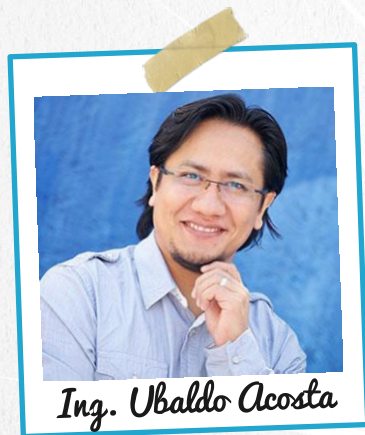


CURSO JAVA EE

CLIENTE SGA JEE WEB WEB SERVICES CON JAX-WS



Por el experto: Ing. Ubaldo Acosta



Experiencia y Conocimiento para tu vida



CURSO JAVA EE

www.globalmentoring.com.mx

OBJETIVO DEL EJERCICIO

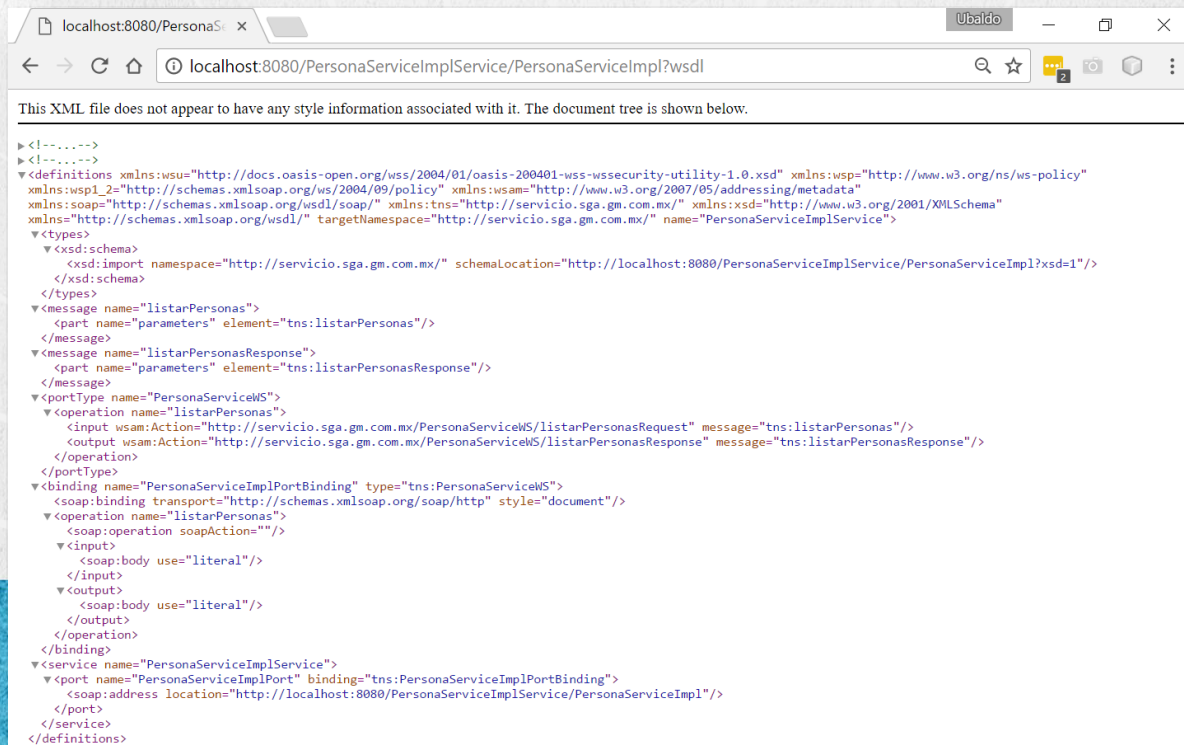
El objetivo del ejercicio crear un cliente del proyecto SgaJeeWeb Web Service de listado de personas creado en el ejercicio anterior. El resultado se muestra a continuación:

```
ClienteSgaWS - NetBeans IDE
File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help
<default confi...
Output
GlassFish Server x Run (TestPersonaServiceWS) x
cd C:\Cursos\JavaEE\Leccion11\ClienteSgaWS; "JAVA_HOME=C:\\Program Files\\Java\\jdk1.8.0_...
Running NetBeans Compile On Save execution. Phase execution is skipped and output director
Scanning for projects...
-----
Building ClienteSgaWS 1.0
-----
--- exec-maven-plugin:1.2.1:exec (default-cli) @ ClienteSgaWS ---
Ejecutando Servicio Listar Personas WS
Persona: Oscar Milan
Persona: Karla Lara
Persona: Maria Esparza
Fin Servicio Listar Personas WS
-----
BUILD SUCCESSFUL
```


PARTIMOS DEL EJERCICIO ANTERIOR

Vamos a partir del ejercicio anterior. Ya debe estar desplegado en Glassfish y funcionando. Debemos poder visualizar el siguiente url:

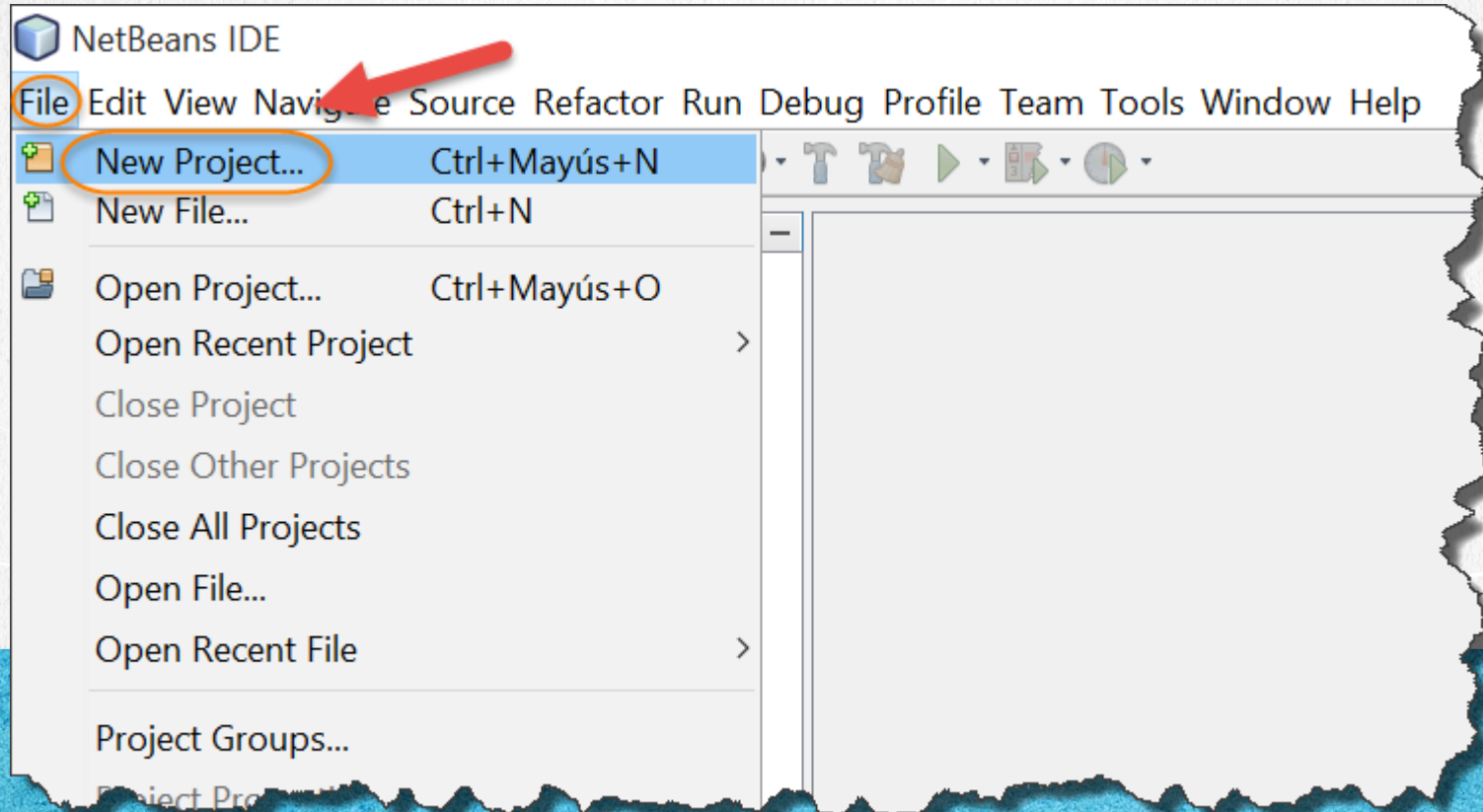
<http://localhost:8080/PersonaServiceImplService/PersonaServiceImpl?wsdl>



```
<?xml version='1.0' encoding='UTF-8'?>
<!DOCTYPE wsdl [
  <?xml schemaLocation='http://localhost:8080/PersonaServiceImplService/PersonaServiceImpl.xsd' type='text/xml'?>
</!DOCTYPE>
<definitions xmlns:wsu="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-utility-1.0.xsd" xmlns:wsp="http://www.w3.org/ns/ws-policy"
  xmlns:wsp1_2="http://schemas.xmlsoap.org/ws/2004/09/policy" xmlns:wsam="http://www.w3.org/2007/05/addressing/metadata"
  xmlns:soap="http://schemas.xmlsoap.org/soap/" xmlns:tns="http://servicio.sga.gm.com.mx/" xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns="http://schemas.xmlsoap.org/wsdl/" targetNamespace="http://servicio.sga.gm.com.mx/" name="PersonaServiceImplService">
  <types>
    <xsd:schema>
      <xsd:import namespace="http://servicio.sga.gm.com.mx/" schemaLocation="http://localhost:8080/PersonaServiceImplService/PersonaServiceImpl.xsd=1"/>
    </xsd:schema>
  </types>
  <message name="listarPersonas">
    <part name="parameters" element="tns:listarPersonas"/>
  </message>
  <message name="listarPersonasResponse">
    <part name="parameters" element="tns:listarPersonasResponse"/>
  </message>
  <portType name="PersonaServiceWS">
    <operation name="listarPersonas">
      <input wsam:Action="http://servicio.sga.gm.com.mx/PersonaServiceWS/listarPersonasRequest" message="tns:listarPersonas"/>
      <output wsam:Action="http://servicio.sga.gm.com.mx/PersonaServiceWS/listarPersonasResponse" message="tns:listarPersonasResponse"/>
    </operation>
  </portType>
  <binding name="PersonaServiceImplPortBinding" type="tns:PersonaServiceWS">
    <soap:binding transports="http://schemas.xmlsoap.org/soap/http" style="document"/>
    <operation name="listarPersonas">
      <soap:operation soapAction="">
        <input>
          <soap:body use="literal"/>
        </input>
        <output>
          <soap:body use="literal"/>
        </output>
      </operation>
    </binding>
  <service name="PersonaServiceImplService">
    <port name="PersonaServiceImplPort" binding="tns:PersonaServiceImplPortBinding">
      <soap:address location="http://localhost:8080/PersonaServiceImplService/PersonaServiceImpl"/>
    </port>
  </service>
</definitions>
```

