1. Se debe crear Nuevo Proyecto Maven o Spring (recomiendo utilizar IntelliJ o Eclipse)
2. Crear el XSD

Situarlo en ReadOutbound\src\main\resources\<NombreXSD>.xsd  
  
Nota: ReadOutbound es la raiz de mi servicio, en su caso deben poner el nombre de la operación y mantener la estructura de packages

Debe tener la estructura estandar de un xsd solo que tiene que tener minimo 2 elementos que terminen con <nombre>“\_request” y <nombre>“\_response” porque estos elementos serán los que le darán nombre a la operacion del WSDL cuando se genere, ejemplo de xsd:

<**xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:tns="http://spring.io/guides/gs-producing-web-service"  
 targetNamespace="http://spring.io/guides/gs-producing-web-service" elementFormDefault="qualified"**>  
  
 *<!-- SHARED -->* <**xsd:complexType name="guardar\_orden\_input\_t"**>  
 <**xsd:sequence**>  
 <**xsd:element name="CRM" type="xsd:string" minOccurs="1" maxOccurs="1"**/>  
 <**xsd:element name="APPNUMBER" type="xsd:string" minOccurs="1" maxOccurs="1"**/>  
 <**xsd:element name="ACTIVITYTYPE" type="xsd:string" minOccurs="1" maxOccurs="1"**/>  
 <**xsd:element name="XA\_TECHNOLOGY" type="xsd:string" minOccurs="1" maxOccurs="1"**/>  
 <**xsd:element name="XA\_MOVE" type="xsd:integer" minOccurs="1" maxOccurs="1"**/>  
 <**xsd:element name="COUNTRY\_CODETUS" type="xsd:string" minOccurs="1" maxOccurs="1"**/>  
 <**xsd:element name="XA\_REGION" type="xsd:string" minOccurs="1" maxOccurs="1"**/>

….

….

….  
 <**xsd:element name="XA\_SELLER" type="xsd:string" minOccurs="1" maxOccurs="1"**/>  
 <**xsd:element name="TIMEZONE" type="xsd:string" minOccurs="1" maxOccurs="1"**/>  
 <**xsd:element name="LANGUAJE" type="xsd:string" minOccurs="1" maxOccurs="1"**/>  
 <**xsd:element name="ADITTIONALDATA" type="xsd:string" minOccurs="1" maxOccurs="1"**/>  
 <**xsd:element name="ADITTIONALSUBDATA" type="xsd:string" minOccurs="1" maxOccurs="1"**/>  
 </**xsd:sequence**>  
 </**xsd:complexType**>  
  
 <**xsd:complexType name="guardar\_orden\_output\_t"**>  
 <**xsd:sequence**>  
 <**xsd:element name="CRM\_APPNUMBER" type="xsd:string" minOccurs="1" maxOccurs="1"**/>  
 <**xsd:element name="estado" type="xsd:integer" minOccurs="1" maxOccurs="1"**/>  
 <**xsd:element name="descripcion" type="xsd:string" minOccurs="1" maxOccurs="1"**/>  
 </**xsd:sequence**>  
 </**xsd:complexType**>  
  
 <**xsd:element name="guardar\_orden\_request"**>  
 <**xsd:complexType**>  
 <**xsd:sequence**>  
 <**xsd:element name="input" type="tns:guardar\_orden\_input\_t" minOccurs="1" maxOccurs="1"**/>  
 </**xsd:sequence**>  
 </**xsd:complexType**>  
 </**xsd:element**>  
  
 <**xsd:element name="guardar\_orden\_response"**>  
 <**xsd:complexType**>  
 <**xsd:sequence**>  
 <**xsd:element name="output" type="tns:guardar\_orden\_output\_t" minOccurs="1" maxOccurs="unbounded"**/>  
 </**xsd:sequence**>  
 </**xsd:complexType**>  
 </**xsd:element**>  
  
</**xsd:schema**>

El xsd anterior al generar el wsdl se forma la siguiente operacion

<wsdl:operation name="guardar\_orden">

<wsdl:input message="tns: guardar\_orden \_request" name=" guardar\_orden\_request"/>

<wsdl:output message="tns: guardar\_orden\_response" name=" guardar\_orden\_response"/>

</wsdl:operation>

1. Crear la definicion del servicio

Crear una clase en la ruta ReadOutbound\src\main\java\com\service\readoutbound\webservice\<NombreServicio>WS

