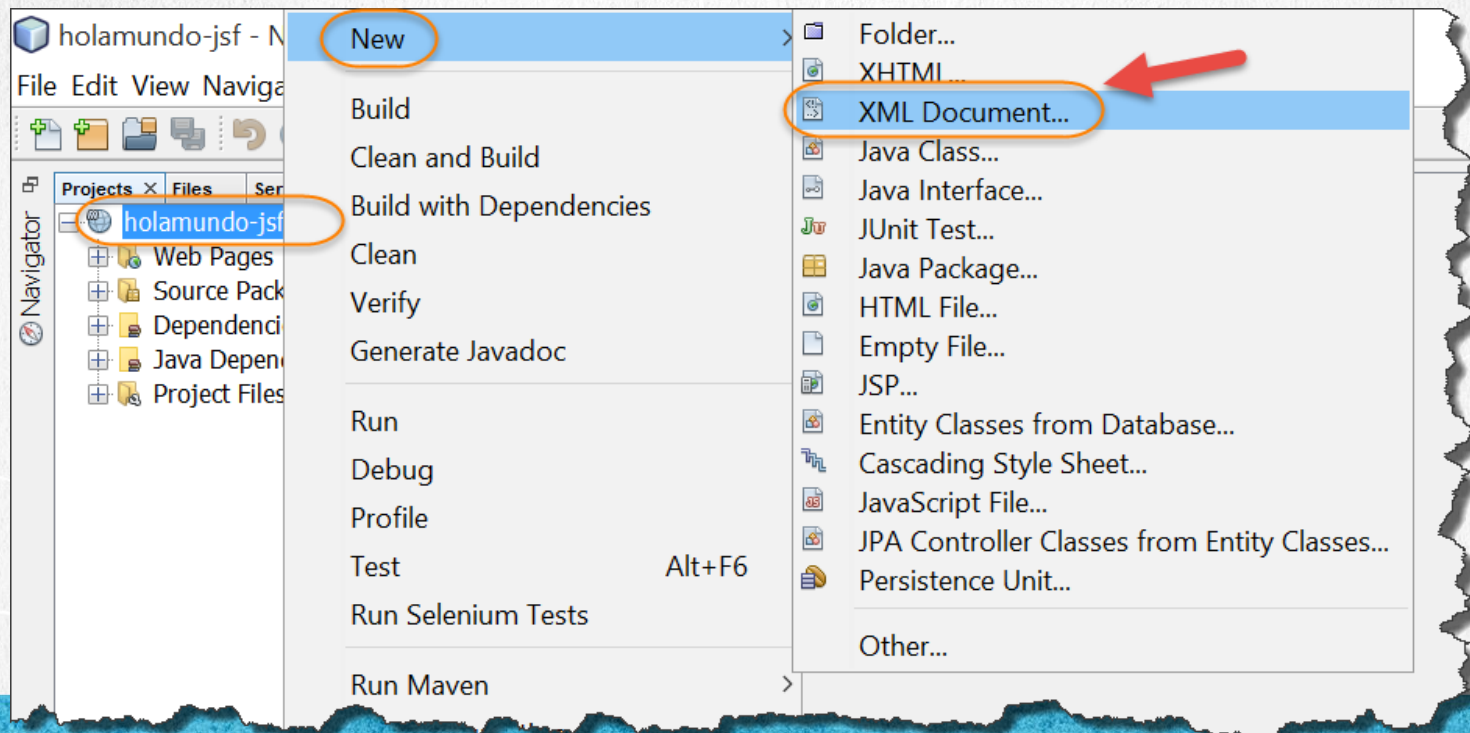
Para poder utilizar este API solo se requiere agregar las librerías de log4j las cuales ya han sido agregadas desde que se agregaron las librerías, y el archivo log4j2.xml en algún lugar que reconozca el classpath, por ejemplo en la carpeta src del proyecto.