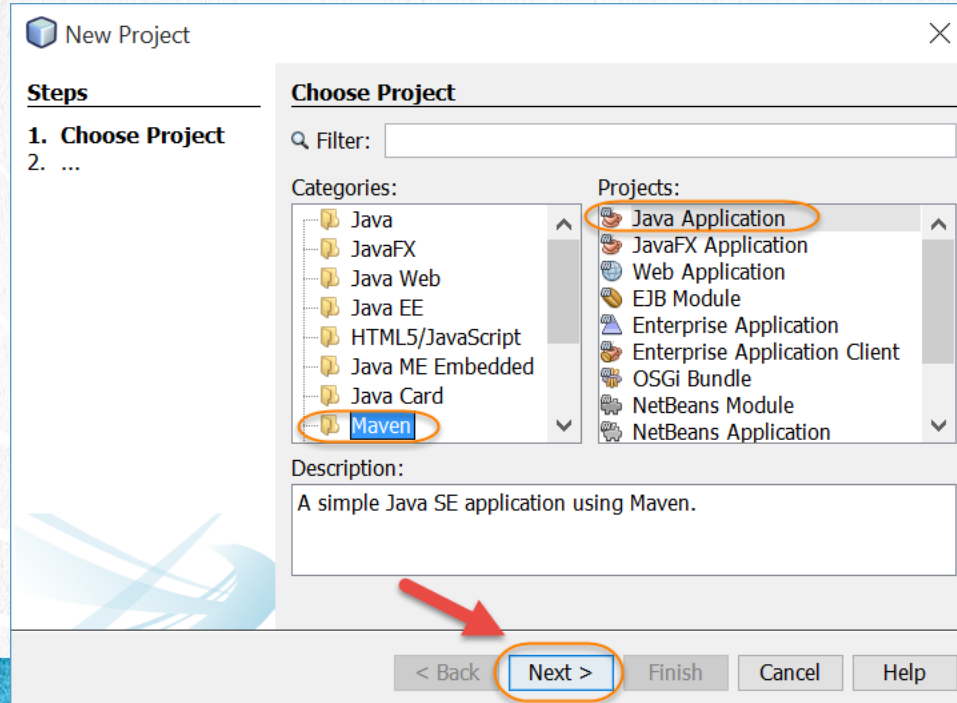
PASO 1. CREACIÓN DEL PROYECTO

Creamos el proyecto ClienteSgaWS:



PASO 1. CREACIÓN DEL PROYECTO

Creamos el proyecto ClienteSgaWS como un proyecto de maven:

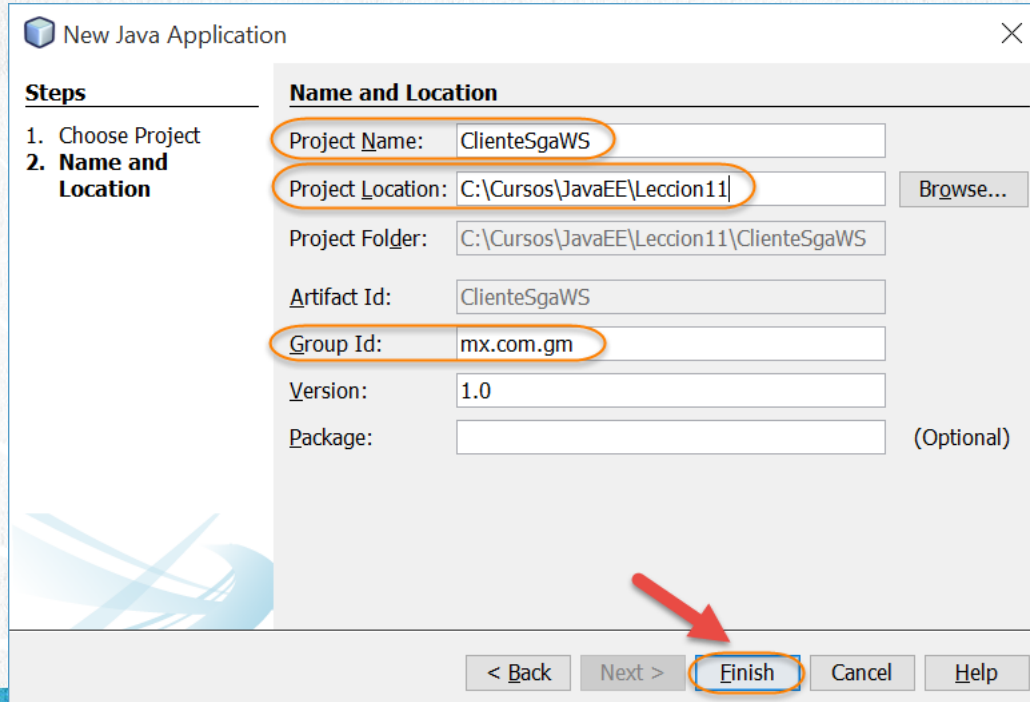


CURSO JAVA EE

www.globalmentoring.com.mx

PASO 1. CREACIÓN DEL PROYECTO

Creamos el proyecto ClienteSgaWS como un proyecto de maven:



New Java Application

Steps

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

Name and Location

Project Name: ClienteSgaWS

Project Location: C:\Cursos\JavaEE\Leccion11 Browse...

