Definiendo EJBs y Web Services para tablas de bases de datos.

RGGH.

Diciembre de 2018.

Para SCE 201803

En este documento se dan las instrucciones para definir EJBs de entidad sobre tablas relacionadas en bases de datos configuradas en Glassfish y los WebServices configurados sobre la funcionalidad de los EJB´s.

La realización de los Web Services se hará en los siguientes pasos:

Paso 1) Definir la aplicación contenedora del Módulo EJB. Seleccionar el servidor Glassfish y EE 7 o EE 6. Dar Ok para que se genere el proyecto.

Paso 2) Definir los EJB de entidad sobre la Base de Datos y sus Beans de sesión (interfases o "Facades") con la funcionalidad habitual de los EJB´s de entidad.

Paso 3) Definir el Web Service y en su caso agregar los métodos adicinales requeridos por la aplicación.

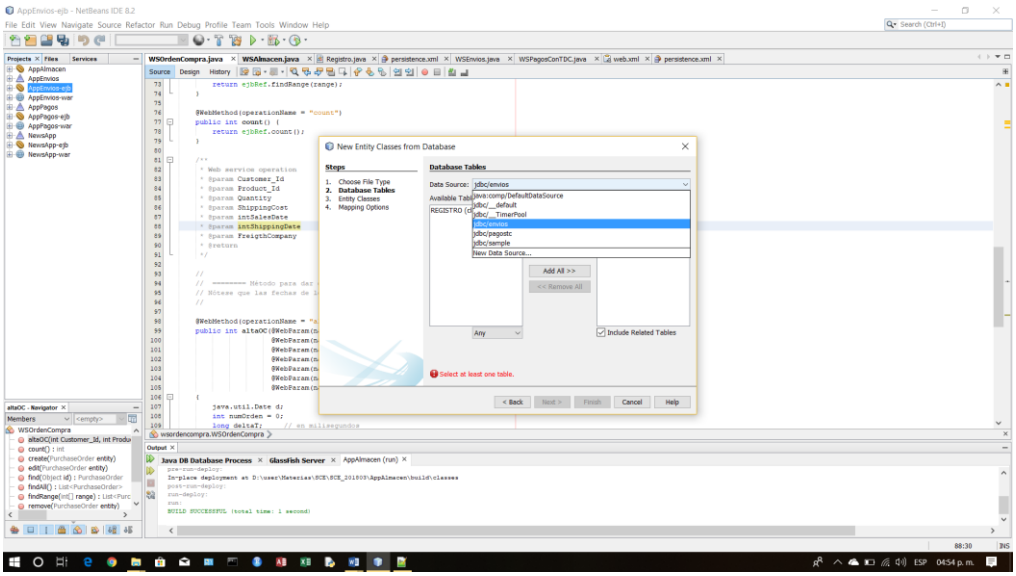
Paso 4) Probar el Web Service y obtener la dirección del WSDL.

El ejemplo se hará con el Módulo de EJB para la base de datos de envíos.

Como precondición se tiene la base de datos configurada en Glassfish. De no ser así, antes de deployar el módulo de EJB´s con los beans de entidad de la base de datos y el o los web services correspondientes se deberá configurar la BD.

Paso 1) Definiendo la aplicación contenedora del Módulo EJB.
En Netbeans generar un nuevo proyecto Java EE-> EJB Module, darle como nombre EJBEnvios.

