



Servicios Web

EJERCICIOS

Cliente



```
Simbolo del sistema
C:\jdk1.6\bin>wsimport http://localhost:8080/circulo2?wsdl
C:\jdk1.6\bin>javac circulo2\ClienteCirculo2.java
C:\jdk1.6\bin>java circulo2.ClienteCirculo2
No se ha introducido el tipo de operacion
C:\jdk1.6\bin>java circulo2.ClienteCirculo2 5
opcion erronea
C:\jdk1.6\bin>java circulo2.ClienteCirculo2 1 3
28.274333882308138
C:\jdk1.6\bin>java circulo2.ClienteCirculo2 2 3
18.84955592153876
C:\jdk1.6\bin>_
```

Servicio



```
Símbolo del sistema - java círculo2.PuntoFinal2WS
C:\jdk1.6\bin>ws-gen -cp . círculo2.CirculoFunciones2
C:\jdk1.6\bin>java PuntoFinal2WS
Exception in thread "main" java.lang.NoClassDefFoundError: PuntoFinal2WS
C:\jdk1.6\bin>java círculo2.PuntoFinal2WS
El Servicio Web para el calculos del círculo
WSDL: http://localhost:8080/círculo2?wsdl
```

WSDL



- `<?xml version="1.0" encoding="UTF-8" ?>`
- `- <definitions xmlns="http://schemas.xmlsoap.org/wsdl/" xmlns:tns="http://circulo2/" xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/" targetNamespace="http://circulo2/" name="CirculoFunciones2Service">`
- `- <types>`
- `- <xsd:schema>`
- `<xsd:import schemaLocation="http://localhost:8080/circulo2?xsd=1" namespace="http://circulo2/" />`
- `</xsd:schema>`
- `</types>`
- `- <message name="getArea">`
- `<part element="tns:getArea" name="parameters" />`
- `</message>`
- `- <message name="getAreaResponse">`
- `<part element="tns:getAreaResponse" name="parameters" />`
- `</message>`
- `- <message name="getCircunferencia">`
- `<part element="tns:getCircunferencia" name="parameters" />`
- `</message>`
- `- <message name="getCircunferenciaResponse">`
- `<part element="tns:getCircunferenciaResponse" name="parameters" />`
- `</message>`



- `<portType name="CirculoFunciones2">`
- - `<operation name="getArea">`
- `<input message="tns:getArea" />`
- `<output message="tns:getAreaResponse" />`
- `</operation>`
- - `<operation name="getCircunferencia">`
- `<input message="tns:getCircunferencia" />`
- `<output message="tns:getCircunferenciaResponse" />`
- `</operation>`
- `</portType>`
- - `<binding name="CirculoFunciones2PortBinding" type="tns:CirculoFunciones2">`
- `<soap:binding style="document" transport="http://schemas.xmlsoap.org/soap/http" />`
- - `<operation name="getArea">`
- `<soap:operation soapAction="" />`
- - `<input>`
- `<soap:body use="literal" />`
- `</input>`
- + `<output>`
- `<soap:body use="literal" />`
- `</output>`
- `</operation>`

WSDL



- + <operation name="getCircunferencia">
- <soap:operation soapAction="" />
- - <input>
- <soap:body use="literal" />
- </input>
- - <output>
- <soap:body use="literal" />
- </output>
- </operation>
- </binding></binding>
- - <service name="CirculoFunciones2Service">
- - <port name="CirculoFunciones2Port"
- binding="tns:CirculoFunciones2PortBinding">
- <soap:address location="http://localhost:8080/circulo2" />
- </port>
- </service>
- </definitions>

