

Tipos de Web Services

Las especificaciones JAX-WS y JAX-RS son los estándares para crear servicios web en Java EE.

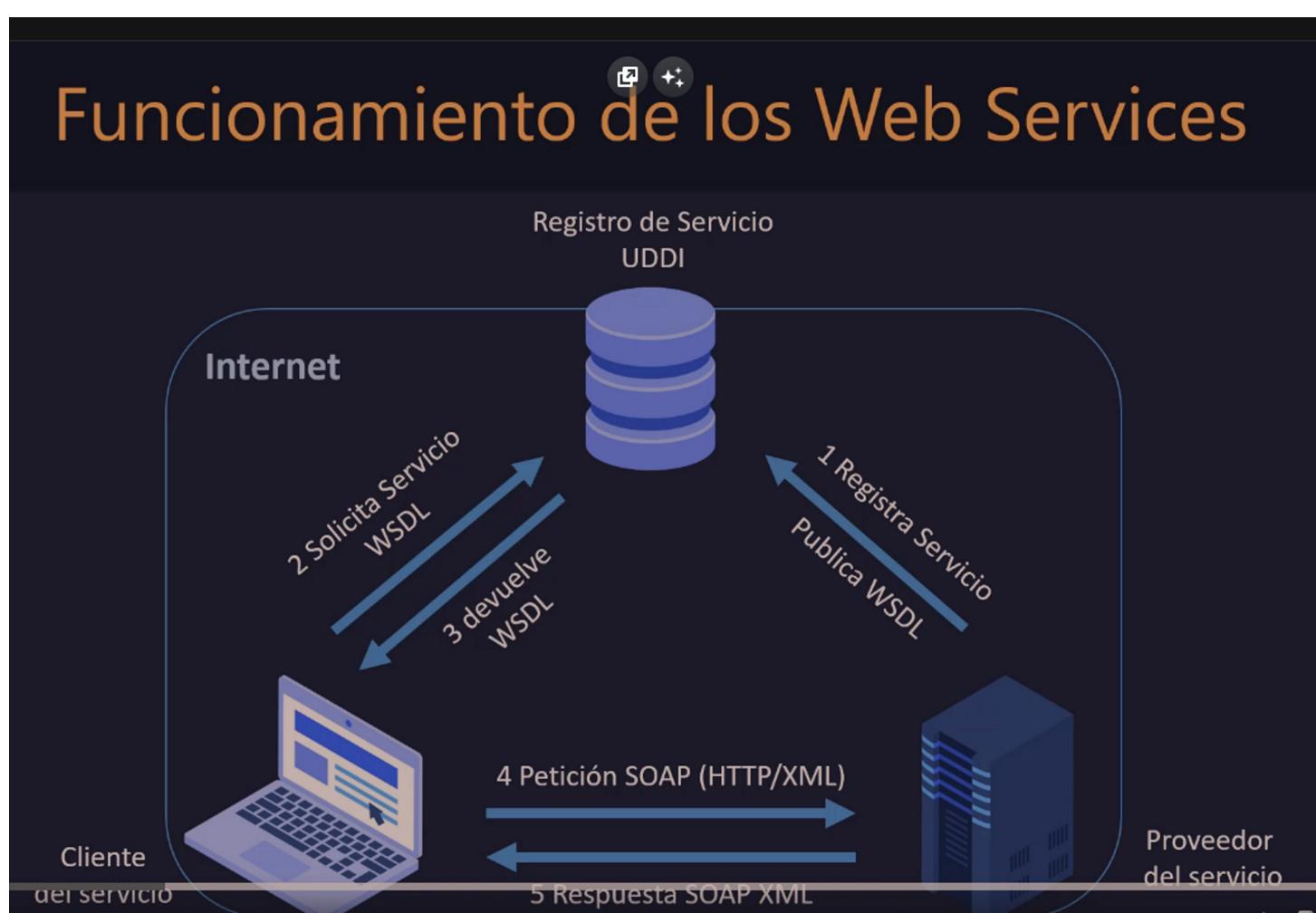
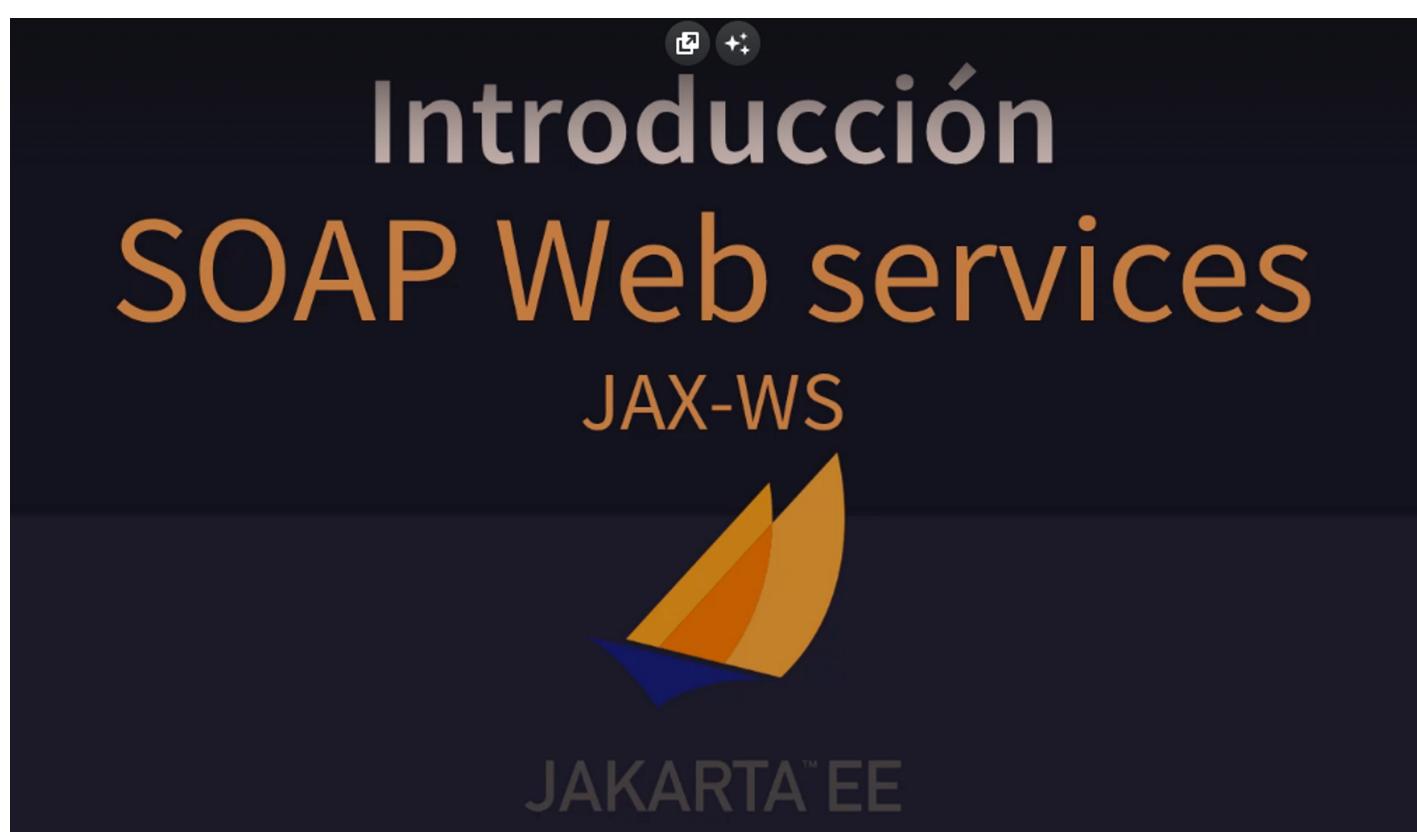
- JAX-WS (Java API for XML Web Services) es una API estandarizada para crear y consumir servicios web basados en SOAP (Simple Object Access Protocol).
- JAX-RS (Java API for RESTful Web Services), una API mucho más nueva y simple basada en REST (Representational State Transfer).