Project Folder: C:\Cursos\JavaEE\Leccion11\ClienteSgaWS

Artifact Id: ClienteSgaWS

Group Id: mx.com.gm

Version: 1.0

Package: (Optional)

< Back Next > **Finish** Cancel Help

CURSO JAVA EE

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>ClienteSgaWS</artifactId>
  <version>1.0</version>
  <packaging>jar</packaging>
  <properties>
    <project.build.sourceEncoding>UTF-8</project.build.sourceEncoding>
    <maven.compiler.source>1.8</maven.compiler.source>
    <maven.compiler.target>1.8</maven.compiler.target>
  </properties>
  <dependencies>
    <dependency>
      <groupId>javax</groupId>
      <artifactId>javaee-api</artifactId>
      <version>7.0</version>
      <scope>provided</scope>
    </dependency>
  </dependencies>
</project>
```

PASO 2. MODIFICAMOS EL CÓDIGO

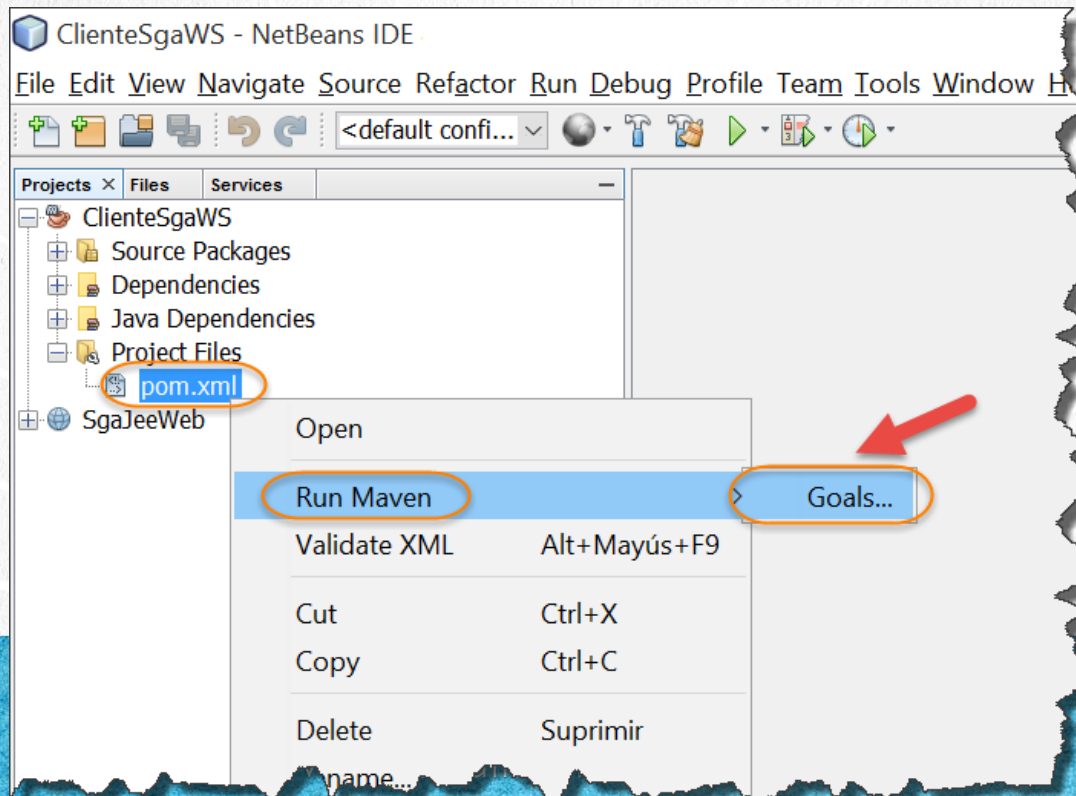
Archivo pom.xml:

Dar click para ir al código