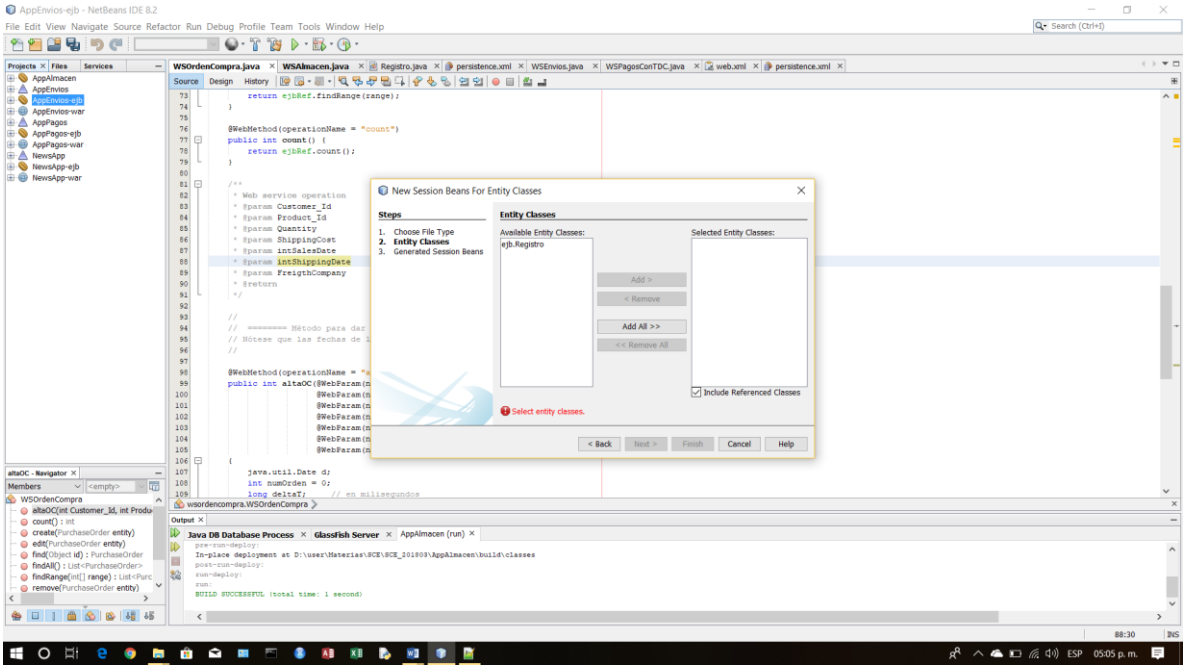
Paso 2) Definir los EJB de entidad sobre la Base de Datos y sus Beans de sesión (interfases o "Facades") con la funcionalidad habitual de los EJB´s de entidad.
Sobre el proyecto damos **New->(Other) Entity Clases from Database** y seleccionamos la base de datos jdbc/envíos.



Ello hace aparecer las Tablas de la base (en este caso la tabla REGISTRO). La seleccionamos para construir el EJB. Solicita un nombre de paquete, especificarle algo así como ejbenvios o simplemente ejb.

Posteriormente se deben crear las fachadas para los EJB´s de entidad que se definieron anteriormente.

Sobre el proyecto dar New->(Other) Session Beans for Entity Classes



Seleccionar los Entity Beans para los que deseamos las fachadas, en este caso ejb.Registro.

Seleccionar el mismo paquete que anteriormente definimos.