EN SOAP:

WSDL, Web Services Description Language

ARCHIVO QUE DESCRIBE LA ESTRUCTURA COMPLETA DE TODO EL SERVICIO WEB. LOS TIPOS DE DATOS DEL MENSAJE Q SE ESTÁ ENVIANDO EN XML.

CLASE 574



UDDI : ES UN REPOSITORIO DONDE SE ALMACENAN LOS ARCHIVOS DE SERVICIOS WEB WSDL. EL ARCHIVO Q TIENE TODA LA INFORMACIÓN DE SERVICIO WEB Y Q DICE CÓMO ENCONTRAR EL SERVICIO WEB.

Componentes JAX-WS

Descubrimiento UDDI

Descripción WSDL

Mensaje SOAP XML

Protocolo de comunicación HTTP

DESCUBRIMIENTO UDDI : REPOSITORIO Q ALMACENA LAS DESCRIPCIONES DE LOS SERVICIOS WEB PARA Q LOS CLIENTES PUEDAN BUSCAR UN SERVICIO WEB QUE SOLICITAN.

DESCRIPCIÓN WSDL : LENGUAJE PARA LA DESCRIPCIÓN DEL SERVICIO WEB , ESTA ESCRITO EN XML.

MENSAJE SOAP XML : SON LOS DATOS, LOS PARÁMETROS DE ENTRADA Y SALIDA.

HTTP : EL TRANSPORTE QUE ES EL PROTOCOLO HTTP.

Estructura del WSDL

```
<definitions ...>
  <types>
    <xsd:schema> ... </xsd:schema>
  </types>

  <message name=""> ... </message>

  <portType name="">
    <operation name="">
      <input .../>
      <output .../>
      <fault .../>
    </operation>
  </portType>

  <binding name="">
    <soap:binding transport="http://schemas.xmlsoap.org/soap/http" ... />
  </binding>

  <service name="">
    <port name="" ...> ... </port>
  </service>
</definitions>
```

-----clase 575 -----

Implementando y publicando Web Services JAX-WS

PARA CREAR PROYECTOS Q VENGAN ...

```
<packaging>war</packaging>

<properties>
  <maven.compiler.source>17</maven.compiler.source>
  <maven.compiler.target>17</maven.compiler.target>
  <project.build.sourceEncoding>UTF-8</project.build.sourceEncoding>
  <project.reporting.outputEncoding>UTF-8</project.reporting.outputEncoding>
</properties>

<dependencies>
  <dependency>
    <groupId>jakarta.platform</groupId>
    <artifactId>jakarta.jakartaee-api</artifactId>
    <version>9.1.0</version>
    <scope>provided</scope>
  </dependency>
</dependencies>

<build>
  <finalName>${project.artifactId}</finalName>
```

```

<plugins>
    <plugin>
        <artifactId>maven-compiler-plugin</artifactId>
        <version>3.8.1</version>
    </plugin>
    <plugin>
        <artifactId>maven-war-plugin</artifactId>
        <version>3.2.3</version>
        <configuration>
            <failOnMissingWebXml>false</failOnMissingWebXml>
        </configuration>
    </plugin>
    <plugin>
        <groupId>org.wildfly.plugins</groupId>
        <artifactId>wildfly-maven-plugin</artifactId>
        <version>2.1.0.Final</version>
    </plugin>
</plugins>
</build>

```

LEVANTA WILDFLY:

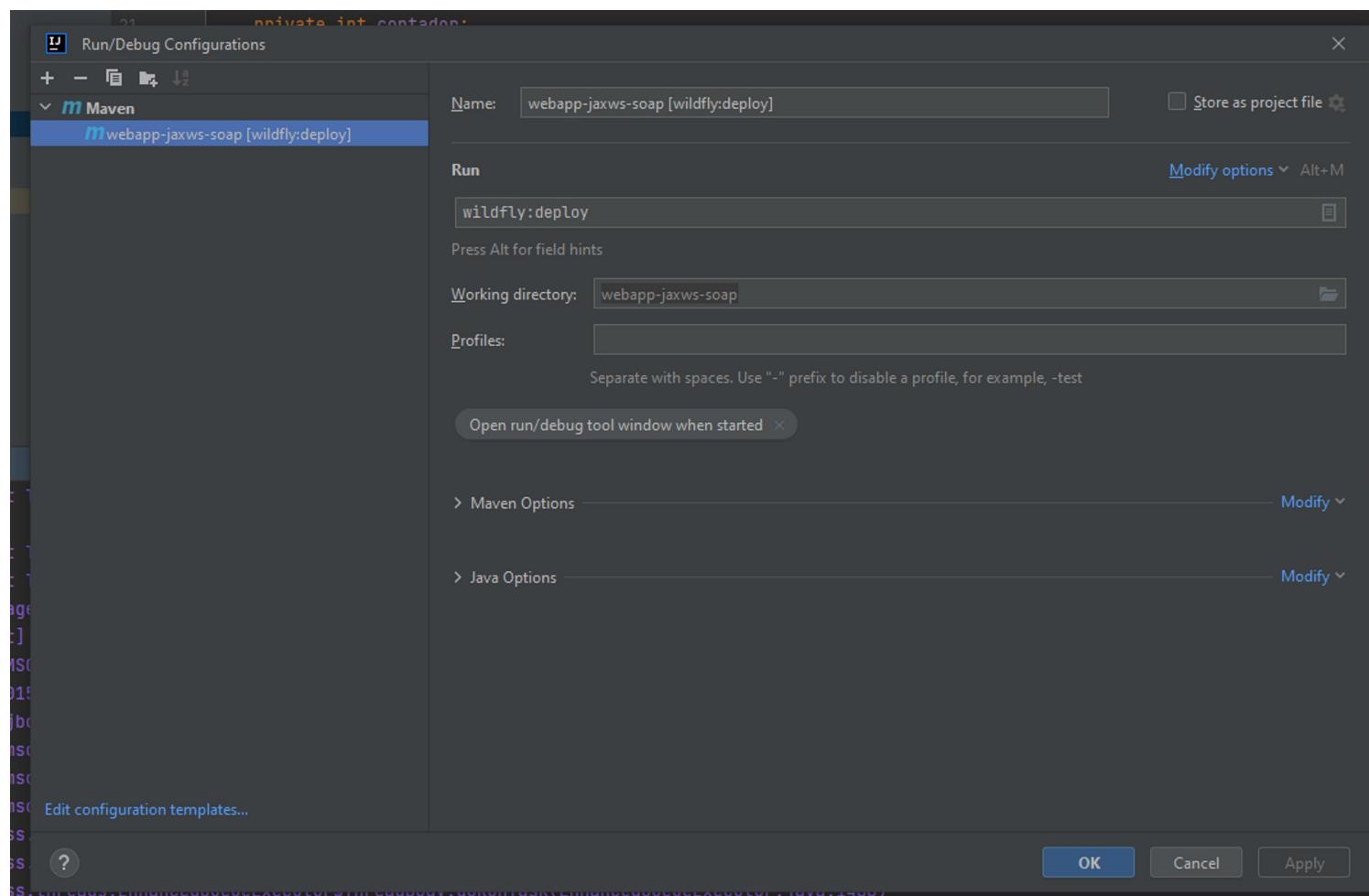
```

PS C:\developjava\wildfly-preview-26.1.3.Final\bin> cd C:\developjava\wildfly-preview-26.1.3.Final\bin
PS C:\developjava\wildfly-preview-26.1.3.Final\bin> ./standalone.bat

```

axel
12345

DESPLIEGUE EN WILDFLY



Main Attributes	
Name:	webapp-jaxws-soap.war
Runtime Name:	webapp-jaxws-soap.war
Context Root:	/webapp-jaxws-soap
Hash:	e11d0ad8bc1b5f8f7a88d0c808aeddd473fefce0
Enabled, Managed, Exploded:	✓✓✗
Status:	OK
Last enabled at:	5/11/23, 7:10 PM
Last disabled at:	n/a

msi | Management Console +

127.0.0.1:9990/console/index.html#runtime:path=standalone-server-column~standalone-host-msilss~webservices:endpoint~webapp-jaxws-soapwar-webapp-jaxws-soapwebappjaxwssoapmodelsservicesservicioWsimpl

WildFly

Homepage Deployments Configuration **Runtime** Patching Access Control

Server	Monitor	Endpoint (1)
msi	Status Management Operations Batch JBeret Datasources EJB IO JAX-RS JNDI JPA Log Files Security Transaction Web Undertow Webservices	Filter by: Endpoint name or deployment name webapp-jaxws-soap:w.j.s.m.services.ServicioWs... webapp-jaxws-soap.war

Main Attributes

Class:	webapp.jaxws.soap.models.services.ServicioWsimpl
Type:	JAXWS_JSE
Context:	webapp-jaxws-soap
WSDL URL:	http://localhost:8080/webapp-jaxws-soap/ServicioWsimpl?wsdl

Processing Time

Request / Response

HAZ CLICK EN WSDL URL ...