```
<build>
  <plugins>
    <plugin>
      <groupId>org.codehaus.mojo</groupId>
      <artifactId>jaxws-maven-plugin</artifactId>
      <version>2.4.1</version>
      <configuration>
        <vmArgs>
          <vmArg>-Djavax.xml.accessExternalSchema=all</vmArg>
        </vmArgs>
      </configuration>
      <executions>
        <execution>
          <goals>
            <goal>wsimport</goal>
          </goals>
          <configuration>
            <wsdlUrls>
              <wsdlUrl>http://localhost:8080/PersonaServiceImplService/PersonaServiceImpl?wsdl</wsdlUrl>
            </wsdlUrls>
            <packageName>clientesga.ws</packageName>
            <sourceDestDir>${basedir}/src/main/java</sourceDestDir>
          </configuration>
        </execution>
      </executions>
    </plugin>
  </plugins>
</build>
</project>
```

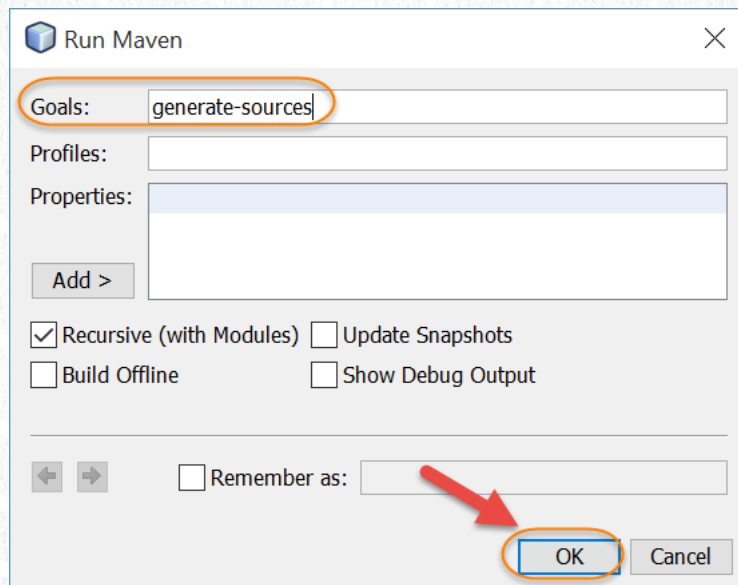

PASO 3. GENERAMOS EL CODIGO DEL CLIENTE WS

Ejecutamos el cliente para generar el código del Web Service por medio del comando **wsimport** agregado al archivo pom.xml:



PASO 3. GENERAMOS EL CODIGO DEL CLIENTE WS

Ejecutamos el cliente para generar el código del Web Service. Escribimos generate-sources en el campo de Goals para generar el código del cliente:

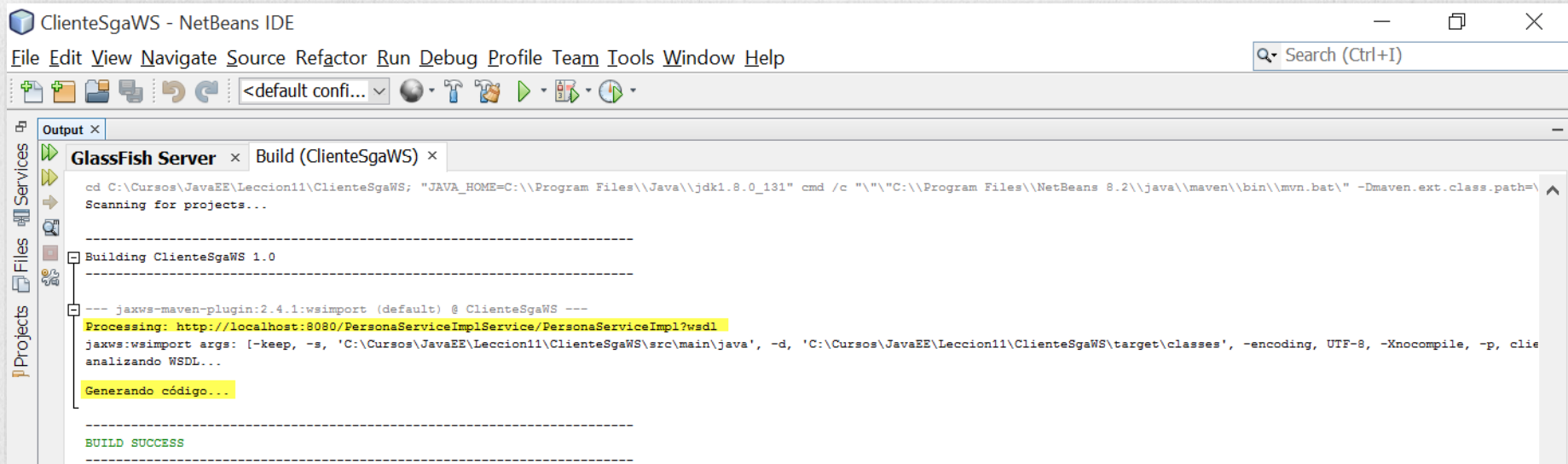


CURSO JAVA EE

www.globalmentoring.com.mx

PASO 3. GENERAMOS EL CODIGO DEL CLIENTE WS

Debemos ver una salida similar a esta:



The screenshot shows the NetBeans IDE interface for a project named 'ClienteSgaWS'. The 'Output' window is open, displaying the build log for 'GlassFish Server' and 'Build (ClienteSgaWS)'. The log shows the following steps:

```
cd C:\Cursos\JavaEE\Leccion11\ClienteSgaWS; "JAVA_HOME=C:\\Program Files\\Java\\jdk1.8.0_131" cmd /c "%C:\\Program Files\\NetBeans 8.2\\java\\maven\\bin\\mvn.bat" -Dmaven.ext.class-path=\\
Scanning for projects...

-----
Building ClienteSgaWS 1.0
-----

--- jaxws-maven-plugin:2.4.1:wsimport (default) @ ClienteSgaWS ---
Processing: http://localhost:8080/PersonaServiceImplService/PersonaServiceImpl?wsdl
jaxws:wsimport args: [-keep, -s, 'C:\\Cursos\\JavaEE\\Leccion11\\ClienteSgaWS\\src\\main\\java', -d, 'C:\\Cursos\\JavaEE\\Leccion11\\ClienteSgaWS\\target\\classes', -encoding, UTF-8, -Xnocompile, -p, clie
analizando WSDL...

Generando código...