ClienteCirculo2.java



```

▪ //hacemos una llamada a la clase de servicio generada automaticamente
▪ package circulo2;

▪
▪ public class ClienteCirculo2
▪ {
▪     //punto de entrada del cliente
▪
▪     public static void main (String[] args)
▪     {
▪         //declaramos la variable cadena encargada de recoger los argumentos
▪         int operacion;
▪         double resultado;
▪         String CadenaSalida=null;
▪         double radio;
▪         String datoCadena=null;
▪
▪         if (args.length>0)
▪         {
▪             //recuperamos la operacion a realizar
▪             operacion=Integer.parseInt(args[0].trim());
▪
▪             try
▪             {
▪                 //establecemos el servicio
▪                 CirculoFunciones2Service servicio= new CirculoFunciones2Service();
▪                 //obtenos el puerto del servicio
▪                 CirculoFunciones2 puerto=servicio.getCirculoFunciones2Port();

```

ClienteCirculo2.java



```
switch(operacion)
{
    case 1:
        //eliminamos los espacios en blanco que pudieramos tener
        datoCadena=args[1].trim();
        radio=Double.valueOf(datoCadena).doubleValue();
        resultado= puerto.getArea(radio);
        CadenaSalida=String.valueOf(resultado);
        break;

    case 2:
        //eliminamos los espacios en blanco que pudieramos tener
        datoCadena=args[1].trim();
        //convertimos la cadena en valor de tipo double
        radio=Double.valueOf(datoCadena).doubleValue();
        resultado= puerto.getCircunferencia(radio);
        CadenaSalida=String.valueOf(resultado);

        break;

    default:
        CadenaSalida="opcion erronea";
}
```

ClienteCirculo2.java



```

    }
    catch (Exception ex)
    {
        CadenaSalida=ex.toString();
    }
    finally
    {
        System.out.println(CadenaSalida);
    }
    }
else
{
    CadenaSalida="No se ha introducido el tipo de operacion";
    System.out.println(CadenaSalida);
}
}
}
```

PuntoFinal2.java



- package circulo2;
- import javax.xml.ws.Endpoint;

- public class PuntoFinal2WS
- {
- public static void main(String[] args) {
- String endpointAddress = "http://localhost:8080/circulo2";
- Endpoint.publish(endpointAddress, new CirculoFunciones2());
- System.out.println("El Servicio Web para el calculos del circulo");
- System.out.println("WSDL: " + endpointAddress + "?wsdl");
- }
- }

CirculoFunciones2.java



- package circulo2;
- import java.lang.Math;
- import javax.jws.WebService;
- @WebService
- public class CirculoFunciones2
- {
- public double getArea (double radio)
- {
- return Math.PI *(radio * radio);
- }
- public double getCircunferencia (double radio)
- {
- return Math.PI * 2 * radio;
- }
- }

Servicio de Almacén



```
Símbolo del sistema - java almacen.PuntoFinalAlmacenWS
C:\jdk1.6\bin>javac almacen\PuntoFinalAlmacenWS.java
C:\jdk1.6\bin>java almacen.PuntoFinalAlmacenWS
El Servicio Web para de almacen
WSDL: http://localhost:8080/almacen?wsdl
-
```

WSDL



- `<?xml version="1.0" encoding="UTF-8" ?>`
- `- <definitions xmlns="http://schemas.xmlsoap.org/wsdl/" xmlns:tns="http://almacen/" xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/" targetNamespace="http://almacen/" name="ServicioAlmacenService">`
- `- <types>`
- `- <xsd:schema>`
- `<xsd:import schemaLocation="http://localhost:8080/almacen?xsd=1" namespace="http://almacen/" />`
- `</xsd:schema>`
- `</types>`
- `message name="meterCamisas">`
- `<part element="tns:meterCamisas" name="parameters" />`
- `</message>`
- `- <message name="meterCamisasResponse">`
- `<part element="tns:meterCamisasResponse" name="parameters" />`
- `</message>`
- `- <message name="sacarCamisas">`
- `<part element="tns:sacarCamisas" name="parameters" />`
- `</message>`
- `- <message name="sacarCamisasResponse">`
- `<part element="tns:sacarCamisasResponse" name="parameters" />`
- `</message>`