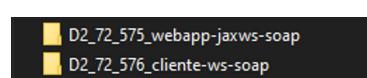
This XML file does not appear to have any style information associated with it. The document tree is shown below.

```

<wsdl:definitions xmlns:xs="http://www.w3.org/2001/XMLSchema" xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/" xmlns:tns="http://services.models.soap.jaxws.webapp/" xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/">
  <wsdl:types>
    <xsd:schema xmlns:xs="http://www.w3.org/2001/XMLSchema" xmlns:tns="http://services.models.soap.jaxws.webapp/" elementFormDefault="unqualified" targetNamespace="http://services.models.soap.jaxws.webapp/" version="1.0">
      <xsd:element name="crear" type="tns:crear"/>
      <xsd:element name="crearResponse" type="tns:crearResponse"/>
      <xsd:element name="listar" type="tns:listar"/>
      <xsd:element name="listarResponse" type="tns:listarResponse"/>
      <xsd:element name="saludar" type="tns:saludar"/>
      <xsd:element name="saludarResponse" type="tns:saludarResponse"/>
    </xsd:schema>
  </wsdl:types>
  <wsdl:types>
    <xsd:complexType name="saludar">
      <xsd:sequence>
        <xsd:element minOccurs="0" name="arg0" type="xsd:string"/>
      </xsd:sequence>
    </xsd:complexType>
    <xsd:complexType name="saludarResponse">
      <xsd:sequence>
        <xsd:element minOccurs="0" name="return" type="xsd:string"/>
      </xsd:sequence>
    </xsd:complexType>
    <xsd:complexType name="crear">
      <xsd:sequence>
        <xsd:element minOccurs="0" name="arg0" type="tns:curso"/>
      </xsd:sequence>
    </xsd:complexType>
    <xsd:complexType name="curso">
      <xsd:sequence>
        <xsd:element minOccurs="0" name="nombre" type="xsd:string"/>
      </xsd:sequence>
    </xsd:complexType>
    <xsd:complexType name="crearResponse">
      <xsd:sequence>
        <xsd:element minOccurs="0" name="return" type="tns:curso"/>
      </xsd:sequence>
    </xsd:complexType>
  </wsdl:types>
  <wsdl:message name="crear">
    <wsdl:part element="tns:crear" name="parameters"/>
  </wsdl:message>
  <wsdl:message name="saludarResponse">
    <wsdl:part element="tns:saludarResponse" name="parameters"/>
  </wsdl:message>
  <wsdl:message name="crearResponse">
    <wsdl:part element="tns:crearResponse" name="parameters"/>
  </wsdl:message>
  <wsdl:message name="listarResponse">
    <wsdl:part element="tns:listarResponse" name="parameters"/>
  </wsdl:message>
</wsdl:definitions>

```

--clase 576--



SE CREA CLIENTE PARA CONSUMIR EL SERVICIO WEB SOAP :

POM:

```

<?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>cliente.ws.soap</groupId>
<artifactId>cliente-ws-soap</artifactId>
<version>1.0-SNAPSHOT</version>
<packaging>jar</packaging>

<properties>
  <maven.compiler.source>17</maven.compiler.source>