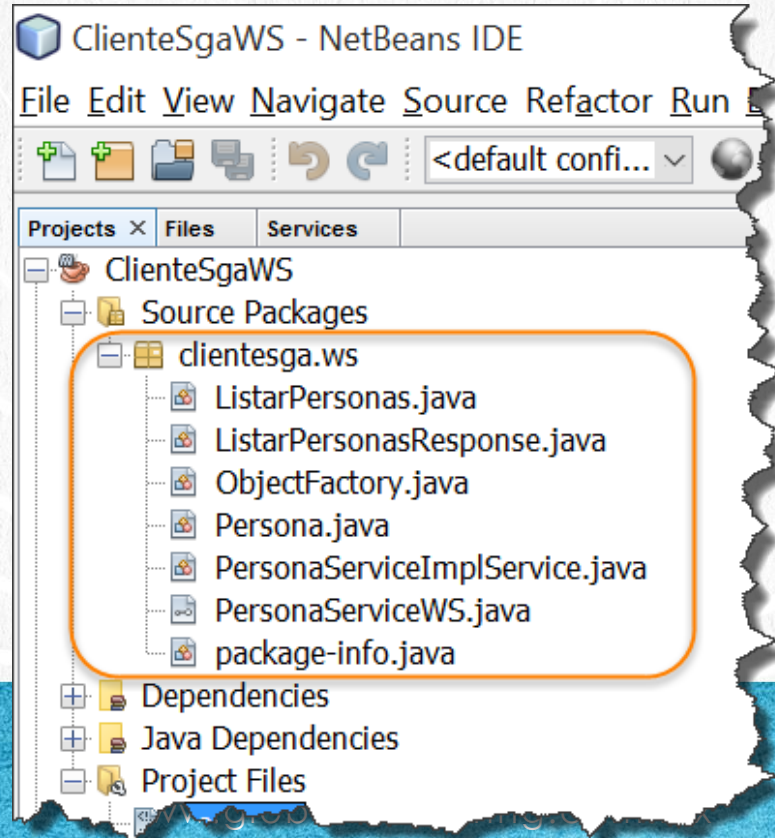
BUILD SUCCESS
```

CURSO JAVA EE

www.globalmentoring.com.mx

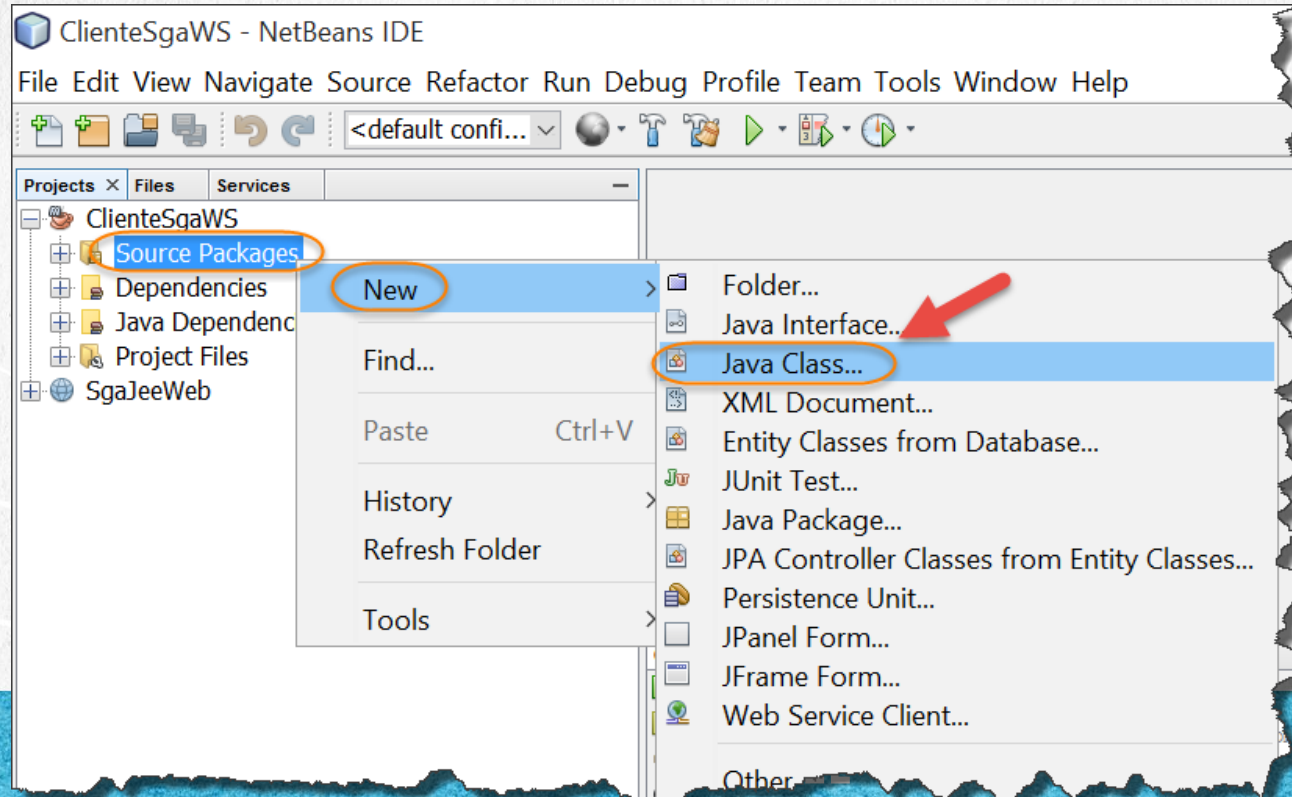
PASO 3. GENERAMOS EL CODIGO DEL CLIENTE WS

Debemos ver el código generado en el proyecto:



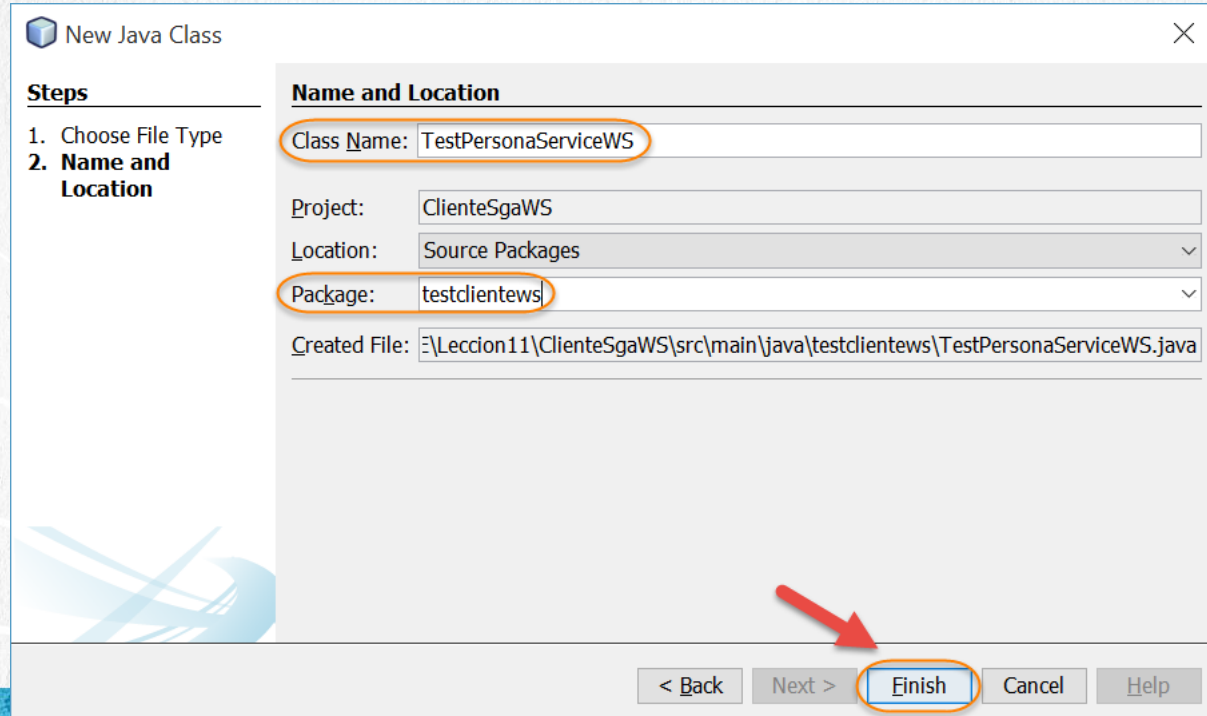
PASO 4. CREACIÓN CLASE JAVA

Creamos la clase TestPersonaServiceWS.java:



PASO 4. CREACIÓN CLASE JAVA

Creamos la clase TestPersonaServiceWS.java:



New Java Class

Steps

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

Name and Location

Class Name: TestPersonaServiceWS

Project: ClienteSgaWS

Location: Source Packages

Package: testclientews

Created File: E:\Leccion11\ClienteSgaWS\src\main\java\testclientews\TestPersonaServiceWS.java

< Back Next > **Finish** Cancel Help

CURSO JAVA EE

www.globalmentoring.com.mx

PASO 5. MODIFICAMOS EL CÓDIGO

Archivo TestPersonaServiceWS.java:

Dar click para ir al código