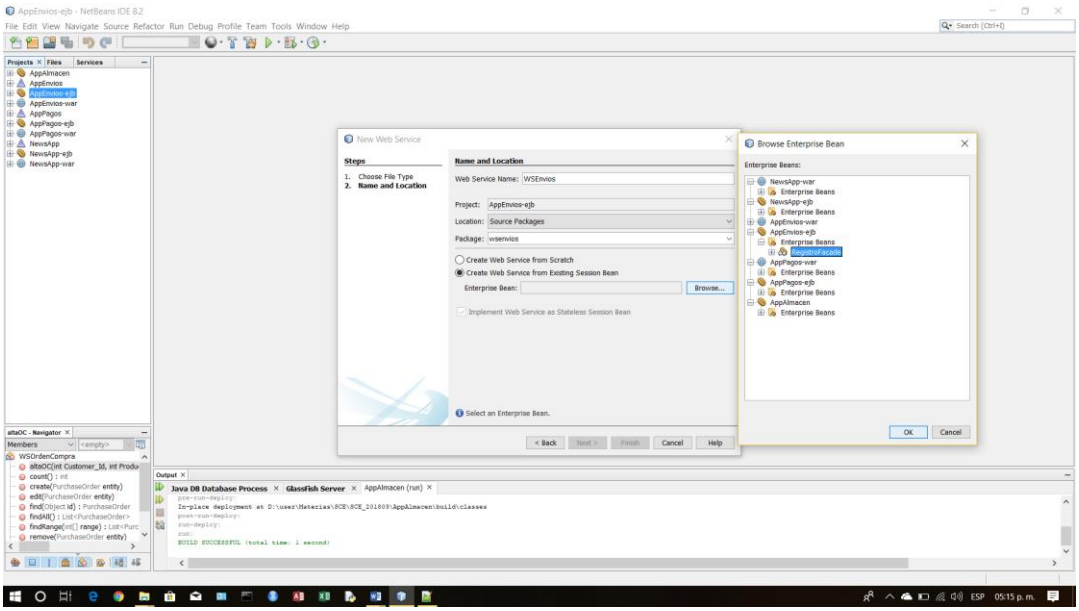
Darle “Save” al proyecto. Esto hará que el proyecto se “deployee” en el Glassfish.

(si la BD no ha sido configurada esto marcará un error de deploy, configurar la BD según las instrucciones del documento ***ConfigurandoBasesDeDatosParaEJByWS_SCE_20803.pdf***).

Paso 3) Definir el Web Service y en su caso agregar los métodos adicionales requeridos por la aplicación.
Para definir el Web Service con el vean de fachada (Facade) correspondiente se da sobre el proyecto en cuestión New->(Other) Web Service.

Seleccionamos la opción de Create Web service from Existing Session Bean (la fachada que creamos en el proyecto en cuestión EJBEnvios).
Damos Next y Finish o Finish.

Nótese que el Web Service tabaja al Bean como StatelessSession Bean.



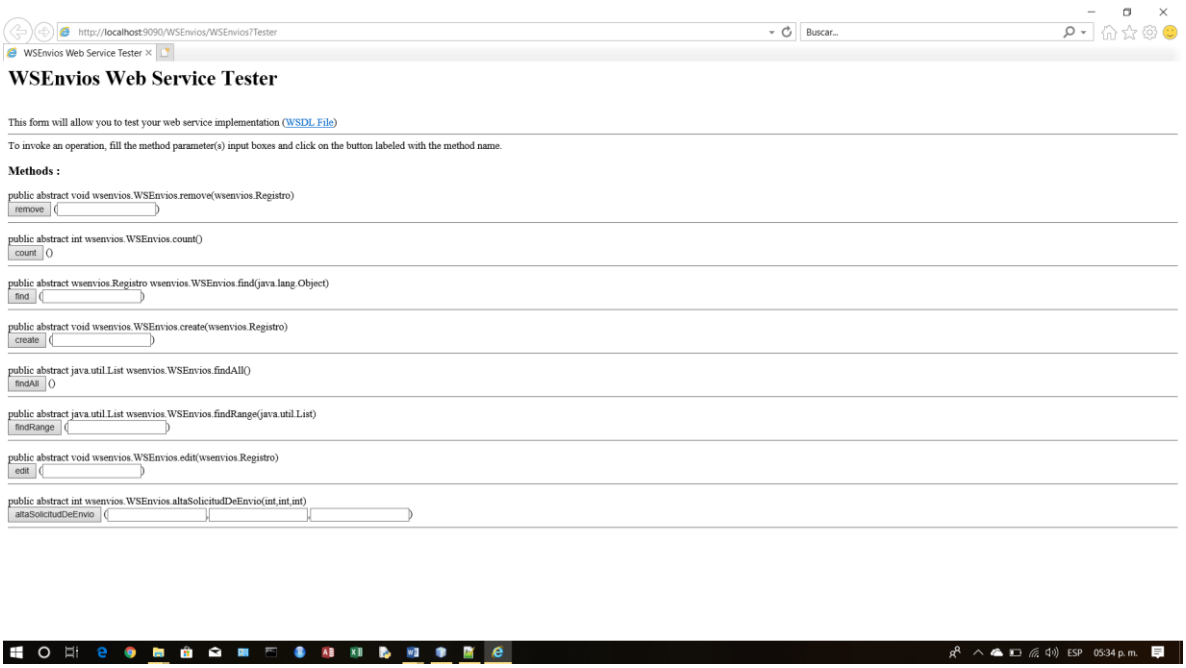
Estando creado el Web Service. A este Web Service le agregaremos el método "**altaSolicitudDeEnvio**". Para ello abrimos el código en java del Web service y con la opción de "***Insert Code…***" seleccionamos "***Add Web Service Operation…"*** y definimos los parámetros correspondientes a la interfaz mostrada y agregamos el código requerido:

```
//
//  ========== Metodo adicional para alta de solicitud de envíos ========
//
   /**
    * Web service operation
    * @param Customer_ID
    * @param Orden_Compra
    * @param Status
```