```

```

<maven.compiler.target>17</maven.compiler.target>
<project.build.sourceEncoding>UTF-8</project.build.sourceEncoding>
</properties>

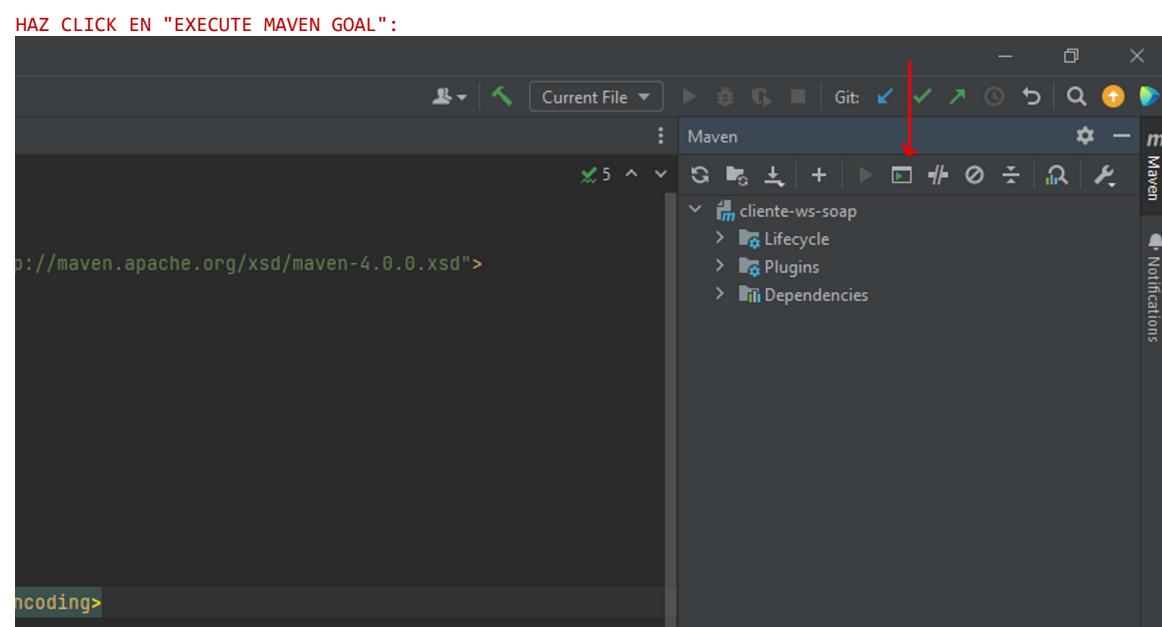
<dependencies>
    <dependency>
        <groupId>com.sun.xml.ws</groupId>
        <artifactId>jaxws-rt</artifactId>
        <version>4.0.1</version>
    </dependency>
</dependencies>

<build>
    <plugins>
        <plugin>
            <groupId>com.sun.xml.ws</groupId>
            <artifactId>jaxws-maven-plugin</artifactId>
            <version>3.0.2</version>
            <executions>
                <!--SE CONFIGURA EL PLUGIN PARA Q GENERE AUTOMATICAMENTE AL CLIENTE, USANDO EL WS IMPORT.-->
                <execution>
                    <goals>
                        <goal>wsimport</goal>
                    </goals>
                    <configuration>
                        <wsdlUrls>
                            <!--SE PONE LA RUTA DEL WSDL XML-->
                            <wsdlUrl>http://localhost:8080/webapp-jaxws-soap/ServicioWsImpl?wsdl</wsdlUrl>
                        </wsdlUrls>
                    </configuration>
                </execution>
            </executions>
        </plugin>
    </plugins>
</build>

</project>

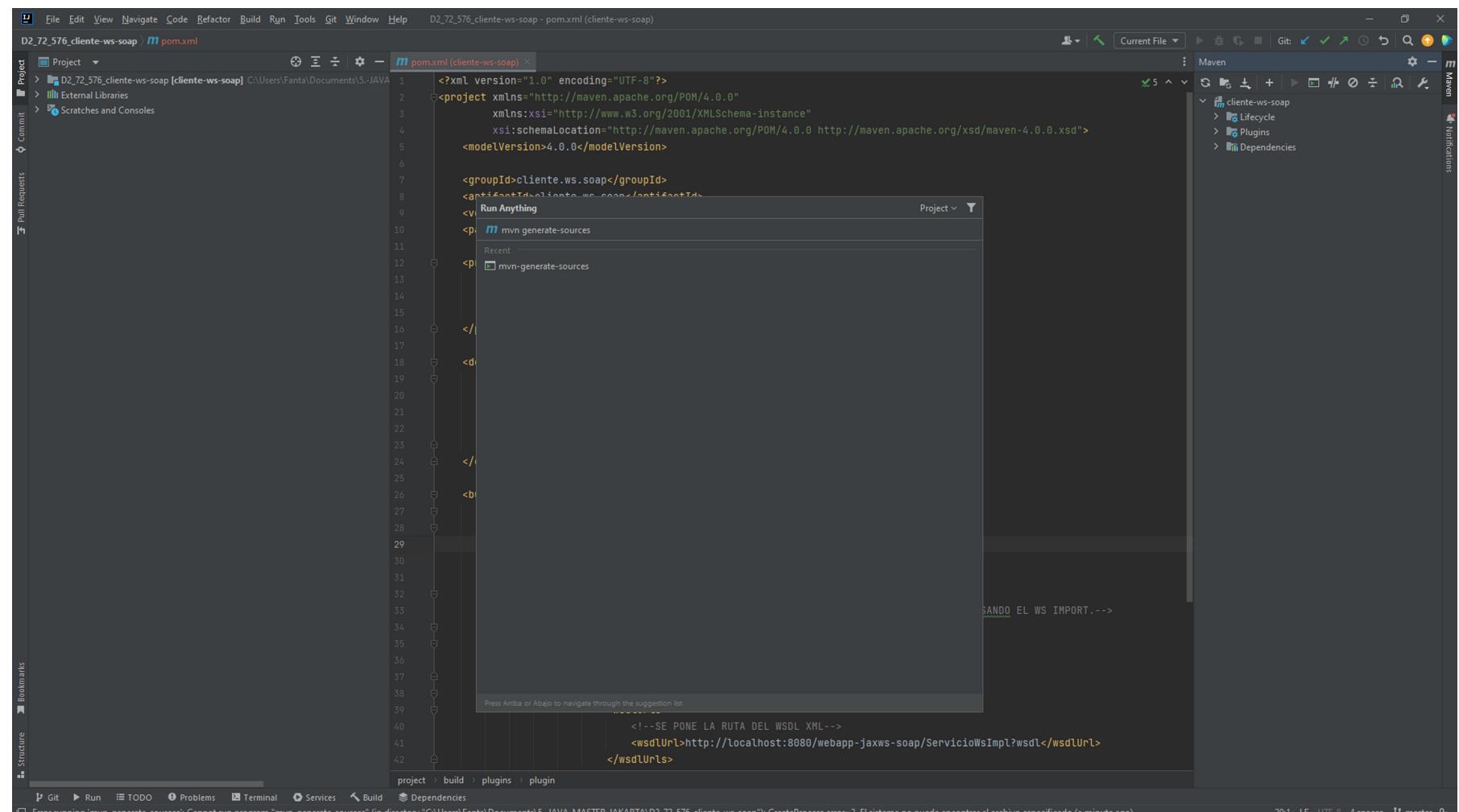
```

LUEGO SE GENERA EL CLIENTE:



ESCRIBE :

`mvn generate-sources`



DALE ENTER :

The screenshot shows the IntelliJ IDEA interface with the Maven tool window open. The terminal window displays the command-line output of a Maven build:

```

D:\2_72_576_cliente-ws-soap\cliente-ws-soap pom.xml (cliente-ws-soap)
[INFO] Scanning for projects...
[INFO] Building cliente-ws-soap 1.0-SNAPSHOT
[INFO]   [ jar ]
[INFO] --- jaxws-maven-plugin:3.0.2:wsimport (default) @ cliente-ws-soap ---
[INFO] Processing: http://localhost:8080/webapp-jaxws-soap/ServicioWsImpl?wsdl
[INFO] jaxws:wsimport args: [-keep, -s, 'C:\Users\Fanta\Documents\5.-JAVA_MASTER_JAKARTA\02_72_576_cliente-ws-soap\target\generated-sources\wsimport', -d, 'C:\Users\Fanta\Documents\5.-JAVA_MASTER_JAKARTA\02_72_576_cliente-ws-soap\target\classes', -encoding, UTF-8, -Xnocompile, "http://localhost:8080/webapp-jaxws-soap/ServicioWsImpl?wsdl"]
analizando WSDL...

Generando código...
[INFO] BUILD SUCCESS
[INFO] Total time: 1.143 s
[INFO] Finished at: 2023-05-11T21:20:18-05:00
[INFO]

Process finished with exit code 0

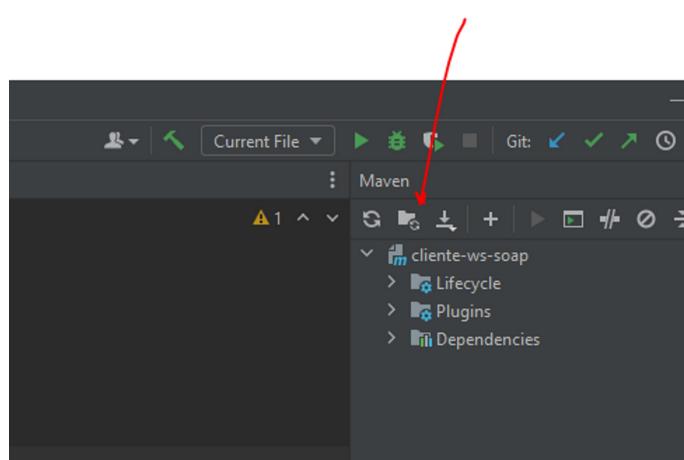
```

