

 UNIVERSIDAD AUTÓNOMA DE MADRID	Escuela Politécnica Superior Ingeniería Informática Prácticas de Sistemas Informáticos 2				
Grupo	2321	Práctica	1a	Fecha	26/02/2024
Alumno/a	Aquino, Santiago, Daniel				
Alumno/a	Gómez, Fernández, Enrique				

Práctica 1: Arquitectura de Jakarta EE (Primera parte)

Ejercicio 1.

Para realizar el ejercicio 1, primero hemos arrancado la máquina virtual y puesto en marcha el dominio de glassfish. Tras esto, hemos ejecutado el comando ant para hacer limpieza y desplegar de cero la aplicación. Antes de ello, hemos hecho los cambios necesarios en los archivos ‘build.properties’ y ‘postgresql.properties’ añadiendo los ids correctos del host servidor y cliente.

```
❸ build.properties
1  # Propiedades de despliegue de aplicacion de VOTO
2  nombre=P1
3  build=${basedir}/build
4  dist=${basedir}/dist
5  src=${basedir}/src
6  web=${basedir}/web
7  paquete=ssii2
8  war=${nombre}.war
9  asadmin=${as.home}/bin/asadmin
10 as.home=${env.JEE_HOME}
11 as.lib=${as.home}/lib
12 as.user=admin
13 as.host=10.4.1.1
14
15 as.port=4848
16 as.passwordfile=${basedir}/passwordfile
17 as.target=server
18
```

• postgresql.properties

```
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=10.4.1.1
17 db.client.port=4848
18
19 db.delimiter=;
20 db.driver=org.postgresql.Driver
21 db.datasource=org.postgresql.ds.PGConnectionPoolDataSource
22 db.vendorname=SQL92
23
24 # Herramientas
25
26 db.createdb=/usr/bin/createdb
27 db.dropdb=/usr/bin/dropdb
28
29 # Scripts de creacion / borrado
30
31 db.create.src=./sql/create.sql
32 db.insert.src=./sql/insert.sql
33 db.delete.src=./sql/drop.sql
```

```
si2@si2srv01:~$ sudo /opt/glassfish7/glassfish/bin/asadmin start-domain domain1
Waiting for domain1 to start .....
Waiting finished after 6,589 ms.
Successfully started the domain : domain1
domain Location: /opt/glassfish7/glassfish/domains/domain1
Log File: /opt/glassfish7/glassfish/domains/domain1/logs/server.log
Admin Port: 4,848
Command start-domain executed successfully.
si2@si2srv01:~$
```

```
enriquegf@enriquegf:~/Descargas/P1-base$ ant replegar limpiar-todo unsetup-db todo
Buildfile: /home/enriquegf/Descargas/P1-base/build.xml

replegar:
    [exec] Command undeploy executed successfully.

limpiar-todo:
    [delete] Deleting directory /home/enriquegf/Descargas/P1-base/build
    [delete] Deleting directory /home/enriquegf/Descargas/P1-base/dist

delete-resource-local:
    [echo] Deleting jdbc resource jdbc/VotoDB

delete-jdbc-resource:
    [exec] JDBC resource jdbc/VotoDB deleted successfully
    [exec] Command delete-jdbc-resource executed successfully.

delete-pool-local:
    [echo] Deleting jdbc-connection-pool VotoPool.

delete-jdbc-connection-pool:
    [exec] JDBC Connection pool VotoPool deleted successfully
    [exec] Command delete-jdbc-connection-pool executed successfully.

delete-db:
    [echo] driver=org.postgresql.Driver
    [echo] url=jdbc:postgresql://10.4.1.1:5432/voto
    [echo] user=alumnodb
    [echo] password=****

unsetup-db:

todo:

create-pool-local:
    [echo] Registering jdbc-connection-pool VotoPool.
    [echo] ds=org.postgresql.ds.PGConnectionPoolDataSource

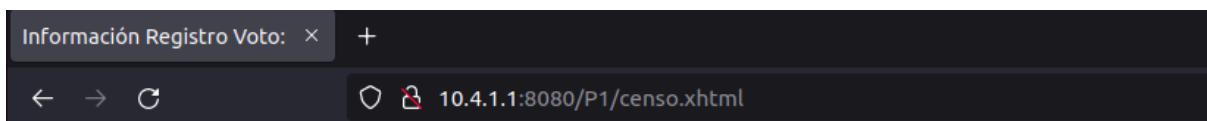
create-jdbc-connection-pool:
    [exec] JDBC connection pool VotoPool created successfully.
    [exec] Command create-jdbc-connection-pool executed successfully.

create-resource-local:
    [echo] Registering jdbc resource jdbc/VotoDB.

create-jdbc-resource:
    [exec] JDBC resource jdbc/VotoDB created successfully.
    [exec] Command create-jdbc-resource executed successfully.

delete-db:
    [echo] Deleting jdbc-connection-pool VotoPool.
```

