

Práctica 1b: Arquitectura de Jakarta EE (Segunda parte)

Cuestión número 1:

Abrir el archivo VotoDAORemote.java y comprobar la definición de dicha interfaz. Anote en la memoria comentarios sobre las librerías Jakarta EE importadas y las anotaciones utilizadas. ¿Para qué se utilizan? Comparar esta interfaz con el fichero WSDL de configuración del web service implementado en la práctica P1A.

Podemos apreciar que la librería importada de Jakarta EE es 'jakarta.ejb.Remote' que importamos para poder utilizar la anotación '@Remote'. Esto marca que la interfaz a la que se lo añadimos, se convierte en interfaz remota, es decir, que puede ser accedida desde un cliente remoto. A diferencia del fichero WSDL implementado en la práctica P1A, esta es una interfaz de java que define métodos de acceso a datos utilizando EJB. Estos métodos proporcionan funcionalidades similares, pero se utilizan en el contexto de una aplicación Java y no en un Web service.

Ejercicio número 1:

```
import jakarta.ejb.Stateless;
import ssii2.servicio.dao.VotoDAORemote;

/* @WebService(): indica que la clase Java implementa un servicio web. */
/* @WebMethod(): /* indica que el método Java que le sigue será exportado como un m
público del servicio. */

/**
 * @author jaime, daniel
 */

@Stateless
public class VotoDAOBean extends DBTester implements VotoDAORemote {
```

Primero, a la clase VotoDAOBean le ponemos la anotación Stateless para convertir la clase en una interfaz remota e importamos jakarta.ejb.Stateless para poder poner la anotación y ssii2.servicio.dao.VotoDAORemote para implementar esta interfaz.

```

public VotoBean[] getVotos(String idProcesoElectoral) {

    PreparedStatement pstmt = null;
    Connection pcon = null;
    ResultSet rs = null;
    VotoBean[] ret = null;
    ArrayList<VotoBean> votos = null;
    String qry = null;

    try {

        // Crear una conexion u obtenerla del pool
        pcon = getConnection();
        qry = SELECT_VOTOS_QRY;
        errorLog(qry + "[idProcesoElectoral=" + idProcesoElectoral + "]");

        // La preparacion del statement
        // es automaticamente tomada de un pool en caso
        // de que ya haya sido preparada con anterioridad

        pstmt = pcon.prepareStatement(qry);
        pstmt.setString(1, idProcesoElectoral);
        rs = pstmt.executeQuery();

        votos = new ArrayList<VotoBean>();

        while (rs.next()) {
            CensoBean c = new CensoBean();
            VotoBean v = new VotoBean();
            v.setIdVoto(rs.getString("idVoto"));
            v.setIdCircunscripcion(rs.getString("idCircunscripcion"));
            v.setIdMesaElectoral(rs.getString("idMesaElectoral"));
            v.setIdProcesoElectoral(rs.getString("idProcesoElectoral"));
            v.setNombreCandidatoVotado(rs.getString("nombreCandidatoVotado"));
            v.setMarcaTiempo(rs.getString("marcaTiempo"));
            v.setCodigoRespuesta(rs.getString("codRespuesta"));

            votos.add(v);
        }

        ret = new VotoBean[ votos.size() ];
        ret = votos.toArray(ret);

        // Cerramos / devolvemos la conexion al pool

        pcon.close();
    } catch (Exception e) {
        errorLog(e.toString());
    } finally {
        try {
            if (rs != null) {
                rs.close(); rs = null;
            }
            if (pstmt != null) {
                pstmt.close(); pstmt = null;
            }
            if (pcon != null) {
                closeConnection(pcon); pcon = null;
            }
        } catch (SQLException e) {
        }
    }
}

return ret;
}

```

Dado que el método ‘getVotos’ de ‘VotoDAORemote’ devuelve un array de votos, hemos tenido que cambiar la implementación de este para que tenga una adecuada nomenclatura.

Ejercicio número 2:

Para este ejercicio, primero se han añadido las importaciones necesarias y se eliminan las que ya no se utilizan.

```
import jakarta.ejb.EJB;
import ssii2.servicio.dao.VotoDAORemote;
```

Se añade como atributo de la clase el objeto proxy que permite acceder al EJB remoto y posteriormente se declara.

```
@EJB(name = "VotoDAOBean", beanInterface = VotoDAORemote.class)
private VotoDAORemote dao;
```

Se modifica la siguiente línea debido al cambio de nomenclatura del método remoto ‘getVotos()’

```
// Llamada al método del servicio web para consultar votos
ssii2.servicio.VotoBean[] votos = dao.getVotos(this.voto.getIdProcesoElectoral());
```

Finalmente se llevan a cabo las modificaciones necesarias para que al compilar el cliente este no de error. Además se han eliminado todas las declaraciones de la instancia del antiguo webservice en todos los métodos de ‘controladorBean’. Además se deben añadir importaciones en VotoDAORemote:

Ejercicio número 3:

Se ha creado el fichero glassfish-web.xml en la ruta web/WEB-INF y se ha especificado la ip donde se desplegará el servidor de la aplicación.

```
web > WEB-INF > glassfish-web.xml
1  <?xml version="1.0" encoding="UTF-8"?>
2  <!DOCTYPE glassfish-web-app PUBLIC "-//GlassFish.org//DTD GlassFish Application Server 3.1 Servlet 3.0//EN" "http://glassfish.org/dtds/glassfish-web-app_3_0-1.dtd">
3  <glassfish-web-app> <ejb-ref> <jndi-name>VotoDAOBean</jndi-name>
4  <jndi-name>corbaname:iiop:10.4.1.1:3700#java:global/P1-ejb-servidor/VotoDAOBean!ssii2.servicio.dao.VotoDAORemote</jndi-name> </ejb-ref>
5  </glassfish-web-app>
```

Tras esto, podemos empaquetar el cliente generando el correspondiente archivo P1-ejb-cliente.war en el subdirectorio dist.

Cuestión número 2:

Abrir el archivo application.xml y explicar su contenido. Verifique el contenido de todos los archivos .jar / .war / .ear que se han construido hasta el momento (empleando el comando jar -tvf). Explique brevemente el contenido y ponga evidencias en la memoria.