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

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

10. CREAR UN ARCHIVO XML

- Creamos el archivo log4j2.xml:



CURSO DE JAVASERVER FACES

www.globalmentoring.com.mx

10. CREAR UN ARCHIVO XML

- Creamos el archivo log4j2.xml:

New XML Document

Steps

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

Name and Location

File Name:

Project:

Folder:

Created File:

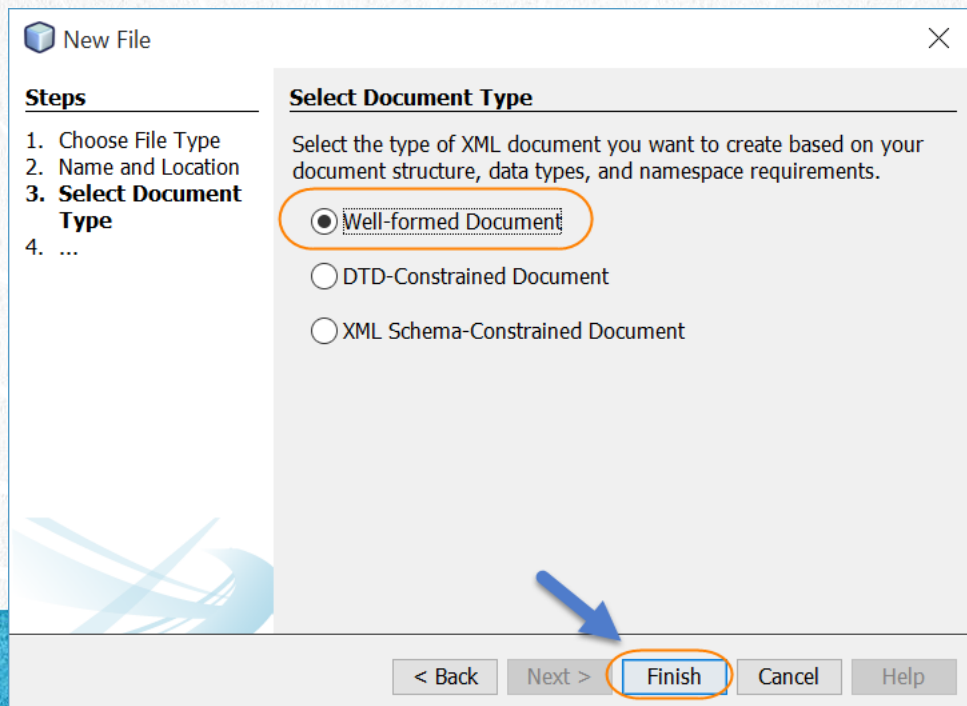
< Back **Next >** Finish Cancel Help

CURSO DE JAVASERVER FACES

www.globalmentoring.com.mx

10. CREAR UN ARCHIVO XML

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



PASO 11. MODIFICAMOS EL CÓDIGO

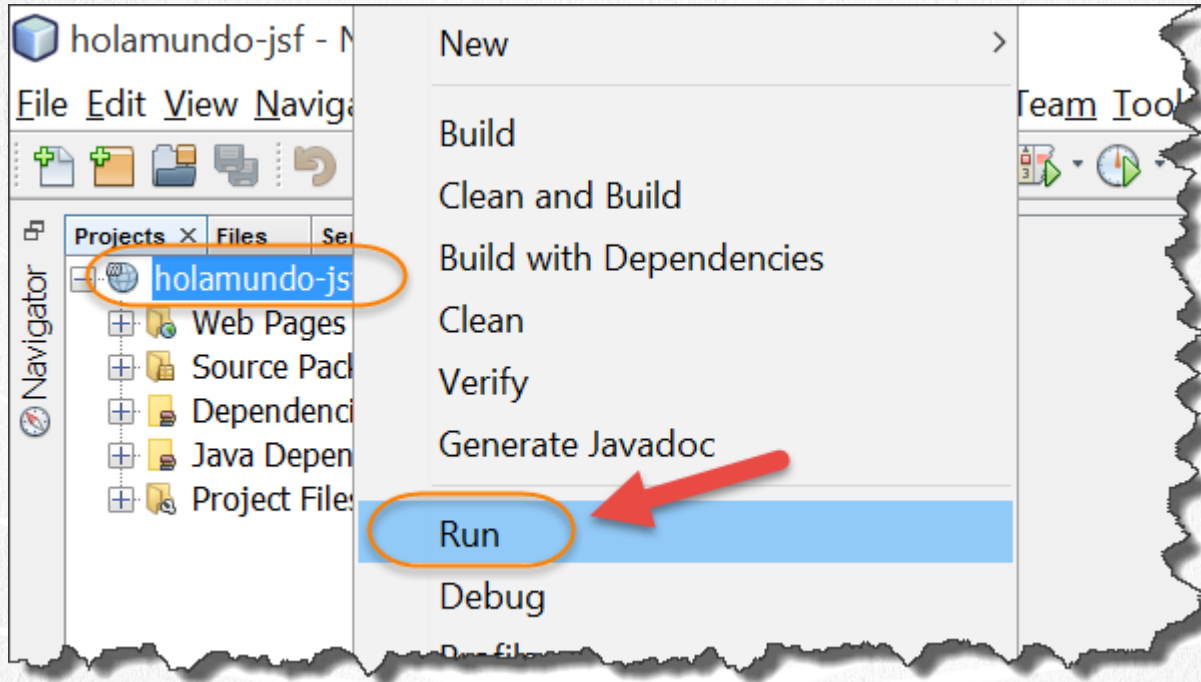
Archivo log4j2.xml:

Dar click para ir al código

```
<?xml version="1.0" encoding="UTF-8"?>
<Configuration status="INFO">
  <Appenders>
    <Console name="Console" target="SYSTEM_OUT">
      <PatternLayout pattern="%d{HH:mm:ss} [%t] %-5level %logger{36} - %msg%n" />
    </Console>
  </Appenders>
  <Loggers>
    <Root level="info">
      <AppenderRef ref="Console" />
    </Root>
  </Loggers>
</Configuration>
```

PASO 12. EJECUCIÓN DEL PROYECTO

Ejecutamos el proyecto:

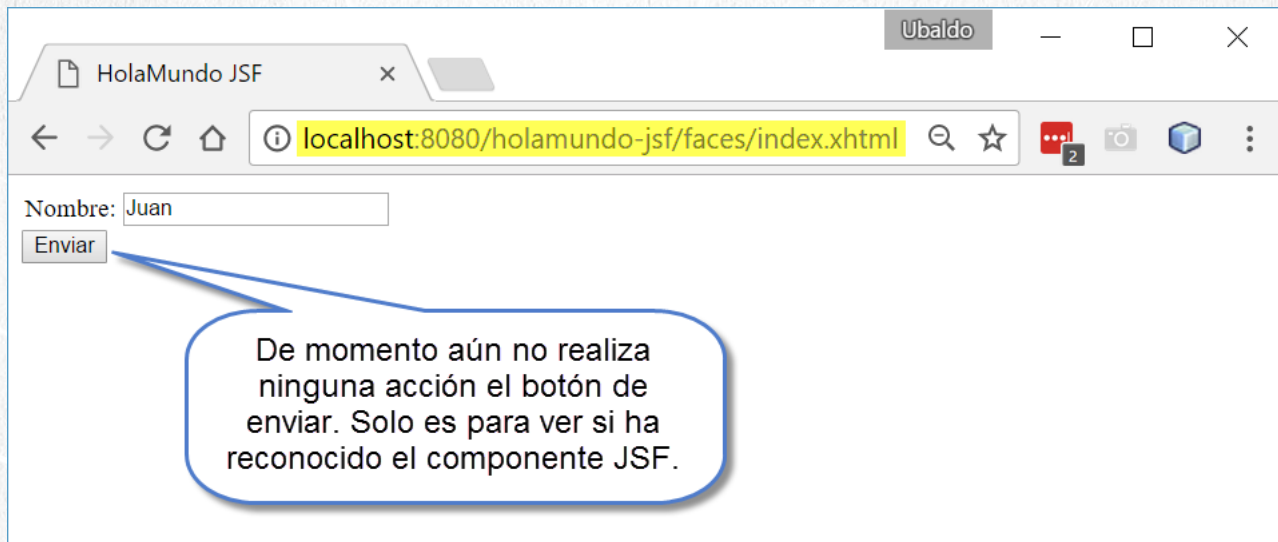


CURSO DE JAVASERVER FACES

www.globalmentoring.com.mx

PASO 12. EJECUCIÓN DEL PROYECTO

Ejecutamos el proyecto obteniendo el siguiente resultado:



CONCLUSIÓN DEL EJERCICIO

Con este ejercicio pusimos en práctica el HolaMundo con JSF. Utilizamos las últimas versiones de la tecnología JSF para la configuración del mismo.

Además vimos crear el archivo web.xml, el cual es uno de los archivos que utilizaremos para realizar varias de las configuraciones de JSF. En particular indicamos que reconozca el Servlet de JSF, y también indicamos la página de inicio desde este archivo de configuración.

Finalmente ejecutamos nuestro ejercicio pudiendo observar que los componentes agregados a la página de index.xhtml se muestran correctamente.



Experiencia y Conocimiento para tu vida

CURSO DE JAVASERVER FACES

www.globalmentoring.com.mx

CURSO ONLINE

JAVASERVER FACES (JSF)

Por: Ing. Ubaldo Acosta



Experiencia y Conocimiento para tu vida



CURSO DE JAVASERVER FACES

www.globalmentoring.com.mx