Tras esto, con la aplicación iniciada, hemos abierto la pestaña de incógnito y hemos buscado la ruta <http://10.4.1.1:8080/P1>, a lo que nos apareció una página para registrar un voto. Tras registrarla, nos apareció el resultado exitoso de haberlo realizado:



Información Sobre el Registro del Voto:

Código Respuesta: 000
 Id Voto: 1
 Id Proceso electoral: 1
 Marca Tiempo Voto: 2024-02-19 20:10:00.636922
 Número DNI: 23424434Y

Podemos comprobar en Dbeaver que el voto ha sido registrado en la base de datos correctamente:

idvoto	idcircunscripcion	idmesaelectoral	idprocesoelectoral	nombrecandidatovotado	codrespuesta	numerodni	marcatiempo
1	1	1	1	Jorge Paniagua	000	23424434Y	2024-02-19 20:10:00.636

Tras esto, copiamos la siguiente ruta,

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:

Consultamos el voto:

idVoto	idCircunscripcion	idMesaElectoral	idProcesoElectoral	Candidato/a Votado/a	Marca de Tiempo	codRespuesta
1	1	1	1	Jorge Paniagua	2024-02-19 20:10:00.636922	000

Lo borramos:

Se han borrado correctamente 1 votos del proceso electoral con identificador 1
<input type="button" value="Volver"/>

Y comprobamos tanto en la web como en dbeaver:

The top part of the image shows a web browser window with the URL 10.4.1.1:8080/P1/testbd.xhtml. The page displays a table titled "Lista de votos del proceso electoral 1" with columns: idVoto, idCircunscripcion, idMesaElectoral, idProcesoElectoral, Candidato/a Votado/a, Marca de Tiempo, and codRespuesta. Below the table is a "Volver" button.

The bottom part of the image shows the DBeaver 23.3.5 interface connected to a database named "voto". The tool bar includes Archivo, Editar, Navegar, Buscar, Editor SQL, Base de Datos, Ventana, Ayuda, and various icons. The main pane shows the database structure with tables: voto, public, censo, and voto. The "voto" table is selected, and its properties, data, and diagram are visible. A grid view shows the table's structure with columns: idvoto, idcircunscripcion, idmesaelectoral, idprocesoelectoral, nombrecandidatovotado, codrespuesta, numerodni, and marcatiempo.

Ejercicio 2.

Como nos indica en el enunciado, miramos el apéndice 10 y seguimos las instrucciones para realizar la conexión directa cambiando el archivo 'DBTester.java':

```
/*
public class DBTester {

    // Información de conexión
    // Para conexiones directas, requerimos: driver, cadena de conexión,
    // usuario y clave
    private static final String JDBC_DRIVER =
        "org.postgresql.Driver";

    // TODO: Definir la cadena de conexión a la base de datos
    // *****
    private static final String JDBC_CONNSTRING =
        "jdbc:postgresql://10.1.1.1:5432/voto;create=true";
    // *****
    private static final String JDBC_USER = "alumnodb";
    private static final String JDBC_PASSWORD = "*****";
```

Tras esto, para aplicar los cambios, primero hacemos reboot de la VM, y posteriormente replegamos con 'ant replegar' y volvemos a lanzar con 'ant todo'. Tras esto, ya está desplegada la aplicación con las nuevas modificaciones y podemos comprobar los cambios, y vemos como con conexión directa, funciona todo, registro, consulta y borrado de voto. Lo comprobamos en la página y en dbeaver:

Aplicacion de Voto Electrónico x +

← → C 10.4.1.1:8080/P1/borradook.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:

Información Registro Voto:

Código Respuesta: 000
 Id Voto: 5
 Id Proceso electoral: 1
 Marca Tiempo Voto: 2024-02-19 23:20:28.223954
 Número DNI: 39739740E

Página Listado de Votos

Lista de votos del proceso electoral 1

idVoto	idCircunscripcion	idMesaElectoral	idProcesoElectoral	Candidato/a Votado/a	Marca de Tiempo	codRespuesta
5	1	1	1	Jorge Paniagui	2024-02-19 23:20:28.223954	000

voto	<voto>	Script	voto	public	censo	voto	x
Propiedades	Datos	Diagrama ER					
voto <input type="text" value="Enter a SQL expression to filter results (use Ctrl+Space)"/>							
123 idvoto <input type="text" value="5"/> abc idcircunscripcion <input type="text" value="1"/> abc idmesaelectoral <input type="text" value="1"/> abc idprocesoelectoral <input type="text" value="1"/> abc nombrecandidatovotado <input type="text" value="Jorge Paniagui"/> abc codrespuesta <input type="text" value="000"/> abc numerodni <input type="text" value="39739740E"/> abc marcatiempo <input type="text" value="2024-02-19 23:20:28.223954"/>							
1							