El archivo application.xml contiene: primero indica que es un archivo xml, su versión (1.0) y que soporta la codificación UTF-8. Después, define el espacio de nombres con xmlns y con http://java.sun.com/xml/ns/javaee/application_5.xsd indica la localización de esquema . Finalmente define el despliegue con etiquetas como display-name (nombre de la aplicación) y module (define cada módulo).

Hemos utilizado el comando jar -tvf para comprobar el contenido de los .war y los .jar. Como sabemos, el .ear es el que contiene la aplicación completa, mientras que el .war contiene todas las clases y configuraciones necesarias del cliente, incluyendo la interfaz web. En la ruta /dist/client/:

```
enriquegf@enriquegf:~/si2/SI2P1B_2321_1/P1-ejb/dist/client$ jar -tvf P1-ejb-cliente.war
  0 Tue Mar 05 11:03:52 CET 2024 META-INF/
109 Tue Mar 05 11:03:50 CET 2024 META-INF/MANIFEST.MF
  0 Tue Mar 05 11:03:52 CET 2024 WEB-INF/
  0 Tue Mar 05 11:03:34 CET 2024 WEB-INF/classes/
  0 Tue Mar 05 11:03:34 CET 2024 WEB-INF/classes/ssii2/
  0 Tue Mar 05 11:03:34 CET 2024 WEB-INF/classes/ssii2/controlador/
  0 Tue Mar 05 11:03:34 CET 2024 WEB-INF/classes/ssii2/interaccion/
  0 Tue Mar 05 11:03:34 CET 2024 WEB-INF/classes/ssii2/servicio/
  0 Tue Mar 05 11:03:34 CET 2024 WEB-INF/classes/ssii2/servicio/dao/
  0 Tue Mar 05 11:03:34 CET 2024 WEB-INF/classes/ssii2/voto/
  0 Tue Mar 05 11:03:08 CET 2024 WEB-INF/lib/
446 Tue Mar 05 11:03:52 CET 2024 WEB-INF/ glassfish-web.xml
4029 Tue Mar 05 11:03:34 CET 2024 WEB-INF/classes/ssii2/controlador/ControladorBean.class
724 Tue Mar 05 11:03:34 CET 2024 WEB-INF/classes/ssii2/interaccion/InteraccionBean.class
763 Tue Mar 05 11:03:34 CET 2024 WEB-INF/classes/ssii2/servicio/CensoBean.class
533 Tue Mar 05 11:03:34 CET 2024 WEB-INF/classes/ssii2/servicio/dao/VotoDAORemote.class
873 Tue Mar 05 11:03:34 CET 2024 WEB-INF/classes/ssii2/voto/CensoBean.class
1554 Tue Mar 05 11:03:34 CET 2024 WEB-INF/classes/ssii2/voto/VotoBean.class
1607 Tue Mar 05 11:03:52 CET 2024 WEB-INF/web.xml
624 Tue Mar 05 11:03:52 CET 2024 borradook.xhtml
2485 Tue Mar 05 11:03:52 CET 2024 censo.xhtml
449 Tue Mar 05 11:03:52 CET 2024 error.xhtml
1228 Tue Mar 05 11:03:52 CET 2024 listadoVotos.xhtml
  795 Tue Mar 05 11:03:52 CET 2024 respuesta.xhtml
  6704 Tue Mar 05 11:03:52 CET 2024 testbd.xhtml
  2504 Tue Mar 05 11:03:52 CET 2024 voto.xhtml
```

Aquí vemos el .jar del servidor que contiene clases y configuraciones necesarias para el despliegue de la aplicación por parte del servidor. En la ruta /dist/server/:

```
enriquegf@enriquegf:~/si2/SI2P1B_2321_1/P1-ejb/dist/server$ jar -tvf P1-ejb-servidor.jar
  0 Tue Mar 05 11:03:20 CET 2024 META-INF/
109 Tue Mar 05 11:03:18 CET 2024 META-INF/MANIFEST.MF
  0 Tue Mar 05 11:03:10 CET 2024 ssii2/
  0 Tue Mar 05 11:03:10 CET 2024 ssii2/servicio/
  0 Tue Mar 05 11:03:10 CET 2024 ssii2/servicio/dao/
255 Tue Mar 05 11:03:20 CET 2024 META-INF/sun-ejb-jar.xml
  763 Tue Mar 05 11:03:10 CET 2024 ssii2/servicio/CensoBean.class
1418 Tue Mar 05 11:03:10 CET 2024 ssii2/servicio/VotoBean.class
1905 Tue Mar 05 11:03:10 CET 2024 ssii2/servicio/dao/DBTester.class
7607 Tue Mar 05 11:03:10 CET 2024 ssii2/servicio/dao/VotoDAOBean.class
  545 Tue Mar 05 11:03:10 CET 2024 ssii2/servicio/dao/VotoDAORemote.class
```

Ejercicio número 4:

```
as.host.client= 10.4.1.2
as.host.server= 10.4.1.1
```

```

postgresql.properties
1 # Propiedades de la BD postgresql
2
3 # Parametros propios de postgresql
4
5 db.name=voto
6 db.user=alumnodb
7 db.password=*****
8 db.port=5432
9 db.host=10.4.1.1
10
11 # Recursos y pools asociados
12
13 db.pool.name=VotoPool
14 db.jdbc.resource.name=jdbc/VotoDB
15 db.url=jdbc:postgresql://${db.host}:${db.port}/${db.name}
16 db.client.host=10.4.1.2
17 db.client.port=4848
18

```

```

web > WEB-INF > glassfish-web.xml
1 <?xml version="1.0" encoding="UTF-8"?>
2 <!DOCTYPE glassfish-web-app PUBLIC "-//GlassFish.org//DTD GlassFish Application Server 3.1 Servlet3.0//EN" "http://glassfish.org/dtds/glassfish-web-app_3_0-1.dtd">
3 <glassfish-web-app> <ejb-ref> <ejb-ref-name>VotoDAOBean</ejb-ref-name>
4 <jndi-name>corbaname:iiop:10.4.1.1:3700#java:global/P1-ejb-servidor/P1-ejb-servidor/VotoDAOBean!ssii2.servicio.dao.VotoDAORemote</jndi-name> </ejb-ref>
5 </glassfish-web-app>