VE EL TARGET , Y REVISA SI CREO LOS RECURSOS PARA USAR EL SERVICIO SOAP:

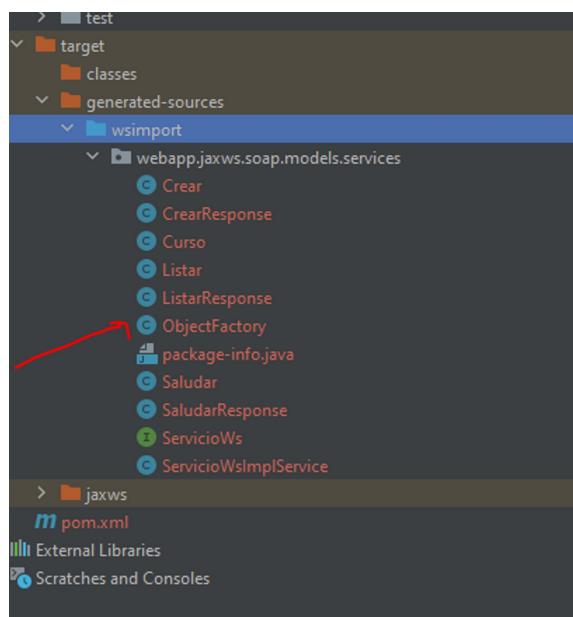
The screenshot shows the IntelliJ IDEA interface with the Project tool window open. The 'target/generated-sources/wsimport' directory is selected, revealing numerous generated Java files:

- jaxws
 - soap
 - models
 - services
 - Crear.java
 - CrearResponse.java
 - Curso.java
 - Listar.java
 - ListarResponse.java
 - ObjectFactory.java
 - package-info.java
 - Saludar.java
 - SaludarResponse.java
 - ServicioWs.java
 - ServicioWsImplService.java

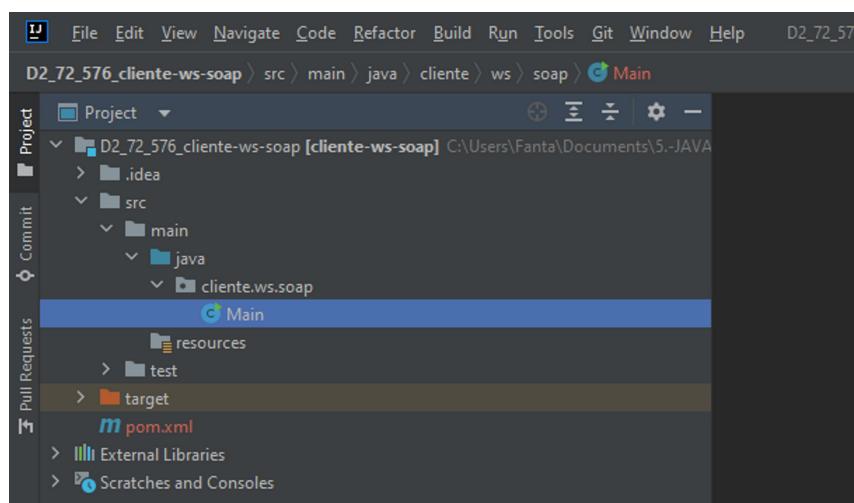
LUEGO DALE ACA :