Página de Resultado de Borrado

Se han borrado correctamente 1 votos del proceso electoral con identificador 1

Volver

Página Listado de Votos

Lista de votos del proceso electoral 1

idVoto	idCircunscripcion	idMesaElectoral	idProcesoElectoral	Candidato/a Votado/a	Marca de Tiempo	codRespuesta
--------	-------------------	-----------------	--------------------	----------------------	-----------------	--------------

Volver

Ejercicio 3.

Inspeccionamos de nuevo el archivo ‘postgresql.properties’ y vemos:

```
db.pool.name=VotoPool
```

```
db.datasource=org.postgresql.ds.PGConnectionPoolDataSource
```

Nos metemos en la consola de administración con el usuario ‘admin’ y contraseña ‘adminadmin’ que se nos proporciona en el archivo ‘build.properties’ y ‘passwordfile’.

Una vez dentro, seleccionamos nuestra JDBC Connection Pool: VotoPool. Esto es lo que aparece:

Edit JDBC Connection Pool

Modify an existing JDBC connection pool. A JDBC connection pool is a group of reusable connections for a particular database.

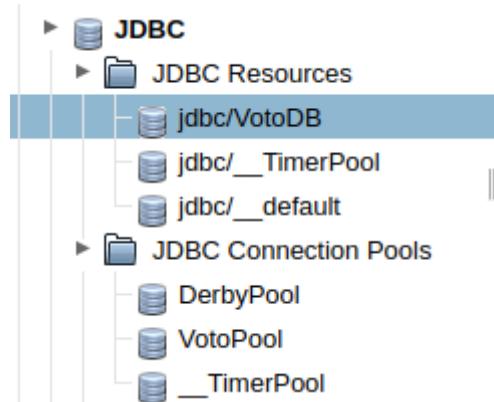
General Settings

Pool Name:	VotoPool
Resource Type:	javax.sql.ConnectionPoolDataSource
Datasource Classname:	org.postgresql.ds.PGConnectionPoolDataSource
Driver Classname:	
Ping:	<input checked="" type="checkbox"/> When enabled, the pool is pinged during creation or reconfiguration to identify and warn of any erroneous values for its attributes
Deployment Order:	100
Description:	

Pool Settings

Initial and Minimum Pool Size:	8 Connections
Maximum Pool Size:	32 Connections
Pool Resize Quantity:	2 Connections
Idle Timeout:	300 Seconds
Max Wait Time:	60000 Milliseconds

Como vemos, coinciden tanto el Pool Name como el Datasource. Además, comprobamos que han sido creados los recursos JDBC y pool de conexiones:



Como vemos en la imagen anterior, los valores para estos parámetros son:
Initial and Minimum Pool Size = 8 Connections

Este se trata del número de hilos concurrentes que el servidor va a administrar como mínimo y que por tanto es también el número de hilos que se crean cuando se inicializa. Es por ello que este parámetro afecta directamente al rendimiento de la aplicación al ser inicializada.

Maximum Pool Size = 32 Connections

Son el número máximo de conexiones que el servidor puede tener para resolver los problemas de los clientes. Al igual que antes, afecta al rendimiento, no pudiendo ser demasiado bajo porque quedarían clientes en espera, ni demasiado alto, ya que podría sobrecargarse el servidor.

Pool Resize Quantity = 2 Connections

En caso de que se cumpla el tiempo máximo de una desconexión, se borrarán este número de conexiones.

Idle Timeout = 300 Seconds

Tiempo de timeout en para una desconexión, si este es demasiado alto, a lo mejor se pierden recursos que no están siendo utilizados y que se necesitan para otras cosas. Si es muy bajo podría hacer que los clientes no acaben sus peticiones.

Max Wait Time = 60000 Milliseconds

Es el tiempo máximo que un cliente espera hasta que envía una señal de timeout. Si es muy baja, puede dar a que el servidor repita paquetes.

Ejercicio 4.

En estos fragmentos de código es donde se declaran las consultas sql estáticas dentro de la aplicación.