```

Modules and Components (2)					
Module Name	Engines	Component Name	Type	Action	
P1-ejb-servidor.jar	[ejb, weld]	VotoDAOBean	StatelessSessionBean		
P1-ejb-servidor.jar					

Ejercicio número 5:

Ahora probaremos a votar con una persona y comprobar que se produce correctamente la inserción, consulta y borrado de votos:

← → C 10.4.1.2:8080/P1-ejb-cliente/censo.xhtml

Complete la información sobre el voto:

Id Mesa electoral:
Id Circunscripción:
Id Proceso electoral:
Nombre candidato/a votado/a:

← → C 10.4.1.2:8080/P1-ejb-cliente/voto.xhtml

Complete la información sobre el censo asociado a la persona que está votando:

Número de DNI:
Nombre y Apellidos:
Fecha de Nacimiento:
Código Autorización:

← → C 10.4.1.2:8080/P1-ejb-cliente/censo.xhtml

Información Sobre el Registro del Voto:

Código Respuesta: 000
Id Voto: 1
Id Proceso electoral: 1
Marca Tiempo Voto: 2024-03-08 00:34:40.300261
Número DNI: 83583583L

← → C 10.4.1.2:8080/P1-ejb-cliente/testbd.xhtml

Complete la información sobre el voto:

Voto:
Id Mesa electoral:
Id Circunscripción:
Id Proceso electoral:
Nombre candidato/a votado/a:

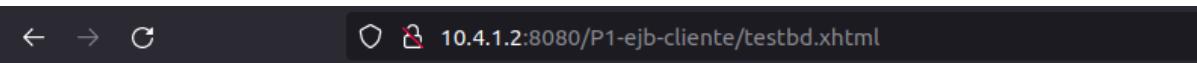
Censo:
Número de DNI:
Nombre y Apellidos:
Fecha de Nacimiento:
Código Autorización:
Debug: Si No
Prepared Statements: Si No
Conexión directa al a BD: Si No

Consulta de Votos:

Id Proceso electoral:

Borrado de Votos:

Id Proceso electoral:



Código Respuesta: 000
Id Voto: 2
Id Proceso electoral: 2
Marca Tiempo Voto: 2024-03-08 00:37:50.139859
Número DNI: 12812814Q

Tras insertar, comprobamos:

Lista de votos del proceso electoral 2						
idVoto	idCircunscripcion	idMesaElectoral	idProcesoElectoral	Candidato/a Votado/a	Marca de Tiempo	codRespuesta
2	1	1	2	Enrique	2024-03-08 00:37:50.139859	000
Volver						

y después borramos:

Lista de votos del proceso electoral 2						
idVoto	idCircunscripcion	idMesaElectoral	idProcesoElectoral	Candidato/a Votado/a	Marca de Tiempo	codRespuesta
2	1	1	2	Enrique	2024-03-08 00:37:50.139859	000
Volver						

Lista de votos del proceso electoral 2						
idVoto	idCircunscripcion	idMesaElectoral	idProcesoElectoral	Candidato/a Votado/a	Marca de Tiempo	codRespuesta
2	1	1	2	Enrique	2024-03-08 00:37:50.139859	000
Volver						

EJERCICIO 6

Creación de las querys a utilizar:

```
private static final String SELECT_VOTOS_RESTANTES_QRY =  
    "select numeroVotosRestantes from censo " +  
    " where numeroDNI=?";  
  
private static final String UPDATE_VOTOS_RESTANTES_QRY =  
    "update censo SET numeroVotosRestantes = ? " +  
    " where numeroDNI=?";
```

La primera simplemente obtiene el contenido de la columna numeroVotosRestantes de la tabla censo para un DNI determinado

La segunda actualiza el valor de la columna numeroVotoRestantes de la tabla censo para un DNI determinado

A continuación se ha definido la función pública obtenerNumeroVotosRestantes

```
public int obtenerNumeroVotosRestantes(String numeroDNI) {  
    Connection con = null;  
    PreparedStatement pstmt = null;  
    ResultSet rs = null;  
    int numeroVotosRestantes = -1;  
  
    try {  
        con = getConnection();  
        String query = SELECT_VOTOS_RESTANTES_QRY;  
        pstmt = con.prepareStatement(query);  
        pstmt.setString(1, numeroDNI);  
        rs = pstmt.executeQuery();  
  
        if (rs.next()) {  
            numeroVotosRestantes = rs.getInt("numeroVotosRestantes");  
        }  
    } catch (SQLException e) {  
        e.printStackTrace();  
    } finally {  
        // Cerramos los recursos en orden inverso de apertura para evitar problemas de dependencia  
        try {  
            if (rs != null) {  
                rs.close();  
            }  
            if (pstmt != null) {  
                pstmt.close();  
            }  
            if (con != null) {  
                // Cerramos la conexión y manejamos cualquier excepción que pueda ocurrir  
                try {  
                    closeConnection(con);  
                } catch (SQLException ex) {  
                    ex.printStackTrace();  
                }  
            }  
        } catch (SQLException ex) {  
            ex.printStackTrace();  
        }  
    }  
    return numeroVotosRestantes;  
}
```

Siguiendo los ejemplos anteriores establece una conexión, y ejecuta el prepareStatement previamente definido para obtener el número de votos restantes para un cierto DNI, pasado como parámetro de entrada en la función..

Finalmente se define un control de errores y el cierre de la conexión y se devuelve el número de votos restantes obtenido

De forma parecida se define la función actualizarNumeroVotosRestante.

```
public void actualizarNumeroVotosRestantes(String numeroDNI, int nuevosVotosRestantes) {  
    Connection con = null;  
    PreparedStatement pstmt = null;  
  
    try {  
        con = getConnection();  
        String query = UPDATE_VOTOS_RESTANTES_QRY;  
        pstmt = con.prepareStatement(query);  
        pstmt.setInt(1, nuevosVotosRestantes);  
        pstmt.setString(2, numeroDNI);  
        pstmt.executeUpdate();  
    } catch (SQLException e) {  
        e.printStackTrace();  
    } finally {  
        // Cerramos los recursos en orden inverso de apertura para evitar problemas de dependencia  
        try {  
            if (pstmt != null) {  
                pstmt.close();  
            }  
            if (con != null) {  
                // Cerramos la conexión y manejamos cualquier excepción que pueda ocurrir  
                try {  
                    closeConnection(con);  
                } catch (SQLException ex) {  
                    ex.printStackTrace();  
                }  
            }  
        } catch (SQLException ex) {  
            ex.printStackTrace();  
        }  
    }  
}
```

La cual establece conexión, ejecuta la query anteriormente definida como prepareStatement, y para un cierto DNI se actualiza el valor de numeroVotosRestantes, por el parámetro de entrada nuevosVotosRestantes.