Por convencion el nombre de la clase termina en “WS” ejemplo de clase

**package** com.service.readoutbound.webservice;  
  
**import** org.springframework.context.annotation.Bean;  
**import** org.springframework.context.annotation.Configuration;  
**import** org.springframework.core.io.ClassPathResource;  
**import** org.springframework.ws.wsdl.wsdl11.DefaultWsdl11Definition;  
**import** org.springframework.xml.xsd.SimpleXsdSchema;  
**import** org.springframework.xml.xsd.XsdSchema;  
  
*/\*\*  
 \* Created by Sebastián Muñoz Zapata  
 \* on 2/12/2018 5:19 PM  
 \*/*@Configuration  
**public class** AwebProcGuardarOrdenWS {  
  
 @Bean(name = **"AwebProcGuardarOrden"**)  
 **public** DefaultWsdl11Definition defaultWsdl11Definition(**XsdSchema awebProcGuardarOrdendSchema**) {  
 DefaultWsdl11Definition wsdl11Definition = **new** DefaultWsdl11Definition();  
 wsdl11Definition.setServiceName(**"agent\_service"**);  
 wsdl11Definition.setRequestSuffix(**"\_request"**);  
 wsdl11Definition.setResponseSuffix(**"\_response"**);  
 wsdl11Definition.setPortTypeName(**"agent\_port\_type"**);  
 wsdl11Definition.setLocationUri(**"/ws"**);  
 wsdl11Definition.setTargetNamespace(**"urn:toatech:agent"**);  
 wsdl11Definition.setSchema(awebProcGuardarOrdendSchema);  
 **return** wsdl11Definition;  
 }  
  
 @Bean  
 **public XsdSchema awebProcGuardarOrdendSchema**() {  
 **return new** SimpleXsdSchema(**new** ClassPathResource(**"AwebProcGuardarOrden.xsd"**));  
 }  
}

Consideraciones

* En el metodo defaultWsdl11Definition recibe por parametro **XsdSchema awebProcGuardarOrdendSchema** el bean de abajo que devuelve la definicion del Xsd
* @Bean(name = **" AwebProcGuardarOrden "**) este nombre sera el nombre del wsdl en el endpoint
* wsdl11Definition.setRequestSuffix(**"\_request"**); según la convencion del xsd para generar la operación
* wsdl11Definition.setResponseSuffix(**"\_response"**); según la convencion del xsd para generar la operación
* wsdl11Definition.setLocationUri(**"/ws"**); path que se incluye en la direccion final del servicio ejemplo: http://localhost:7001/AgendaWeb/**ws**/ **AwebProcGuardarOrden**
* **return new** SimpleXsdSchema(**new** ClassPathResource(**"**<NombreXSD>**"**)); nombre del xsd generado en paso 1

1. Generar las clases de las entidades del XSD

Las clases de las entidesdes se generan mediante maven ejecutando el commando **mvn clean package**

Este commando genera las clases entidades a partir del xsd. Las clases generadas pueden verse en la ruta ReadOutbound\src\main\java\io\spring\guides\gs\_producing\_web\_service

1. Crear la clase Service que es la encargada de la logica de la operacion

Crear una clase en la ruta ReadOutbound\src\main\java\com\service\readoutbound\service\<NombreEndpoint>Service

Por convencion el nombre de la clase termina en “Service”. Esta es la clase encargada de la logica de la operacion que ejecuta la llamada al SP y retorna la informacion para ser respondida en el servicio.

**package** com.service.readoutbound.service;  
  
**import** org.springframework.stereotype.Service;  
  
**import** javax.persistence.EntityManager;  
**import** javax.persistence.ParameterMode;  
**import** javax.persistence.PersistenceContext;  
**import** javax.persistence.StoredProcedureQuery;  
**import** java.util.ArrayList;  
**import** java.util.List;  
  
*/\*\*  
 \* Created by Sebastián Muñoz Zapata  
 \* on 3/1/2018 11:18 AM  
 \*/*@Service  
**public class** AwebProcGuardarOrdenService {  
  
 @PersistenceContext  
 **private** EntityManager **entityManager**;  
  