- `message name="meterAbrigos">`
- `<part element="tns:meterAbrigos" name="parameters" />`
- `</message>`
- `- <message name="meterAbrigosResponse">`
- `<part element="tns:meterAbrigosResponse" name="parameters" />`
- `</message>`
- `+ <message name="sacarAbrigos">`
- `<part element="tns:sacarAbrigos" name="parameters" />`
- `</message>`
- `- <message name="sacarAbrigosResponse">`
- `<part element="tns:sacarAbrigosResponse" name="parameters" />`
- `</message>`
- `- <message name="meterChaquetas">`
- `<part element="tns:meterChaquetas" name="parameters" />`
- `</message>`
- `- <message name="meterChaquetasResponse">`
- `<part element="tns:meterChaquetasResponse" name="parameters" />`
- `</message>`
- `- <message name="sacarChaquetas">`
- `<part element="tns:sacarChaquetas" name="parameters" />`
- `</message>`
- `- <message name="sacarChaquetasResponse">`
- `<part element="tns:sacarChaquetasResponse" name="parameters" />`
- `</message>`

WSDL



```
<portType name="ServicioAlmacen" >
  - <operation name="meterCamisas">
    <input message="tns:meterCamisas" />
    <output message="tns:meterCamisasResponse" />
  </operation>
  - <operation name="sacarCamisas">
    <input message="tns:sacarCamisas" />
    <output message="tns:sacarCamisasResponse" />
  </operation>
  - <operation name="meterAbrigos">
    <input message="tns:meterAbrigos" />
    <output message="tns:meterAbrigosResponse" />
  </operation>
  - <operation name="sacarAbrigos">
    <input message="tns:sacarAbrigos" />
    <output message="tns:sacarAbrigosResponse" />
  </operation>
  - <operation name="meterChaquetas">
    <input message="tns:meterChaquetas" />
    <output message="tns:meterChaquetasResponse" />
  </operation>
  - <operation name="sacarChaquetas">
    <input message="tns:sacarChaquetas" />
    <output message="tns:sacarChaquetasResponse" />
  </operation>
</portType>
```

WSDL



- <binding name="**ServicioAlmacenPortBinding**" type="**tns:ServicioAlmacen**">
- <soap:binding style="**document**" transport="**http://schemas.xmlsoap.org/soap/http**" />
- - <operation name="**meterCamisas**">
- <soap:operation soapAction="" />
- - <input>
- <soap:body use="**literal**" />
- </input>
- - <output>
- <soap:body use="**literal**" />
- </output>
- </operation>
- - <operation name="**sacarCamisas**">
- <soap:operation soapAction="" />
- - <input>
- <soap:body use="**literal**" />
- </input>
- - <output>
- <soap:body use="**literal**" />
- </output>
- </operation>

WSDL



- - <operation name="**meterAbrigos**">
- <soap:operation soapAction="" />
- - <input>
- <soap:body use="**literal**" />
- </input>
- - <output>
- <soap:body use="**literal**" />
- </output>
- </operation>
- - <operation name="**sacarAbrigos**">
- <soap:operation soapAction="" />
- - <input>
- <soap:body use="**literal**" />
- </input>
- - <output>
- <soap:body use="**literal**" />
- </output>
- </operation>

WSDL



```
▪ <operation name="meterChaquetas">
▪   <soap:operation soapAction="" />
▪ - <input>
▪   <soap:body use="literal" />
▪ </input>
▪ - <output>
▪   <soap:body use="literal" />
▪ </output>
▪ </operation>
▪ - <operation name="sacarChaquetas">
▪   <soap:operation soapAction="" />
▪ - <input>
▪   <soap:body use="literal" />
▪ </input>
▪ - <output>
▪   <soap:body use="literal" />
▪ </output>
▪ </operation>
▪ </binding>
▪ - <service name="ServicioAlmacenService">
▪ - <port name="ServicioAlmacenPort" binding="tns:ServicioAlmacenPortBinding">
▪   <soap:address location="http://localhost:8080/almacen" />
▪ </port>
▪ </service>
▪ </definitions>
```