Finalmente se capturan los errores y se cierra la conexión

En ControladorBean se añade una nueva captura de errores de tipo EJBException de la siguiente manera, y añadiendo por supuesto el import necesario.

```
public String enviarVoto() {
    if (this.interaccion.getDebug() == true) {
        this.escribirLog(log:"Solicitado el registro del voto.");
    }
    /* Instanciamos el objeto que presta la lógica de negocio de la aplicación */

    try {
        // Traducimos la información del voto al formato del servicio web
        ssi2.servicio.VotoBean votoParaServicio = traducirVotoParaServicio(this.voto);

        // Llamamos al método registraVoto() del servicio web
        ssi2.servicio.VotoBean votoRegistrado = dao.registraVoto(votoParaServicio);

        // Actualizamos los atributos del voto con los valores devueltos por el servicio web
        this.voto.setIdVoto(votoRegistrado.getIdVoto());
        this.voto.setMarcaTiempo(votoRegistrado.getMarcaTiempo());
        this.voto.setCodigoRespuesta(votoRegistrado.getCodigoRespuesta());

        // Todo ha ido bien. Vamos a la página de respuesta
        if (this.interaccion.getDebug() == true) {
            this.escribirLog(log:"¡Voto registrado correctamente!");
        }
        return "respuesta";
    } catch (EJBException ejbEx) {
        ejbEx.printStackTrace();
        return "error";
    } catch (Exception e) {
        // Manejamos la excepción
        e.printStackTrace();
        return "error";
    }
}
```

EJERCICIO 7

	numerodni	nombre	fechanacimiento	aniocenso	codigoautorizacion	numerovotosrestantes
1	83583583L	Restituto Sparrow Martinez	31/08/73	2024	535	3
2	67867868T	Randall Martinez Mojamoto	04/06/96	2024	056	3
3	12812814Q	Gabriel Ribas Pomares	30/05/54	2024	605	3
4	92992992L	Benjamin Reyes Lopez	06/05/90	2024	095	3
5	50850851I	Emiliano Reyes Gracia	16/11/81	2024	428	3
6	47047047F	Luis Mas Moreno	27/08/54	2024	620	3
7	29829830H	Emiliano Locke Gracia	08/05/60	2024	796	3
8	26926927J	Camilo Marques Gibson	28/05/94	2024	716	3
9	97797797K	Eva Ribera Pelaez	24/04/51	2024	697	3
10	18618619K	Armando Linus Ribera	28/04/89	2024	424	3
11	30630631P	Luisa Linus Coll	13/02/00	2024	090	3
12	59659659J	Alberto Martinez Poza	28/09/93	2024	466	3
13	27627628Z	Jack Garau Morales	17/05/89	2024	947	3
14	87387387L	Kate Moss Sparrow	28/12/92	2024	853	3
15	94994994D	Sofia Poza Gracia	06/07/76	2024	104	3
16	49349349T	Jack Punset Gonzalez	11/01/59	2024	794	3
17	32932933O	Petra Marques Pelaez	09/06/73	2024	599	3
18	77477477X	Gonzalo Locke Ribas	20/12/78	2024	594	3
19	84084083M	Jose Punset Pomares	19/01/66	2024	194	3
20	59059059U	Benjamin Reyes Coll	17/06/61	2024	231	3
21	72472472B	Jack San Moss	15/02/61	2024	839	3
22	03603605D	Eva Mas Moreno	17/04/57	2024	242	3
23	10410411O	Kate San Gibson	31/05/82	2024	142	3
24	72872872Q	Alberto Mas Avila	16/07/95	2024	462	3
25	87787787O	Kate San Moss	22/07/52	2024	587	3
26	48448448J	Pedro Moreno Ribera	26/10/69	2024	572	3
27	67667667L	Enjuto San Dominguez	11/04/50	2024	793	3

Como podemos observar, tras las modificaciones, se ha añadido correctamente una nueva columna al censo con la información de los números de votos restantes de cada persona

Ahora probaremos a votar con una persona y comprobar que se produce una reducción de su columna "numerovotosrestantes"

	Id numerodni	Nombre	Fecha nacimiento	Anio censo	Código autorización	Numerovotosrestantes
1	83583583L	Restituta Sparrow Martinez	31/08/73	2024	535	3

10.4.1.3:8080/P1-ejb-cliente/borradook.xhtml;jsessionid=de5b1c462894f4ccbbc81584712f

Complete la información sobre el voto:

Voto:

Id Mesa electoral:
 Id Circunscripción:
 Id Proceso electoral:
 Nombre candidato/a votado/a:

Censo:

Número de DNI:
 Nombre y Apellidos:
 Fecha de Nacimiento:
 Código Autorización:
 Debug: Si No
 Prepared Statements: Si No
 Conexión directa al a BD: Si No

10.4.1.3:8080/P1-ejb-cliente/testbd.xhtml;jsessionid=de5b1c462894f4ccbbc81584712f

Información Sobre el Registro del Voto:

Código Respuesta: 000
 Id Voto: 3
 Id Proceso electoral: 1
 Marca Tiempo Voto: 2024-03-08 12:57:20.659328
 Número DNI: 83583583L

y efectivamente se ha modificado el campo

	Restituta Sparrow Martinez	31/08/73	2024	535	2
1	Restituta Sparrow Martinez	31/08/73	2024	535	2

A continuación realizaremos un voto duplicado para la misma persona en el mismo proceso electoral

Complete la información sobre el voto:

Voto:

Id Mesa electoral:
 Id Circunscripción:
 Id Proceso electoral:
 Nombre candidato/a votado/a:

Censo:

Número de DNI:
 Nombre y Apellidos:
 Fecha de Nacimiento:
 Código Autorización:
 Debug: Si No
 Prepared Statements: Si No
 Conexión directa al a BD: Si No

Votamos lo siguiente de nuevo y se muestra lo siguiente:

Se ha producido el siguiente error:

Y como podemos comprobar no se ha modificado su campo de votos restantes:

83583583L Restituta Sparrow Martinez 31/08/73 2024 535 2

Y no se ha generado un nuevo voto en la tabla voto

Lista de votos del proceso electoral 1

idVoto	idCircunscripcion	idMesaElectoral	idProcesoElectoral	Candidato/a Votado/a	Marca de Tiempo	codRespuesta
3	1	1	1	daniel	2024-03-08 12:57:20.659328	000

[Volver](#)

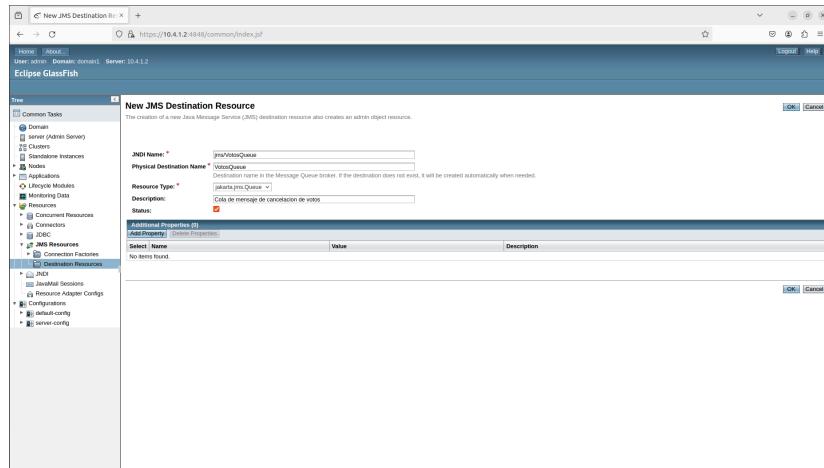
Ejercicio 8

Creación connection factory de forma manual correctamente:

The screenshot shows the Eclipse GlassFish administration interface. On the left, there's a tree view of the server configuration. In the center, a dialog box titled "New JMS Connection Factory" is open. It contains two main sections: "General Settings" and "Pool Settings". Under "General Settings", the "JNDI Name" is set to "jms\VotoConnectionFactory" and the "Resource Type" is "jakarta.jms.QueueConnectionFactory". The "Description" field contains "Factoría de conexiones de la cola de votos" and the "Status" is "Active". Under "Pool Settings", the "Initial and Minimum Pool Size" is set to 8, "Maximum Pool Size" is 32, and "Pool Resize Quantity" is 2. There are also fields for "Idle Timeout" (300 seconds), "Max Wait Time" (60000 milliseconds), and "On Any Failure" (checkbox for "Close All Connections"). The "Transaction Support" dropdown is set to "Required". At the bottom, there's an "Additional Properties" section with an "Add Property" button and a table for setting properties like "Name", "Value", and "Description". The table shows one entry: "Select | Name | Value | Description" with the note "No items found".

Ejercicio 9

Creación cola de mensajes de forma manual correctamente:



Ejercicio 10:

Añadimos el nombre de la connectionfactory al archivo sun-ejb-jar.xml:

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE sun-ejb-jar PUBLIC "-//Sun Microsystems, Inc.//DTD Application Server 9.0 EJB 3.0//EN" "http://www.sun.com/software/appserver/dtds/sun-ejb-jar_3_0-0.dtd">
<sun-ejb-jar>
    <enterprise-beans>
        <ejb>
            <ejb-name>VotoCancelacionJMSBean</ejb-name>
            <mdb-connection-factory>
                <!-- TODO - definir el nombre de la connection factory -->
                <jndi-name>jms/VotoConnectionFactory</jndi-name>
            </mdb-connection-factory>
        </ejb>
    </enterprise-beans>
</sun-ejb-jar>
```

Creamos las 3 consultas sql exigidas para obtener el código de respuesta del voto con un idVoto dado, para actualizar el código de respuesta a valor 999 el voto con un idVoto dado y para rectificar el número de votos restante de la persona que registró el voto:

```
private static final String SELECT_CODIGO_RESPUESTA =
    "SELECT codRespuesta " +
    "FROM voto " +
    "WHERE idVoto = ?";

private static final String UPDATE_CODIGO_RESPUESTA =
    "UPDATE voto " +
    "SET codRespuesta = 999 " +
    "WHERE idVoto = ?";

private static final String UPDATE_NUMERO_VOTOS_RESTANTES =
    "UPDATE censo " +
    "SET numeroVotosRestantes = numeroVotosRestantes + 1 " +
    "WHERE numeroDNI = (SELECT numeroDNI FROM voto WHERE idVoto = ?);
```

Tras esto se modifica el método onMessage que obtiene el idVoto de la cola de mensajes, comprueba si el voto con dicho idVoto tiene código de respuesta 000 y en ese caso actualice el código de respuesta a 999 y rectifique el número de votos restantes. Todo esto se ha hecho con los prepared statements como vimos en ejercicios anteriores. Finalmente cerramos la conexión. El control de errores se hace mediante el try/catch que obtiene y devuelve el error si existe.