 **public** List<Object> **AwebProcGuardarOrden**(String CRM, String APPNUMBER, String ACTIVITYTYPE, String XA\_TECHNOLOGY,  
 Integer XA\_MOVE, Integer XA\_URGENT, Integer XA\_COMPLAIN\_TYPE, Integer SCHEDULEWEB,  
 String COUNTRY\_CODETUS, String XA\_REGION, String XA\_PROVINCE, String STATEPROVINCE,  
 String XA\_COMUNA, String CITY, String XA\_NODE, String STATUS, String XA\_INSTALL\_DETAILS,  
 Integer XA\_NR\_POINTS\_TV, Integer XA\_NR\_POINT\_INTERNET, Integer XA\_NR\_POINT\_PHONE,  
 String XA\_STREET\_TYPE, String XA\_STREET\_NR, String STREETADDRESS,  
 String XA\_ADDRESS\_COMPLEMENT, String XA\_ADDRESS\_COMPLEMENT\_2, String XA\_REF\_SEC,  
 String POSTALCODE, String COORDSTATUS, String LONGITUDE, String LATITUDE,  
 String SETPOSITIONINROUTE, Integer ACTIVITYID, String POSITION, String RECORDTYPE,  
 String CUSTOMERNUMBER, String XA\_RUT, String CUSTOMERNAME, Integer XA\_CUSTOMER\_TYPE,  
 Integer XA\_CUSTORMER\_CATEGORY, String CUSTOMERCELL, String CUSTOMERPHONE,  
 String XA\_CONTACT\_PHONE\_2, String CUSTOMEREMAIL, Integer XA\_SELL\_TYPE, String XA\_SELLER,  
 String TIMEZONE, String LANGUAJE, String ADITTIONALDATA, String ADITTIONALSUBDATA  
 ) {  
 StoredProcedureQuery query = **entityManager**.createStoredProcedureQuery(**"AWEB\_PROC\_GUARDAR\_ORDEN"**);  
 query.registerStoredProcedureParameter(**"CRM"**, String.**class**, ParameterMode.***IN***);  
 query.registerStoredProcedureParameter(**"APPNUMBER"**, String.**class**, ParameterMode.***IN***);  
 query.registerStoredProcedureParameter(**"ACTIVITYTYPE"**, String.**class**, ParameterMode.***IN***);  
 query.registerStoredProcedureParameter(**"XA\_TECHNOLOGY"**, String.**class**, ParameterMode.***IN***);  
 query.registerStoredProcedureParameter(**"XA\_MOVE"**, Integer.**class**, ParameterMode.***IN***);

……….  
 query.registerStoredProcedureParameter(**"XA\_URGENT"**, Integer.**class**, ParameterMode.***IN***);   
 query.registerStoredProcedureParameter(**"XA\_SELL\_TYPE"**, Integer.**class**, ParameterMode.***IN***);  
 query.registerStoredProcedureParameter(**"XA\_SELLER"**, String.**class**, ParameterMode.***IN***);  
 query.registerStoredProcedureParameter(**"TIMEZONE"**, String.**class**, ParameterMode.***IN***);  
 query.registerStoredProcedureParameter(**"LANGUAJE"**, String.**class**, ParameterMode.***IN***);  
 query.registerStoredProcedureParameter(**"ADITTIONALDATA"**, String.**class**, ParameterMode.***IN***);   
  
 query.registerStoredProcedureParameter(**"CRM\_APPNUMBER"**, String.**class**, ParameterMode.***OUT***);  
 query.registerStoredProcedureParameter(**"estado"**, Integer.**class**, ParameterMode.***OUT***);  
 query.registerStoredProcedureParameter(**"descripcion"**, String.**class**, ParameterMode.***OUT***);  
  
 query.setParameter(**"CRM"**, CRM);  
 query.setParameter(**"APPNUMBER"**, APPNUMBER);  
 query.setParameter(**"ACTIVITYTYPE"**, ACTIVITYTYPE);  
 query.setParameter(**"XA\_TECHNOLOGY"**, XA\_TECHNOLOGY);  
 query.setParameter(**"XA\_MOVE"**, XA\_MOVE);  
 query.setParameter(**"XA\_URGENT"**, XA\_URGENT);

………

query.setParameter(**"XA\_SELLER"**, XA\_SELLER);  
 query.setParameter(**"TIMEZONE"**, TIMEZONE);  
 query.setParameter(**"LANGUAJE"**, LANGUAJE);  
 query.setParameter(**"ADITTIONALDATA"**, ADITTIONALDATA);  
 query.setParameter(**"ADITTIONALSUBDATA"**, ADITTIONALSUBDATA);  
 query.execute();  
 List<Object> result = **new** ArrayList<>();  
 result.add(query.getOutputParameterValue(**"CRM\_APPNUMBER"**));  
 result.add(query.getOutputParameterValue(**"estado"**));  
 result.add(query.getOutputParameterValue(**"descripcion"**));  
 **return** result;  
 }  
}