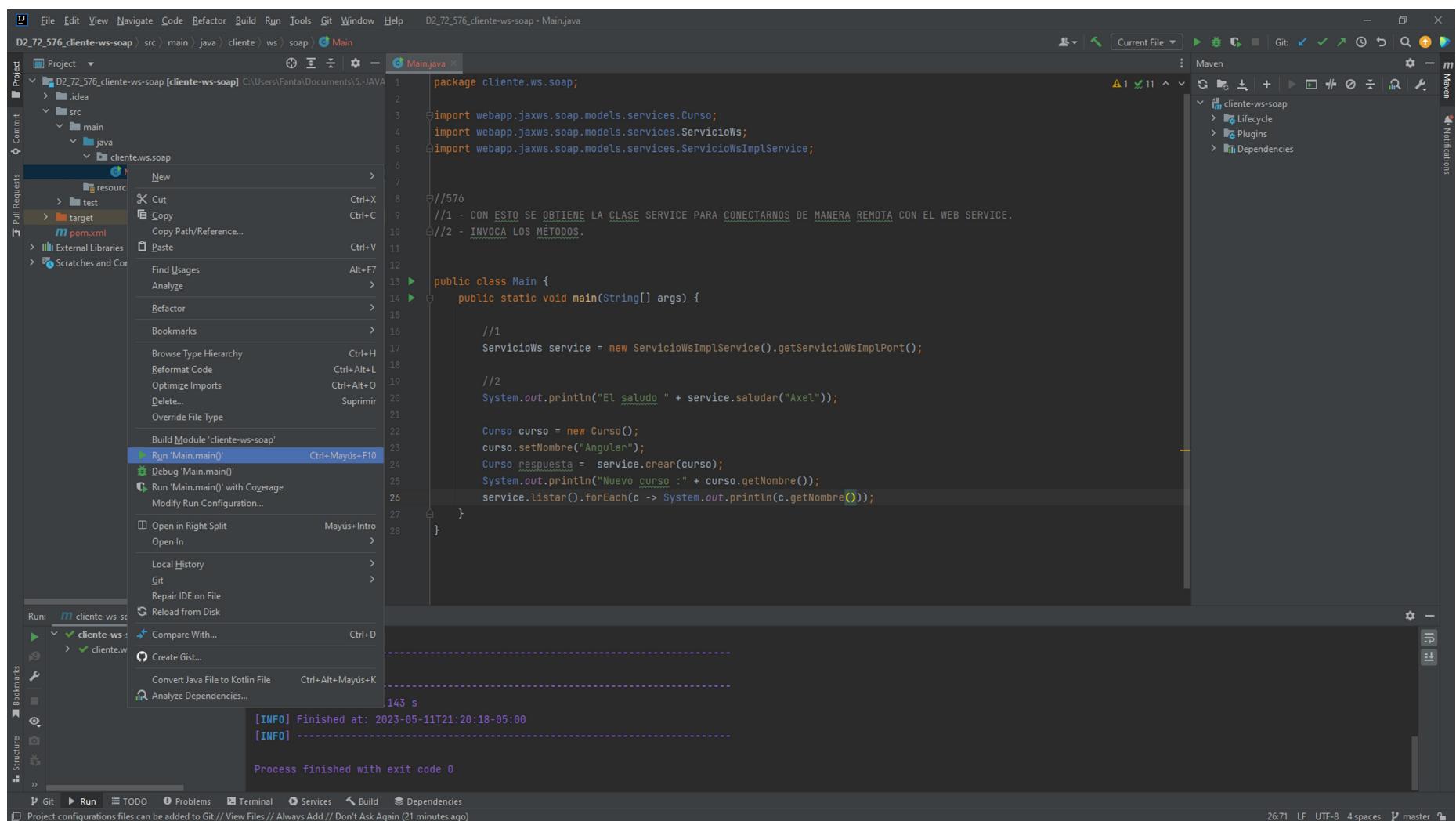
Y YA QUEDAN EN AZUL :



CREA TU CLASE MAIN :



DALE RUN



SE CONECTÓ CON EL SERVICIO WEB DE WILDFLY :

```

18
19 //2
20 System.out.println("-----");
21
22 Curso curso = new Curso();
23 curso.setNombre("curso");
24 curso.setDuracion(10);
25 curso.setPrecio(1000000);
26 curso.setEstado(true);
27 }

```

Run: Main

```

C:\developjava\jdk-17.0.2\bin\java.exe ...
El saludo Hola que tal Axel
Nuevo curso :Angular
Java
Spring
Jakarta ee

Process finished with exit code 0

```

TERMINAL DEL PROYECTO DEL SERVICIO WEB :

```

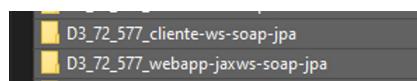
10 //1 //2
11 @WebService(endpointInterface = "webapp.jaxws.soap.models.services.ServicioWs")
12 public class ServicioWsImpl implements ServicioWs{
13
14     private int contador;
15
16     @Override
17     @WebMethod //3
18     public String saludar(String persona) {
19         System.out.println("imprimiendo dentro del servicio web con instancia : " + this);
20
21         extension.microprofile.opentracing.spi.sender.WildFlySender@33cd3ab0, closeEnqueueTimeout=1000), sampler=ConstSampler(decision=true, tags={sampler.type=const, sampler.param=true}), tags={hostname=MSI, jaeger.version=Java-1.6.0, ip=192.168.0.6}, zipkinSharedRpcSpan=false, expandExceptionLogs=false, useTraceId128Bit=false)
22 21:11:16,133 INFO [org.jipijapa] (MSC service thread 1-1) JIPIORMW53020253: Second level cache enabled for webapp-jsf3.war#ejemploJpa
23 21:11:16,134 INFO [org.apache.cxf.wsdl.service.factory.ReflectionServiceFactoryBean] (MSC service thread 1-4) Creating Service {http://services.models.soap.jaxws.webapp/}ServicioWsImplService from class webapp.jaxws.soap.models.services.ServicioWs
24 21:11:16,150 INFO [org.infinispan.CONTAINER] (ServerService Thread Pool -- 79) ISPN000556: Starting user marshaller 'org.wildfly.clustering.infinispan.spi.marshalling.InfinispanProtoStreamMarshaller'
25 21:11:16,150 INFO [org.infinispan.CONTAINER] (ServerService Thread Pool -- 78) ISPN000556: Starting user marshaller 'org.wildfly.clustering.infinispan.marshalling.jboss.JBossMarshaller'
26 21:11:16,493 INFO [org.jboss.weld.Version] (MSC service thread 1-1) WELD-000900: 4.0.3 (Final)
27 21:11:16,655 INFO [org.apache.cxf.endpoint.ServerImpl] (MSC service thread 1-4) Setting the server's publish address to be http://localhost:8080/webapp-jaxws-soap/ServicioWsImpl
28 21:11:18,805 INFO [org.jboss.ws.cxf.deployment] (MSC service thread 1-4) JBWS024074: WSDL published to: file:/C:/developjava/wildfly-preview-26.1.3.Final/standalone/data/wsdl/webapp-jaxws-soap.war/ServicioWsImplService.wsdl
29 21:11:16,807 INFO [org.jboss.as.jpa] (ServerService Thread Pool -- 79) WFLYJPA0010: Starting Persistence Unit (phase 1 of 2) Service 'webapp-jsf3.war#ejemploJpa'
30 21:11:16,815 INFO [org.infinispan.CONFIG] (MSC service thread 1-1) ISPN000152: Passivation configured without an eviction policy being selected. Only manually evicted entities will be passivated.
31 21:11:18,845 INFO [org.jboss.as.server] (Controller Boot Thread) WFLYSRV0010: Deployed "webapp-jsf3.war" (runtime-name : "webapp-jsf3.war")
32 21:11:18,845 INFO [org.jboss.as.server] (Controller Boot Thread) WFLYSRV0010: Deployed "com.mysql:mysql-connector-j-8.0.32.jar" (runtime-name : "com.mysql:mysql-connector-j-8.0.32.jar")
33 21:11:18,881 INFO [org.jboss.as.server] (Controller Boot Thread) WFLYSRV0212: Resuming server
34 21:11:18,884 INFO [org.jboss.as] (Controller Boot Thread) WFLYSRV0025: WildFly Preview 26.1.3.Final (WildFly Core 18.1.2.Final) started in 6734ms - Started 661 of 843 services (349 services are lazy, passive or on-demand) - Server configuration file in use: standalone.xml
35 21:11:18,886 INFO [org.jboss.as] (Controller Boot Thread) WFLYSRV0060: Http management interface listening on http://127.0.0.1:9990/management
36 21:11:18,887 INFO [org.jboss.as] (Controller Boot Thread) WFLYSRV0051: Admin console listening on http://127.0.0.1:9990
37 21:42:49,559 INFO [stdout] (default task-1) imprimiendo dentro del servicio web con instancia : webapp.jaxws.soap.models.services.ServicioWsImpl@30ded28c
38 21:42:49,560 INFO [stdout] (default task-1) valor del contador en el m\u00f3dulo saludar es: 1
39 21:42:49,578 INFO [stdout] (default task-1) Curso guardado con exito... Angular