```

    // la desactualización
    public void onMessage(Message inMessage) {
        TextMessage msg = null;

        try {
            if (inMessage instanceof TextMessage) {
                msg = (TextMessage) inMessage;
                logger.info("MESSAGE BEAN: Message received: " + msg.getText());
                String idVoto = msg.getText();

                Connection con = getConnection();

                PreparedStatement selectStmt = con.prepareStatement(SELECT_CODIGO_RESPUESTA);
                selectStmt.setInt(1, Integer.parseInt(idVoto));
                ResultSet rs = selectStmt.executeQuery();
                String codRespuesta = "";
                if (rs.next()) {
                    codRespuesta = rs.getString("codRespuesta");
                }
                rs.close();
                selectStmt.close();
                if (codRespuesta.equals("000")) {
                    PreparedStatement updateRespStmt = con.prepareStatement(UPDATE_CODIGO_RESPUESTA);
                    updateRespStmt.setInt(1, Integer.parseInt(idVoto));
                    updateRespStmt.executeUpdate();
                    updateRespStmt.close();

                    PreparedStatement updateVotosStmt = con.prepareStatement(UPDATE_NUMERO_VOTOS_RESTANTES);
                    updateVotosStmt.setInt(1, Integer.parseInt(idVoto));
                    updateVotosStmt.executeUpdate();
                    updateVotosStmt.close();
                }
                con.close();
            } else {
                logger.warning(
                    "Message of wrong type: "
                    + inMessage.getClass().getName());
            }
        } catch (JMSException e) {
            e.printStackTrace();
            mdc.setRollbackOnly();
        } catch (Throwable te) {
            te.printStackTrace();
        }
    }
}

ava: Ready

```

Ejercicio 11:

Para la preparación del cliente se pueden incluir tanto recursos JMS estáticos, que pueden aportar un mayor rendimiento ya que no tiene que hacer búsquedas dinámicas en tiempos de ejecución.

```

@Resource(mappedName = "jms/VotoConnectionFactory")
private static ConnectionFactory connectionFactory;

@Resource(mappedName = "jms/VotosQueue")
private static Queue queue;

```

o dinámicos, cuya ventaja es que no es necesario que conozcamos el nombre en tiempo de compilación, mientras que en las estáticas si.

```

//InitialContext jndi = new InitialContext();
//connectionFactory = (ConnectionFactory)jndi.lookup("jms/VotoConnectionFactory");
//queue = (Queue)jndi.lookup("jms/VotosQueue");

```

Ejercicio 12:

Añadimos en los archivos jsm.properties y build.properties los valores correctos de la connectionfactory name y queue name para que los puedan utilizar correctamente los archivos build.xml y jms.xml y poder crear ambas. También ponemos las ips necesarias para que accedan al servidor del admin y poder realizar las operaciones.

```
jms.factoryname=jms/VotoConnectionFactory
jms.name=jms/VotosQueue
jms.physname=Voto
```

```
as.host.server=10.4.1.2
```

```
as.host.mdb=10.4.1.2
```

Tras borrar la connection factory y la cola creadas manualmente y ejecutar en la terminal ant todo, vemos cómo se han vuelto a crear correctamente.

The screenshot shows two pages from the Eclipse GlassFish Admin Console:

- JMS Connection Factories:** This page lists two connection factories:

Select	JNDI Name	Logical JNDI Name	Enabled	Resource Type	Description
<input type="checkbox"/>	jms/_defaultConnectionFactory	java.comp/DefaultJMSConnectionFactory	✓	jakarta.jms.ConnectionFactory	
<input type="checkbox"/>	jms/VotoConnectionFactory		✓	jakarta.jms.QueueConnectionFactory	
- JMS Destination Resources:** This page lists one destination resource:

Select	JNDI Name	Enabled	Resource Type	Description
<input type="checkbox"/>	jms/VotosQueue	✓	jakarta.jms.Queue	

En el fichero jms.xml podemos observar lo siguiente:

```
<!-- ... -->
<target name="create-jms-resource"
      description="creates jms destination resource">
    <exec executable="${asadmin}">
        <arg line="--user ${as.user}" />
        <arg line="--passwordfile ${as.passwordfile}" />
        <arg line="--host ${as.host.server}" />
        <arg line="--port ${as.port}" />
        <arg line="create-jms-resource"/>
        <arg line="--restype ${jms.restype}" />
        <arg line="--enabled=true" />
        <arg line="--property ${jms.resource.property}" />
        <arg line="${jms.resource.name}" />
    </exec>
</target>

<target name="create-jms-connection-factory"
      description="creates jms connection factory">
    <exec executable="${asadmin}">
        <arg line="--user ${as.user}" />
        <arg line="--passwordfile ${as.passwordfile}" />
        <arg line="--host ${as.host.server}" />
        <arg line="--port ${as.port}" />
        <arg line="create-jms-resource"/>
        <arg line="--restype ${jms.restype}" />
        <arg line="--enabled=true" />
        <arg line="${jms.resource.name}" />
    </exec>
</target>
```

Estos comandos se ejecutan desde asadmin creando los recursos jms. Los comandos equivalentes son:

```
asadmin --user admin --passwordfile passwordfile --host 10.4.1.2 --port 4848
create-jms-resource --restype jakarta.jms.QueueConnectionFactory
--enabled=true --property Name=Voto jms/VotosQueue
```

```
asadmin --user admin --passwordfile passwordfile --host 10.4.1.2 --port 4848
create-jms-resource --restype jakarta.jms.QueueConnectionFactory
--enabled=true --property Name=Voto jms/VotoConnectionFactory
```

Ejercicio 13

El código modificado es el siguiente, donde se establece la correcta conexión y envío del mensaje a través de la cola, verificando la referencia en el apéndice.

```
public static void main(String[] args) {
    Connection connection = null;
    Session session = null;
    MessageProducer messageProducer = null;
    TextMessage message = null;

    if (args.length != 1) {
        System.err.println("Uso: VotoQueueMessageProducer [-browse | <msg>]");
        return;
    }

    try {
        // Inicializar connectionFactory y queue mediante JNDI
        InitialContext jndi = new InitialContext();
        connectionFactory = (ConnectionFactory) jndi.lookup("jms/VotoConnectionFactory");
        queue = (Queue) jndi.lookup("jms/VotosQueue");

        connection = connectionFactory.createConnection();
        session = connection.createSession(false, Session.AUTO_ACKNOWLEDGE);
        if (args[0].equals("-browse")) {
            browseMessages(session);
        } else {
            // Crear un productor de mensajes para la cola
            messageProducer = session.createProducer(queue);

            // Crear un mensaje de texto y establecer su contenido
            message = session.createTextMessage();
            message.setText(args[0]);

            // Enviar el mensaje a la cola
            messageProducer.send(message);
            System.out.println("Mensaje enviado correctamente: " + args[0]);
        }
    } catch (Exception e) {
        System.out.println("Excepcion : " + e.toString());
    } finally {
        if (connection != null) {
            try {
                connection.close();
            } catch (JMSEException e) {
                e.printStackTrace();
            }
        }
        System.exit(0);
    }
}