Consideraciones

La clase tiene un solo metodo **AwebProcGuardarOrden** que recive los mismos parametros que el SP y retorna una lista de objetos que se forma de la salida del SP. Primeramente se crea la deifnicion del SP mediante el metodo StoredProcedureQuery query = **entityManager**.createStoredProcedureQuery(**"<nombre del SP>"**); y se definen los parametros de entrada y salida mediante el metodo “query.registerStoredProcedureParameter(**"XA\_MOVE"**, Integer.**class**, ParameterMode.***IN***);” para cada uno de los parametros de entrada y “query.registerStoredProcedureParameter(**"descripcion"**, String.**class**, ParameterMode.***OUT***);” para cada uno de los parametros de salida. Posteriormente se llenan los parametros de entrada del SP “query.setParameter(**"APPNUMBER"**, APPNUMBER);” y finalmente se ejecuta el SP con “query.execute();” y se llena la lista resultado con los parametros de salida del SP “result.add(query.getOutputParameterValue(**"CRM\_APPNUMBER"**));” por cada uno de los output del SP.

1. Crear la clase del endpoint

Crear una clase en la ruta ReadOutbound\src\main\java\com\service\readoutbound\endpoint\<NombreEndpoint>Endpoint

Por convencion el nombre de la clase termina en “Endpoint”. Esta clase es la que recibe el la llamada del servicio y llama a la logica para realizer la accion del mismo ejemplo de clase

**package** com.service.readoutbound.endpoint;  
  
**import** com.service.readoutbound.service.AwebProcGuardarOrdenService;  
**import** io.spring.guides.gs\_producing\_web\_service.GuardarOrdenInputT;  
**import** io.spring.guides.gs\_producing\_web\_service.GuardarOrdenOutputT;  
**import** io.spring.guides.gs\_producing\_web\_service.GuardarOrdenRequest;  
**import** io.spring.guides.gs\_producing\_web\_service.GuardarOrdenResponse;  
**import** org.springframework.beans.factory.annotation.Autowired;  
**import** org.springframework.ws.server.endpoint.annotation.Endpoint;  
**import** org.springframework.ws.server.endpoint.annotation.PayloadRoot;  
**import** org.springframework.ws.server.endpoint.annotation.RequestPayload;  
**import** org.springframework.ws.server.endpoint.annotation.ResponsePayload;  
  
**import** java.math.BigInteger;  
**import** java.util.List;  
  
*/\*\*  
 \* Created by Sebastián Muñoz Zapata  
 \* on 2/12/2018 3:44 PM  
 \*/*@Endpoint  
**public class** AwebProcGuardarOrdenEndpoint {  
 **private static final** String ***NAMESPACE\_URI*** = **"http://spring.io/guides/gs-producing-web-service"**;  
  
 @Autowired  
 **private** AwebProcGuardarOrdenService **guardarOrdenService**;  
  