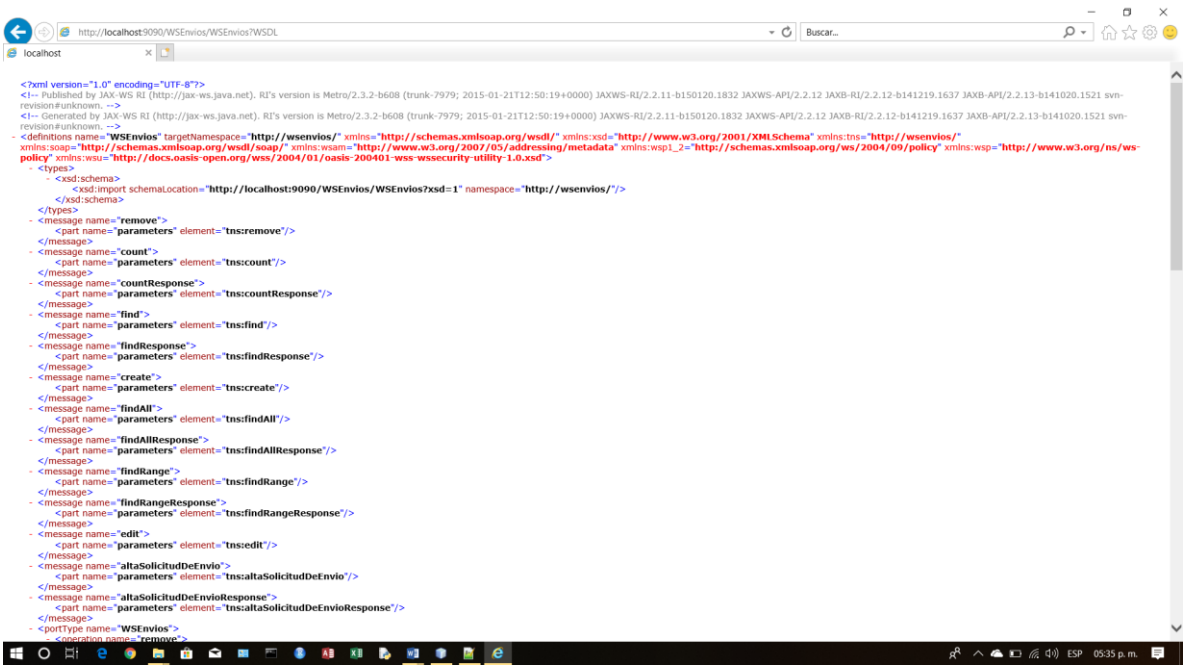
```
     * @return
     */
    @WebMethod(operationName = "altaSolicitudDeEnvio")
    public int altaSolicitudDeEnvio(
            @WebParam(name = "Customer_ID") int Customer_ID,
            @WebParam(name = "Orden_Compra") int Orden_Compra,
            @WebParam(name = "Status") char Status)
    {
        Registro objReg = new Registro();
        int intEntrega_id = ejbRef.count()+1;
        objReg.setEntregaId(intEntrega_id);
        objReg.setCustomerId(Customer_ID);
        objReg.setOrdenCompra(Orden_Compra);
        objReg.setStatus('P');
        ejbRef.create(objReg);
        return intEntrega_id;
    }
```