```

EL WEB SERVICE ES STATELESS ... O SEA UNA INSTANCIA COMPARTIDA POR LOS CLIENTES.

-----clase 577 -----

577. Ejemplo JAX-WS EJB y JPA



EL PROCESO PARA ARMAR LOS PROYECTOS ES IGUAL AL ANTERIOR

- 1 LEVANTA WILDFLY
- 2 DESPLIEGA EL PROYECTO DE SERVICIO SOAP

SERVICIO SOAP DEPLEGADO EN WILDFLY

3 COPIA LA URL Y PEGALA EN EL POM DEL PROYECTO CLIENTE

```

<build>
    <plugins>
        <plugin>
            <groupId>com.sun.xml.ws</groupId>
            <artifactId>jaxws-maven-plugin</artifactId>
            <version>3.0.2</version>
            <executions>
                <!--SE CONFIGURA EL PLUGIN PARA Q GENERE AUTOMATICAMENTE AL CLIENTE, USANDO EL WS IMPORT.-->
                <execution>
                    <goals>
                        <goal>wsimport</goal>
                    </goals>
                    <configuration>
                        <wsdlUrls>
                            <!--SE PONE LA RUTA DEL WSDL XML-->
                            <wsdlUrl>http://localhost:8080/webapp-jaxws-soap-jpa/CursoServicioWsImpl?wsdl</wsdlUrl>
                        </wsdlUrls>
                    </configuration>
                </execution>
            </executions>
        </plugin>
    </plugins>

```

- 1 - CARGA EL TARGET
- 2 - REVISA EL TARGET
- 3 - HACE EL CÓDIGO
- 4 - RUN CON MAIN
- 5 - SE CONECTÓ CON EL SERVICIO SOAP

File Edit View Navigate Code Refactor Build Run Tools Git Window Help D3_72_577_cliente-ws-soap-jpa - Main.java

Main.java

```

1 package cliente.ws.soap;
2
3 import webapp.jaxws.soap.models.services.Curso;
4 import webapp.jaxws.soap.models.services.CursoServicioWs;
5 import webapp.jaxws.soap.models.services.CursoServiceImplService;
6
7 //577
8 //1 - CON ESTO SE OBTIENE LA CLASE SERVICE PARA CONECTARNOS DE MANERA REMOTA CON EL WEB SERVICE.
9 //2 - INVOCAR LOS MÉTODOS.
10
11 public class Main {
12
13     public static void main(String[] args) {
14
15         //1
16         CursoServicioWs service = new CursoServiceImplService().getCursoServicioWsImplPort();
17         Curso curso = new Curso();
18         curso.setNombre("React");
19         curso.setDescripcion("react js");
20         curso.setDuracion(500); //ES UN DOUBLE
21         curso.setInstructor("Andrés Guzmán");
22         Curso respuesta = service.guardar(curso);
23         System.out.println("Nuevo curso : id:" + curso.getId() + " desc:" + curso.getNombre());
24         service.listar().forEach(c -> System.out.println(c.getNombre()));
25     }
26 }

```

Project: D3_72_577_cliente-ws-soap-jpa [cliente-ws-soap-jpa] C:\Users\

- src
 - main
 - java
 - cliente.ws.soap
- resources
- test
- target
- classes
- generated-sources
- annotations
- wlexport
- webapp.jaxws.soap.models.services
 - Curso
 - CursoServicioWs
 - CursoServiceImplService
 - Guardar
 - GuardarResponse
 - Listar
 - ListarResponse
 - ObjectFactory
 - package-info.java
- jaxws
- pom.xml
- External Libraries
- Scratches and Consoles

Run: Main X

C:\developjava\jdk-17.0.2\bin\java.exe ...

Nuevo curso : id:null desc:React

java

javaee

angular

spring

React

Process finished with exit code 0

WORKBENCH

Tables	
alumnos	
categorias	
clientes	
clientes_detalles	
cursos	
cursoss	
direccion	
direcciones	
facturas	
productos	
tbl_alumnos_cursos	
tbl_clientes_direcciones	
usuarios	
Views	
Stored Procedures	
Functions	

Result Grid | Filter Rows: [] | Edit: [] | Export/Import: []

	id	nombre	descripcion	instructor	duracion
1	1	java	java	andres	100
2	2	javaee	jakarta	andres	50
3	3	angular	angular	andres	40
4	4	spring	spring boot	andres	40
5	React	react js	Andrés Guzmán	50	