Cliente Almacén



```
Símbolo del sistema
C:\jdk1.6\bin>java almacen.ClienteAlmacen
No se ha introducido el tipo de operacion
C:\jdk1.6\bin>java almacen.ClienteAlmacen 3 3
no se puede meter mas abrigos en el almacen
C:\jdk1.6\bin>java almacen.ClienteAlmacen 4 8
41
C:\jdk1.6\bin>java almacen.ClienteAlmacen 4 7
34
C:\jdk1.6\bin>java almacen.ClienteAlmacen 3 5
39
C:\jdk1.6\bin>java almacen.ClienteAlmacen 3 7
46
C:\jdk1.6\bin>java almacen.ClienteAlmacen 3 8
no se puede meter mas abrigos en el almacen
C:\jdk1.6\bin>java almacen.ClienteAlmacen 3 6
no se puede meter mas abrigos en el almacen
C:\jdk1.6\bin>java almacen.ClienteAlmacen 3 4
50
C:\jdk1.6\bin>java almacen.ClienteAlmacen 3 2
no se puede meter mas abrigos en el almacen
C:\jdk1.6\bin>java almacen.ClienteAlmacen 5 8
no se puede meter mas chaquetas del almacen
C:\jdk1.6\bin>java almacen.ClienteAlmacen 8
opcion erronea
C:\jdk1.6\bin>java almacen.ClienteAlmacen 5 7
no se puede meter mas chaquetas del almacen
C:\jdk1.6\bin>java almacen.ClienteAlmacen 6 9
39
C:\jdk1.6\bin>
```

Código Servicio Almacén



```
package almacen;
import javax.jws.WebService;
@WebService
public class ServicioAlmacen
{
    private static int numeroDeCamisas=50;
    private static int numeroDeAbrigos=50;
    private static int numeroDeChaquetas=50;
    public int meterCamisas ( int numero)
    {
        if (numeroDeCamisas + numero <=50)
        {
            numeroDeCamisas += numero;
            return numeroDeCamisas;
        }
        else
        {
            return -1;
        }
    }
}
```

Código Servicio Almacén



```
public int sacarCamisas (int numero)
{
    if (numeroDeCamisas - numero >=0)
    {
        numeroDeCamisas -= numero;
        return numeroDeCamisas;
    }
    else
    {
        return -1;
    }
}
```

Código Servicio Almacén



```
public int meterAbrigos ( int numero )
{
    if (numeroDeAbrigos + numero <51)
    {
        numeroDeAbrigos+= numero;
        return numeroDeAbrigos;
    }
    else
    {
        return -1;
    }
}

public int sacarAbrigos ( int numero)
{
    if (numeroDeAbrigos - numero >=0 )
    {
        numeroDeAbrigos -= numero;
        return numeroDeAbrigos;
    }
    else
    {
        return -1;
    }
}
```

Código Servicio Almacén



```
public int meterChaquetas (int numero)
{
    if (numeroDeChaquetas + numero <51)
    {
        numeroDeCamisas += numero;
        return numeroDeChaquetas;
    }
    else
    {
        return -1;
    }
}

public int sacarChaquetas (int numero)
{
    if (numeroDeChaquetas - numero >=0)
    {
        numeroDeChaquetas -= numero;
        return numeroDeChaquetas;
    }
    else
    {
        return -1; }
}
}
```

Código Cliente



```
//hacemos una llamada a la clase de servicio generada automaticamente
package almacen;
public class ClienteAlmacen
{
    //punto de entrada del cliente
    int operacion;
    int resultado;
    int numero;
    String CadenaSalida=null;
    String datoCadena=null;

    private void mensaje (int opcion)
    {
        switch (opcion)
        {
            case 1:
                CadenaSalida="no se puede meter mas camisas en el almacen";
                break;
            case 2:
                CadenaSalida="no se puede sacar mas camisas del almacen";
                break;
            case 3:
                CadenaSalida="no se puede meter mas abrigos en el almacen";
                break;
            case 4:
                CadenaSalida="no se puede sacar mas abrigos del almacen";
                break;
        }
    }
}
```