 @PayloadRoot(namespace = ***NAMESPACE\_URI***, localPart = **"guardar\_orden\_request"**)  
 @ResponsePayload  
 **public** GuardarOrdenResponse guardarOrden(@RequestPayload GuardarOrdenRequest request) {  
 GuardarOrdenInputT input = request.getInput();  
 GuardarOrdenResponse response = **new** GuardarOrdenResponse();  
 GuardarOrdenOutputT outputT = **new** GuardarOrdenOutputT();  
 **try** {  
 List<Object> result = **guardarOrdenService**.AwebProcGuardarOrden(input.getCRM(), input.getAPPNUMBER(),  
 input.getACTIVITYTYPE(), input.getXATECHNOLOGY(), input.getXAMOVE().intValue(),  
 input.getXAURGENT().intValue(), input.getXACOMPLAINTYPE().intValue(), input.getSCHEDULEWEB().intValue(),  
 input.getCOUNTRYCODETUS(), input.getXAREGION(), input.getXAPROVINCE(), input.getSTATEPROVINCE(),  
 input.getXACOMUNA(), input.getCITY(), input.getXANODE(), input.getSTATUS(), input.getXAINSTALLDETAILS(),  
 input.getXANRPOINTSTV().intValue(), input.getXANRPOINTINTERNET().intValue(), input.getXANRPOINTPHONE().intValue(),  
 input.getXASTREETTYPE(), input.getXASTREETNR(), input.getSTREETADDRESS(),  
 input.getXAADDRESSCOMPLEMENT(), input.getXAADDRESSCOMPLEMENT2(), input.getXAREFSEC(),  
 input.getPOSTALCODE(), input.getCOORDSTATUS(), input.getLONGITUDE(), input.getLATITUDE(),  
 input.getSETPOSITIONINROUTE(), input.getACTIVITYID().intValue(), input.getPOSITION(), input.getRECORDTYPE(),  
 input.getCUSTOMERNUMBER(), input.getXARUT(), input.getCUSTOMERNAME(), input.getXACUSTOMERTYPE().intValue(),  
 input.getXACUSTORMERCATEGORY().intValue(), input.getCUSTOMERCELL(), input.getCUSTOMERPHONE(),  
 input.getXACONTACTPHONE2(), input.getCUSTOMEREMAIL(), input.getXASELLTYPE().intValue(), input.getXASELLER(),  
 input.getTIMEZONE(), input.getLANGUAJE(), input.getADITTIONALDATA(), input.getADITTIONALSUBDATA());  
 outputT.setCRMAPPNUMBER((String) result.get(0));  
 outputT.setEstado(BigInteger.*valueOf*((Integer) result.get(1)));  
 outputT.setDescripcion((String) result.get(2));  
 } **catch** (Exception error) {  
 outputT.setCRMAPPNUMBER(**""**);  
 outputT.setEstado(BigInteger.*valueOf*(0));  
 outputT.setDescripcion(**"No Ok"**);  
 }  
 response.getOutput().add(outputT);  
 **return** response;  
 }  
}

Consideraciones

* Inyectar un objeto de la clase del paso 4 “**private** AwebProcGuardarOrdenService **guardarOrdenService**;” para que realice la logica de la llamada
* @PayloadRoot(namespace = ***NAMESPACE\_URI***, localPart = **"guardar\_orden\_request"**) el localpart es el nombre del elemento que declaramos en el xsd como el request del servicio
* Crear el metodo “**public** GuardarOrdenResponse guardarOrden(@RequestPayload GuardarOrdenRequest request)” que recibe como parametron la clase del request que se generó en el paso 3 mediante maven tiene el mismo nombre del request pero con notacion CamelCase y que retorna “GuardarOrdenResponse” que representa la clase del response generada en el paso 3
* Se obtiene la entrada de la llamada al servicio con “GuardarOrdenInputT input = request.getInput();” para tomar los datos del request notese que GuardarOrdenInputT es la clase que representa al elemento que definimos en el xsd
* Se inicializa el objeto de respuesta “GuardarOrdenResponse response = **new** GuardarOrdenResponse();” que es del tipo de la respuesta del xsd
* Se inicializa el objecto del elemento contenido en el response “GuardarOrdenOutputT outputT = **new** GuardarOrdenOutputT();”
* Se realiza la llamada al service pasandole todas las entradas del SP “List<Object> result = **guardarOrdenService**.AwebProcGuardarOrden(input.getCRM(), input.getAPPNUMBER(),” esta llamada se incluye en un Try catch para garantizar la respuesta al WS en cualquier caso.
* Se setean los parametros del element respuesta “outputT.setCRMAPPNUMBER((String) result.get(0));” para todos los parametros del element
* Se adiciona el element al response “response.getOutput().add(outputT);” y se retorna.

1. Ejecutar la aplicacion

Al ejecutar la aplicacion ir a la ruta http://localhost:7001/AgendaWeb/**ws**/**AwebProcGuardarOrden** y si toda ha salido bien debe mostrar un wsdl como este:

**Nota**: Debe generarse un WAR, el cual debe ser instalado en el WebLogic Haibu: http://159.203.68.81:7001/console.

<wsdl:definitions targetNamespace="urn:toatech:agent" xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/" xmlns:sch="http://spring.io/guides/gs-producing-web-service" xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/" xmlns:tns="urn:toatech:agent">

<wsdl:types>

<xsd:schema elementFormDefault="qualified" targetNamespace="http://spring.io/guides/gs-producing-web-service" xmlns:tns="http://spring.io/guides/gs-producing-web-service" xmlns:xsd="http://www.w3.org/2001/XMLSchema">