} // class
```

Configuración de JMS Host

Edit JMS Host

The Java Message Service (JMS) host specifies the system where the JMS service is running.

[Load Defaults](#)

Configuration Name: server-config

Name:	default_JMS_host
Host:	10.4.1.2
Name or IP address; if name, must contain only alphanumeric, underscore, dash, or dot characters	
Port:	\$(JMS_PROVIDER_PORT)
Listener port for servicing JMS requests	
Admin Username: *	admin
User name for maintaining the JMS service; can be up to 255 characters, must contain only alphanumeric, underscore, dash, or dot characters	
Admin Password: *	*****
Password for JMS administrator	
Confirm New Password: *	*****

Establecemos el host a 10.4.1.2

Detener la ejecución de mdb, y verificamos que efectivamente los mensajes se introducen en la cola, y al estar desactivado mdb, no se consumen los mensajes, por lo tanto se quedan esperando a ser leídos en ella

Applications

Applications can be enterprise or web applications, or various kinds of modules. Restart an application or module by clicking on the reload link, this action will apply only to the targets that the application or module is enabled on.

Deployed Applications (3)							
Select	Name	Deployment Order	Enabled	Engines	Action		
<input type="checkbox"/>	P1-ejb-cliente	100	✓	web	Launch Redeploy Reload		
<input type="checkbox"/>	P1-ejb-servidor	100	✓	ear, ejb	Redeploy Reload		
<input type="checkbox"/>	P1-jms-mdb	100	✗	ejb	Redeploy Reload		

Total time: 3 seconds

```
~/Doc/i/t/s/p/SI2P1B_2321_1/P1-jms ~/glassfish7/glassfish/bin/appclient -targetserver 10.4.1.2 -client dist/clientjms/P1-jms-clientjms.jar 1
Mar 10, 2024 5:04:26 PM org.glassfish.enterprise.iiop.impl.GlassFishORBManager getORB
INFO: ORB initialization succeeded: com.sun.corba.ee.impl.orb.ORBImpl@76396509
Mar 10, 2024 5:04:26 PM org.hibernate.validator.internal.util.Version <clinit>
INFO: HV000001: Hibernate Validator 8.0.1.Final
Mar 10, 2024 5:04:26 PM com.sun.messaging.jms.ra.ResourceAdapter start
INFO: MQJMSRA_RA1101: Eclipse OpenMQ JMS Resource Adapter: Version: 6.4.0 (Build 1-a) Compile: February 25 2023
Mar 10, 2024 5:04:26 PM com.sun.messaging.jms.ra.ResourceAdapter start
INFO: MQJMSRA_RA1101: Eclipse OpenMQ JMS Resource Adapter starting: broker is REMOTE, connection mode is TCP
Mar 10, 2024 5:04:26 PM com.sun.messaging.jms.ra.ResourceAdapter start
INFO: MQJMSRA_RA1101: Eclipse OpenMQ JMS Resource Adapter Started:REMOTE
Mar 10, 2024 5:04:26 PM com.sun.enterprise.resource.pool.PoolManagerImpl createAndInitPool
INFO: Created connection pool and added it to PoolManager: Pool [org.glassfish.resourcebase.resources.api.PoolInfo@af8b3cd6[jndiName=jms/VotoConnectionFactory-Connection-Pool, applicationName=null, moduleName=null]] PoolSize=0 FreeResources=0 QueueSize=0 matching=on validation=off
Mensaje enviado correctamente: 1
```

```

q ~ ~/Doc/i/t/s/p/SI2P1B_2321_1/P1-jms ~/glassfish7/glassfish/bin/appclient -targetserver 10.4
.1.2 -client dist/clientjms/P1-jms-clientjms.jar -browse
Mar 10, 2024 5:05:00 PM org.glassfish.enterprise.iiop.impl.GlassFishORBManager getORB
INFO: ORB initialization succeeded: com.sun.corba.ee.impl.orb.ORBImpl@69ec93c2
Mar 10, 2024 5:05:01 PM org.hibernate.validator.internal.util.Version <clinit>
INFO: HV000001: Hibernate Validator 8.0.1.Final
Mar 10, 2024 5:05:01 PM com.sun.messaging.jms.ra.ResourceAdapter start
INFO: MQJMSRA_RA1101: Eclipse OpenMQ JMS Resource Adapter: Version: 6.4.0 (Build 1-a) Compile: February
25 2023
Mar 10, 2024 5:05:01 PM com.sun.messaging.jms.ra.ResourceAdapter start
INFO: MQJMSRA_RA1101: Eclipse OpenMQ JMS Resource Adapter starting: broker is REMOTE, connection mode is T
CP
Mar 10, 2024 5:05:01 PM com.sun.messaging.jms.ra.ResourceAdapter start
INFO: MQJMSRA_RA1101: Eclipse OpenMQ JMS Resource Adapter Started:REMOTE
Mar 10, 2024 5:05:01 PM com.sun.enterprise.resource.pool.PoolManagerImpl createAndInitPool
INFO: Created connection pool and added it to PoolManager: Pool [org.glassfish.resourcebase.resources.api.
PoolInfo@af8b3cd6[jndiName=jms/VotoConnectionFactory-Connection-Pool, applicationName=null, moduleName=nul
l]] PoolSize=0 FreeResources=0 QueueSize=0 matching=on validation=off
Mensajes en cola:
1

```

A continuació, volvemos a activarlo,

Applications

Applications can be enterprise or web applications, or various kinds of modules. Restart an application or module by clicking on the reload link, this action will apply only to the targets that the application or module is enabled on.

Deployed Applications (3)						
Select	Name	Deployment Order	Enabled	Engines	Action	
<input type="checkbox"/>	P1-ejb-cliente	100	✓	web	Launch Redeploy Reload	
<input type="checkbox"/>	P1-ejb-servidor	100	✓	ear, ejb	 Redeploy Reload	
<input type="checkbox"/>	P1-jms-mdb	100	✓	ejb	 Redeploy Reload	

Introducimos un nuevo voto con votoid = 3 con ID Proceso Electoral = 1

Complete la información sobre el voto:

Voto:

Id Mesa electoral:	<input type="text" value="1"/>
Id Circunscripción:	<input type="text" value="1"/>
Id Proceso electoral:	<input type="text" value="1"/>
Nombre candidato/a votado/a:	<input type="text" value="daniel"/>

Censo:

Número de DNI:	<input type="text" value="83583583L"/>
Nombre y Apellidos:	<input type="text" value="Restitura Sparrow Martinez"/>
Fecha de Nacimiento:	<input type="text" value="31/08/73"/>
Código Autorización:	<input type="text" value="535"/>
Debug:	<input type="radio"/> Si <input type="radio"/> No
Prepared Statements:	<input type="radio"/> Si <input type="radio"/> No
Conexión directa al a BD:	<input type="radio"/> Si <input type="radio"/> No

[Registrar Voto](#)

Lista de votos del proceso electoral 1

idVoto	idCircunscripcion	idMesaElectoral	idProcesoElectoral	Candidato/a Votado/a	Marca de Tiempo	codRespuesta
3	1	1	1	daniel	2024-03-10 20:26:56.338188	000

[Volver](#)

Y verificamos que se ha realizado la votación