```
package testclientews;

import java.util.List;

import clientesga.ws.Persona;
import clientesga.ws.PersonaServiceImplService;
import clientesga.ws.PersonaServiceWS;

public class TestPersonaServiceWS {

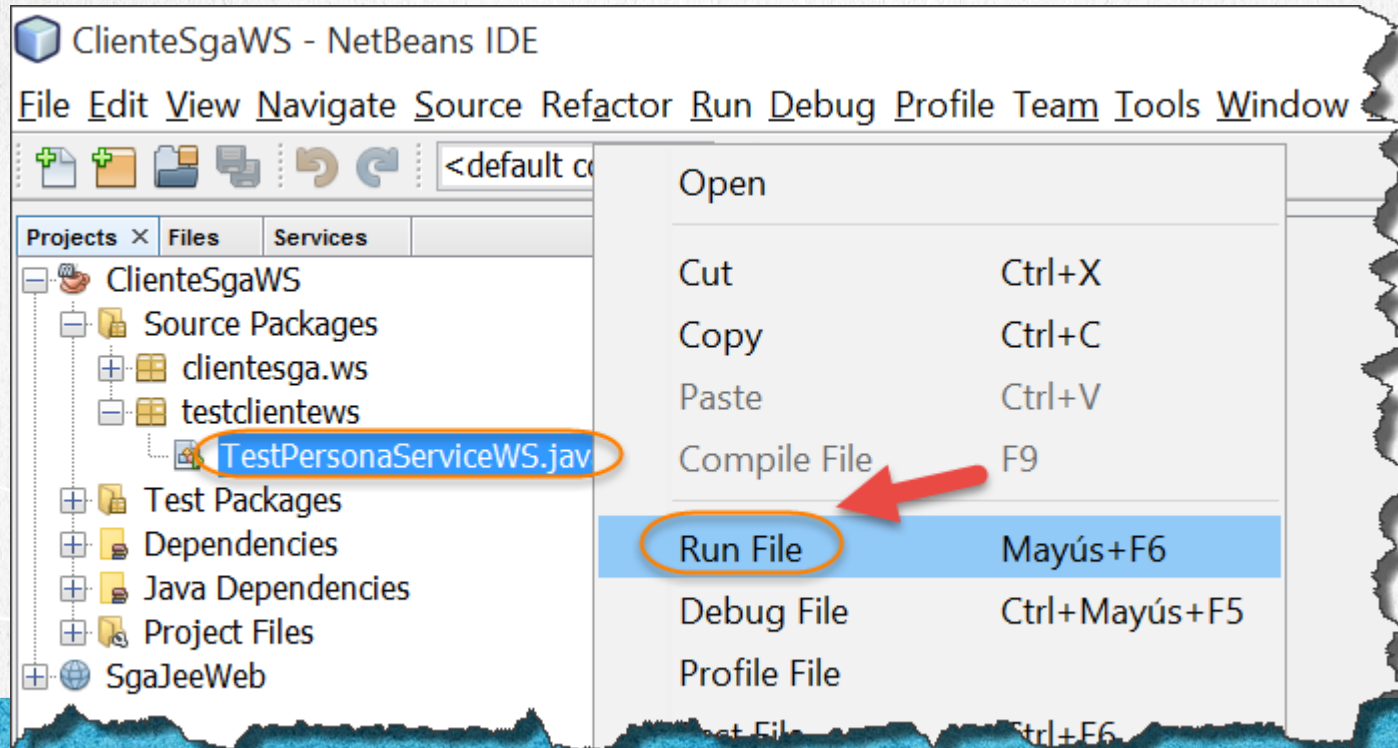
    public static void main(String[] args) {
        PersonaServiceWS personaService = new PersonaServiceImplService().getPersonaServiceImplPort();
        System.out.println("Ejecutando Servicio Listar Personas WS");
        List<Persona> personas = personaService.listarPersonas();
        for (Persona persona : personas) {
            System.out.println("Persona: " + persona.getNombre() + " " + persona.getApellidoPaterno());
        }
        System.out.println("Fin Servicio Listar Personas WS");
    }
}
```

CURSO JAVA EE

www.globalmentoring.com.mx

PASO 6. EJECUTAMOS LA APLICACIÓN

Ejecutamos la clase:

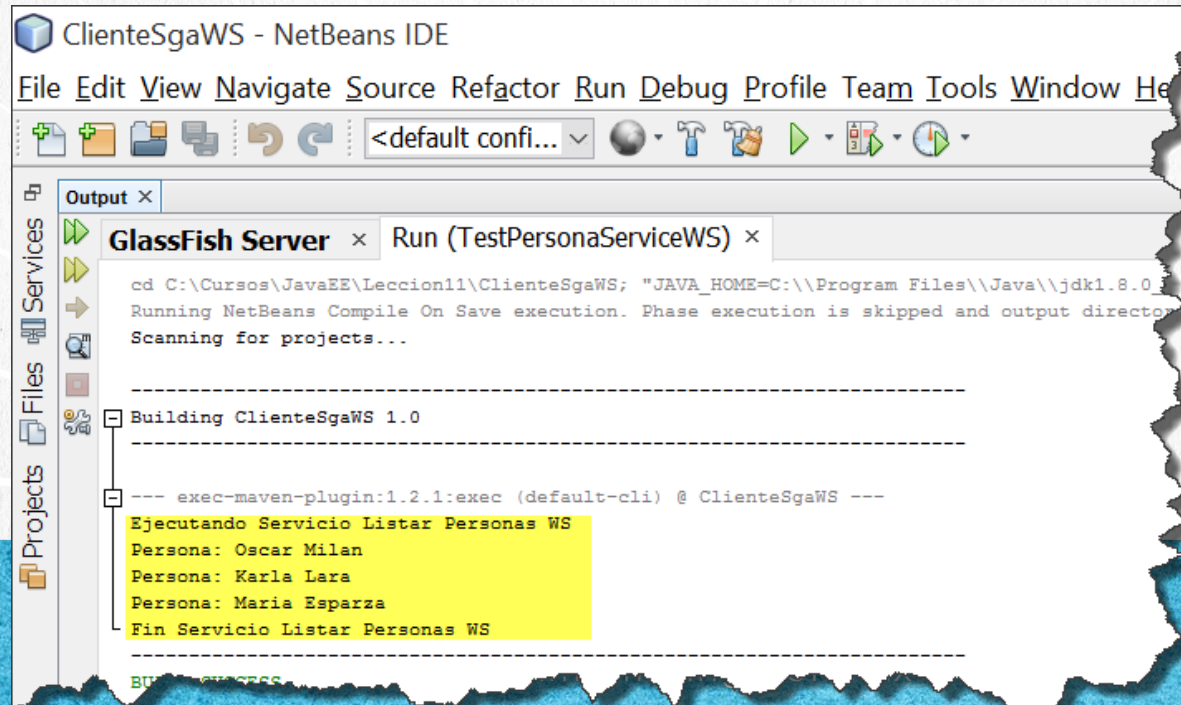


CURSO JAVA EE

www.globalmentoring.com.mx

PASO 6. EJECUTAMOS LA APLICACIÓN

Vemos el resultado de hacer la consulta al servicio Web. Como hemos comentado ya debe estar desplegado el Servicio Web del ejercicio anterior, ya que este cliente consume el servicio web de listarPersonas, y se recibe la respuesta de la llamada al Servicio Web solicitado:



The screenshot shows the NetBeans IDE interface for a project named 'ClienteSgaWS'. The 'Output' window is open, displaying the execution log of the 'Run (TestPersonaServiceWS)' task. The log shows the compilation of the project and the execution of a Maven plugin. The output of the plugin is highlighted in yellow, showing the results of a web service call to 'listarPersonas WS'.