<!--SHARED-->

<xsd:complexType name="consultar\_orden\_input\_t">

<xsd:sequence>

<xsd:element maxOccurs="1" minOccurs="0" name="P\_ID\_ORDER" type="xsd:string"/>

</xsd:sequence>

</xsd:complexType>

<xsd:complexType name="consultar\_orden\_output\_t">

<xsd:sequence>

<xsd:element maxOccurs="1" minOccurs="1" name="p\_APPTNUMBER" type="xsd:string"/>

<xsd:element maxOccurs="1" minOccurs="1" name="P\_ID\_CRM" type="xsd:string"/>

<xsd:element maxOccurs="1" minOccurs="1" name="p\_ACTIVITYTYPE" type="xsd:string"/>

<xsd:element maxOccurs="1" minOccurs="1" name="p\_XA\_TECHNOLOGY" type="xsd:string"/>

<xsd:element maxOccurs="1" minOccurs="1" name="p\_XA\_RETRIES" type="xsd:string"/>

<xsd:element maxOccurs="1" minOccurs="1" name="p\_XA\_MOVE" type="xsd:integer"/>

<xsd:element maxOccurs="1" minOccurs="1" name="p\_XA\_URGENT" type="xsd:integer"/>

……….

……….

<xsd:element maxOccurs="1" minOccurs="1" name="p\_UPDATE\_AT" type="xsd:date"/>

<xsd:element maxOccurs="1" minOccurs="1" name="p\_RESOURCEID" type="xsd:string"/>

<xsd:element maxOccurs="1" minOccurs="1" name="p\_XA\_CONTACT\_PHONE\_2" type="xsd:string"/>

<xsd:element maxOccurs="1" minOccurs="1" name="p\_WORKZONE" type="xsd:string"/>

<xsd:element maxOccurs="1" minOccurs="1" name="P\_ACTIVO" type="xsd:string"/>

<xsd:element maxOccurs="1" minOccurs="1" name="p\_TIPO\_ESTADO" type="xsd:string"/>

<xsd:element maxOccurs="1" minOccurs="1" name="estado" type="xsd:integer"/>

<xsd:element maxOccurs="1" minOccurs="1" name="descripcion" type="xsd:string"/>

</xsd:sequence>

</xsd:complexType>

<xsd:element name="consultar\_orden\_request">

<xsd:complexType>

<xsd:sequence>

<xsd:element maxOccurs="1" minOccurs="1" name="input" type="tns:consultar\_orden\_input\_t"/>

</xsd:sequence>

</xsd:complexType>

</xsd:element>

<xsd:element name="consultar\_orden\_response">

<xsd:complexType>

<xsd:sequence>

<xsd:element maxOccurs="unbounded" minOccurs="1" name="output" type="tns:consultar\_orden\_output\_t"/>

</xsd:sequence>

</xsd:complexType>

</xsd:element>

</xsd:schema>

</wsdl:types>

<wsdl:message name="consultar\_orden\_request">

<wsdl:part element="sch:consultar\_orden\_request" name="consultar\_orden\_request"/>

</wsdl:message>

<wsdl:message name="consultar\_orden\_response">

<wsdl:part element="sch:consultar\_orden\_response" name="consultar\_orden\_response"/>

</wsdl:message>

<wsdl:portType name="agent\_port\_type">

<wsdl:operation name="consultar\_orden">

<wsdl:input message="tns:consultar\_orden\_request" name="consultar\_orden\_request"/>

<wsdl:output message="tns:consultar\_orden\_response" name="consultar\_orden\_response"/>

</wsdl:operation>

</wsdl:portType>

<wsdl:binding name="agent\_port\_typeSoap11" type="tns:agent\_port\_type">

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

<wsdl:operation name="consultar\_orden">

<soap:operation soapAction=""/>

<wsdl:input name="consultar\_orden\_request">

<soap:body use="literal"/>

</wsdl:input>

<wsdl:output name="consultar\_orden\_response">

<soap:body use="literal"/>

</wsdl:output>

</wsdl:operation>

</wsdl:binding>

<wsdl:service name="agent\_service">

<wsdl:port binding="tns:agent\_port\_typeSoap11" name="agent\_port\_typeSoap11">

<soap:address location="http://159.203.68.81:7001/AgendaWeb/ws"/>

</wsdl:port>

</wsdl:service>

</wsdl:definitions>