```
~ Doc/i/t/s/s/p/SI2P1B_2321_1/P1-jms ~glassfish7/glassfish/bin/appclient -targetserver 10.4.1.2 -client dist/clientjms/P1-jms-clientjms.jar 3 Mar 10, 2024 8:27:23 PM org.glassfish.enterprise.iiop.impl.GlassFishORBManager getORB INFO: ORB initialization succeeded: com.sun.corba.ee.impl.orb.ORBImpl@29d25e65 Mar 10, 2024 8:27:23 PM org.hibernate.validator.internal.util.Version <clinit> INFO: HV000001: Hibernate Validator 8.0.1.Final Mar 10, 2024 8:27:23 PM com.sun.messaging.jms.ra.ResourceAdapter start INFO: MQJMSRA_RA1101: Eclipse OpenMQ JMS Resource Adapter: Version: 6.4.0 (Build 1-a) Compile: February 25 2023 Mar 10, 2024 8:27:23 PM com.sun.messaging.jms.ra.ResourceAdapter start INFO: MQJMSRA_RA1101: Eclipse OpenMQ JMS Resource Adapter starting: broker is REMOTE, connection mode is TCP Mar 10, 2024 8:27:23 PM com.sun.messaging.jms.ra.ResourceAdapter start INFO: MQJMSRA_RA1101: Eclipse OpenMQ JMS Resource Adapter Started:REMOTE Mar 10, 2024 8:27:23 PM com.sun.enterprise.resource.pool.PoolManagerImpl createAndInitPool INFO: Created connection pool and added it to PoolManager: Pool [org.glassfish.resourcebase.resources.api.PoolInfo@af8b3cd6[jndiName=jms/VotoConnectionFactory-Connection-Pool, applicationName=null, moduleName=null]] PoolSize=0 FreeResources=0 QueueSize=0 matching=on validation=off Mensaje enviado correctamente: 3
```

Enviamos el mensaje de cancelación para el voto con ID = 3

```
~ Doc/i/t/s/s/p/SI2P1B_2321_1/P1-jms ~glassfish7/glassfish/bin/appclient -targetserver 10.4.1.2 -client dist/clientjms/P1-jms-clientjms.jar -browse Mar 10, 2024 8:28:01 PM org.glassfish.enterprise.iiop.impl.GlassFishORBManager getORB INFO: ORB initialization succeeded: com.sun.corba.ee.impl.orb.ORBImpl@1788cb61 Mar 10, 2024 8:28:02 PM org.hibernate.validator.internal.util.Version <clinit> INFO: HV000001: Hibernate Validator 8.0.1.Final Mar 10, 2024 8:28:02 PM com.sun.messaging.jms.ra.ResourceAdapter start INFO: MQJMSRA_RA1101: Eclipse OpenMQ JMS Resource Adapter: Version: 6.4.0 (Build 1-a) Compile: February 25 2023 Mar 10, 2024 8:28:02 PM com.sun.messaging.jms.ra.ResourceAdapter start INFO: MQJMSRA_RA1101: Eclipse OpenMQ JMS Resource Adapter starting: broker is REMOTE, connection mode is TCP Mar 10, 2024 8:28:02 PM com.sun.messaging.jms.ra.ResourceAdapter start INFO: MQJMSRA_RA1101: Eclipse OpenMQ JMS Resource Adapter Started:REMOTE Mar 10, 2024 8:28:02 PM com.sun.enterprise.resource.pool.PoolManagerImpl createAndInitPool INFO: Created connection pool and added it to PoolManager: Pool [org.glassfish.resourcebase.resources.api.PoolInfo@af8b3cd6[jndiName=jms/VotoConnectionFactory-Connection-Pool, applicationName=null, moduleName=null]] PoolSize=0 FreeResources=0 QueueSize=0 matching=on validation=off Cola de mensajes vacía!
```

Y observamos que no hay mensajes en la cola de mensajes, es decir, el mensaje ha sido leído.

Volvemos a revisar el estado del voto y encontramos que se ha completado correctamente la cancelación de dicho voto

Lista de votos del proceso electoral 1

idVoto	idCircunscripcion	idMesaElectoral	idProcesoElectoral	Candidato/a Votado/a	Marca de Tiempo	codRespuesta
3	1	1	1	daniel	2024-03-10 20:26:56.338188	999

[Volver](#)

El código de respuesta del voto ha cambiado a "999" y Restituta continúa con la columna de numeroVotosRestantes = 3

83583583L	Restituta Sparrow Martinez	31/08/73	2024	535	3
-----------	----------------------------	----------	------	-----	---

Para una verificación más precisa ejecutaremos las querys necesarias desde la base de datos para acceder a estos valores

idvoto	idcircunscripcion	idmesaelectoral	idprocesoelectoral	nombre candidatovotado	codrespuesta	numerodni	marcatiempo
3	1	1	1	daniel	999	83583583L	2024-03-10 20:26:56.338188

numerodni	nombre	fechanacimiento	aniocenso	codigoautorizacion	numerovotosrestantes	
83583583L	Restituta Sparrow Martinez	31/08/73	2024	535		3