```
cd C:\Cursos\JavaEE\Leccion11\ClienteSgaWS; "JAVA_HOME=C:\\Program Files\\Java\\jdk1.8.0_...
Running NetBeans Compile On Save execution. Phase execution is skipped and output director...
Scanning for projects...

-----
Building ClienteSgaWS 1.0
-----

--- exec-maven-plugin:1.2.1:exec (default-cli) @ ClienteSgaWS ---
Ejecutando Servicio Listar Personas WS
Persona: Oscar Milan
Persona: Karla Lara
Persona: Maria Esparza
Fin Servicio Listar Personas WS

BUILD SUCCESSFUL
```


CONCLUSIÓN DEL EJERCICIO

- ✓ Con este ejercicio pudimos observar cómo exponer los métodos del EJB's de PersonaService utilizando Web Services con el API de JAX-WS.
- ✓ Observamos cómo hacer una prueba del Web Service una vez desplegado sobre el servidor GlassFish.
- ✓ Además, revisamos cómo crear las clases Java del Cliente del Web Service a partir del documento WSDL, todo esto con ayuda del comando ***wsimport***, el cual es parte del JDK de Java.
- ✓ Con esto hemos visto el proceso completo de cómo crear un SOAP Web Service y el cliente respectivo.
- ✓ Además, vimos cómo excluir algunos atributos a enviar utilizando JAXB en nuestra clase de dominio Persona.
- ✓ Por último y lo más importante, es que utilizamos una arquitectura de 3 capas y vimos cómo exponer la lógica de negocio como un servicio Web a cualquier cliente interesado en la información del sistema SGA.

CURSO ONLINE

JAVA EMPRESARIAL JAVA EE

Por: Ing. Ubaldo Acosta



Experiencia y Conocimiento para tu vida



CURSO JAVA EE

www.globalmentoring.com.mx