Damos Clear and Build al Proyecto del EJB Module en cuestión.
Damos deploy al proyecto.

Paso 4) Probar el Web Service y obtener la dirección del WSDL.
Para probar el proyecto, una vez deployado el proyecto probamos el Web Service por medio de la opción de Test Web service que contienen el Netbeans para tales modalidades. De ahí obtenemos el WSDL:



Y su WSDL y su correspondiente URL:


http://localhost:9090/WSEnvios/WSEnvios?WSDL
Buscar...
localhost
<?xml version="1.0" encoding="UTF-8"?>
<!-- Published by JAX-WS RI (http://jax-ws.java.net). RI's version is Metro/2.3.2-b608 (trunk-7979; 2015-01-21T12:50:19+0000) JAXWS-RI/2.2.11-b150120.1832 JAXWS-API/2.2.12 JAXB-RI/2.2.12-b141219.1637 JAXB-API/2.2.13-b141020.1521 svn-revision#unknown. -->
<!-- Generated by JAX-WS RI (http://jax-ws.java.net). RI's version is Metro/2.3.2-b608 (trunk-7979; 2015-01-21T12:50:19+0000) JAXWS-RI/2.2.11-b150120.1832 JAXWS-API/2.2.12 JAXB-RI/2.2.12-b141219.1637 JAXB-API/2.2.13-b141020.1521 svn-revision#unknown. -->
<definitions name="WSEnvios" targetNamespace="http://wsenvios/" xmlns="http://schemas.xmlsoap.org/wsdl/" xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:tns="http://wsenvios/" xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/" xmlns:wsam="http://www.w3.org/2007/05/addressing/metadata" xmlns:wsp1_2="http://schemas.xmlsoap.org/ws/2004/09/policy" xmlns:wsp="http://www.w3.org/ns/ws-policy" xmlns:wsu="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-utility-1.0.xsd">
<types>
<xsd:schema>
<xsd:import schemaLocation="http://localhost:9090/WSEnvios/WSEnvios?xsd=1" namespace="http://wsenvios/"/>
</xsd:schema>
</types>
<message name="remove">
<part name="parameters" element="tns:remove"/>
</message>
<message name="count">
<part name="parameters" element="tns:count"/>
</message>
<message name="countResponse">
<part name="parameters" element="tns:countResponse"/>
</message>
<message name="find">
<part name="parameters" element="tns:find"/>
</message>
<message name="findResponse">
<part name="parameters" element="tns:findResponse"/>
</message>
<message name="create">
<part name="parameters" element="tns:create"/>
</message>
<message name="findAll">
<part name="parameters" element="tns:findAll"/>
</message>
<message name="findAllResponse">
<part name="parameters" element="tns:findAllResponse"/>
</message>
<message name="findRange">
<part name="parameters" element="tns:findRange"/>
</message>
<message name="findRangeResponse">
<part name="parameters" element="tns:findRangeResponse"/>
</message>
<message name="edit">
<part name="parameters" element="tns:edit"/>
</message>
<message name="altaSolicitudDeEnvio">
<part name="parameters" element="tns:altaSolicitudDeEnvio"/>
</message>
<message name="altaSolicitudDeEnvioResponse">
<part name="parameters" element="tns:altaSolicitudDeEnvioResponse"/>
</message>
<portType name="WSEnvios">
<operation name="remove">
ESP
05:35 p. m.