Código Cliente



```
case 5:
    CadenaSalida="no se puede meter mas chaquetas del almacen";
    break;
case 6:
    CadenaSalida="no se puede sacar mas chaquetas del almacen";
    break;
}
}
String datoCadena=null;

private void mensaje (int opcion)
{
    switch (opcion)
    {
        case 1:
            CadenaSalida="no se puede meter mas camisas en el almacen";
            break;
        case 2:
            public static void main (String[] args)
            {
                //declaramos la variable cadena encargada de recoger los argumentos
```

Código Cliente



```
if (args.length>0)
{
    //recuperamos la operacion a realizar
    operacion=Integer.parseInt(args[0].trim());

    try
    {
        //establecemos el servicio
        ServicioAlmacenService servicio= new ServicioAlmacenService();
        //obtenos el puerto del servicio
        ServicioAlmacen puerto=servicio.getServicioAlmacenPort();
        switch(operacion)
        {

            case 1:
                //eliminamos los espacios en blanco que pudieramos tener
                datoCadena=args[1].trim();
                numero=Integer.valueOf(datoCadena).intValue();
                resultado= puerto.meterCamisas(numero);
                if (resultado!= -1)
                {
                    CadenaSalida=String.valueOf(resultado);
                }
                else
                {
                    mensaje(1);
                }
                CadenaSalida=String.valueOf(resultado);
                break;
```

Código Cliente



case 2:

```
//eliminamos los espacios en blanco que pudieramos tener
datoCadena=args[1].trim();
//convertimos la cadena en valor de tipo double
numero=Integer.valueOf(datoCadena).intValue();
resultado= puerto.sacarCamisas(numero);
if (resultado!= -1)
{
    CadenaSalida=String.valueOf(resultado);
}
else
{
    mensaje(2);
}
CadenaSalida=String.valueOf(resultado);

break;
```

Código de Cliente



case 3:

```
//eliminamos los espacios en blanco que pudieramos tener
datoCadena=args[1].trim();
numero=Integer.valueOf(datoCadena).intValue();
resultado= puerto.meterAbrigo(numero);
if (resultado!= -1)
{
    CadenaSalida=String.valueOf(resultado);
}
else
{
    mensaje(3);
}
break;
```

Código de Cliente



case 4:

```
//eliminamos los espacios en blanco que pudieramos tener
datoCadena=args[1].trim();
numero=Integer.valueOf(datoCadena).intValue();
resultado= puerto.sacarCamisas(numero);
if (resultado!= -1)
{
    CadenaSalida=String.valueOf(resultado);
}
else
{
    mensaje(4);
}
CadenaSalida=String.valueOf(resultado);
break;
```

Código de Cliente



case 6:

```
//eliminamos los espacios en blanco que pudieramos tener
datoCadena=args[1].trim();
numero=Integer.valueOf(datoCadena).intValue();
resultado= puerto.meterCamisas(radio);
if (resultado!= -1)
{
    CadenaSalida=String.valueOf(resultado);
}
else
{
    mensaje(6);
}
CadenaSalida=String.valueOf(resultado);
break;

default:
CadenaSalida="opcion erronea";
```

```
}
```

Código de Cliente



```
    }
    catch (Exception ex)
    {
        CadenaSalida=ex.toString();
    }
    finally
    {
        System.out.println(CadenaSalida);
    }
    }
    else
    {
        CadenaSalida="No se ha introducido el tipo de operacion";
        System.out.println(CadenaSalida);
    }
    }
```

Código de PuntoFinalAlmacenWS



```
▪ package almacen;  
▪ import javax.xml.ws.Endpoint;  
  
▪ public class PuntoFinalAlmacenWS  
▪ {  
▪     public static void main(String[] args) {  
▪         String endpointAddress = "http://localhost:8080/almacen";  
▪         Endpoint.publish(endpointAddress, new ServicioAlmacen());  
▪         System.out.println("El Servicio Web para de almacen");  
▪         System.out.println("WSDL: " + endpointAddress + "?wsdl");  
▪     }  
▪ }
```