Primero tenemos la consulta de si la información del censo es correcta, en la cual se buscan los datos proporcionados por el votante para ver si hay alguna persona censada con esos datos.

```
private static final String SELECT_CENSO_QRY =
    "select * from censo " +
    "where numeroDNI=? " +
    " and nombre=? " +
    " and fechaNacimiento=? " +
    " and codigoAutorizacion=? ";
```

Y aquí tenemos el registro o inserción del voto en la base de datos, en la cual se realiza un insert into voto:

```

private static final String INSERT_VOTO_QRY =
    "insert into voto(" +
    "idCircunscripcion,idMesaElectoral,idProcesoElectoral,nombreCandidatoVotado,numeroDNI) " +
    " values (?,?,?,?,?)";

```

A su vez, ambas se instancian como métodos de la clase VotoDAO.

```

/**
 * getQryCompruebaCenso
 */
private String getQryCompruebaCenso(CensoBean censo) {
    String qry = "select * from censo " +
        "where numeroDNI ='" + censo.getNumeroDNI() +
        "' and nombre=' " + censo.getNombre() +
        "' and fechaNacimiento=' " + censo.getFechaNacimiento() +
        "' and codigoAutorizacion=' " + censo.getCodigoAutorizacion() + "'";
    return qry;
}

/**
 * getQryInsertVoto
 */
private String getQryInsertVoto(VotoBean voto) {
    String qry = "insert into voto(" +
        "idCircunscripcion," +
        "idMesaElectoral,idProcesoElectoral," +
        "nombreCandidatoVotado, numeroDNI)" +
        " values (" +
        "''' + voto.getIdCircunscripcion() + '''," +
        "''' + voto.getIdMesaElectoral() + '''," +
        "''' + voto.getIdProcesoElectoral() + '''," +
        "''' + voto.getNombreCandidatoVotado() + '''," +
        "''' + voto.getCenso().getNumeroDNI() + ''')";
    return qry;
}

```

Ejercicio 5.

El método errorLog se encarga de imprimir la traza de depuración. En primer lugar, vemos que se utiliza para guardar todas las consultas SQL que se llevan a cabo. Además, en caso de que algo falle, también el guarda mensaje de error que se le indique como: “¡El censo no está actualizado!” o “¡El voto tiene campos vacíos!”. Además de esto, guarda los errores que encuentra el propio programa llamando al método como ‘errorLog(e.toString());’ siendo e de tipo Exception.

Tras volver a abrir la aplicación desde el ‘testbd.xhtml’ y registrar un nuevo voto con el modo debug activado, hemos visto tanto el log del servidor de aplicaciones tanto del portal web:

https://10.4.1.1:4848/common/logViewer/logViewer.jsf?instanceName=server&logLevel=INFO&viewResults=true

Log Viewer

View, search, and filter a server log file using basic and advanced options. Refer to the Log Levels page for information about log levels you can filter here.

[Advanced Search](#)

Search Criteria

Text search: Only log entries containing the specified text will be displayed. Search is case sensitive.

Timestamp: Most Recent Specific Range:

Log Level: INFO Do not include more severe messages

Log entries are limited to those stored in the log file. Set appropriate log level in the Log Level page to ensure data is logged.

[Modify Search](#) Search Close

Instance: **Log File:**

Log Viewer Results (40)

Records before 527 Log File Record Numbers 527 through 566 Records after 566		
Record Number	Log Level	Message
566	INFO	[LOG INFO]: Voto registrado correctamente! (details)
565	SEVERE	[directConnection=true] select idVoto, codRespuesta, marcaTiempo from voto where idProcesoElectoral = ? and numeroDNI = ? and nombre = ? and fechaNacimiento = ? and codigoAutorizacion = ?
564	SEVERE	[directConnection=true] insert into voto(idCircunscripcion, idMesaElectoral, idProcesoElectoral, nombre, apellido, numeroDNI, fechaNacimiento, codRespuesta, marcaTiempo, idVoto) values (?, ?, ?, ?, ?, ?, ?, ?, ?, ?)
563	SEVERE	[directConnection=true] select * from censo where numeroDNI = ? and nombre = ? and fechaNacimiento = ? and codigoAutorizacion = ?
562	INFO	Created connection pool and added it to PoolManager: Pool [org.glassfish.resourcebase.resources.api.PoolInfo@eb0889af[jndiName=VotoPool, applicationName=null, moduleName=null]]
561	WARNING	Probably the pool org.glassfish.resourcebase.resources.api.PoolInfo@eb0889af[jndiName=VotoPool, applicationName=null, moduleName=null] is not yet initialized (lazy-loading), trying to check ...
560	INFO	[LOG INFO]: Solicitado el registro del voto. (details)

Logger **Timestamp** **Name-Value Pairs**

jakarta.enterprise.logging.stdout 20 feb 2024 10:11:37.543 0
jakarta.enterprise.logging.stderr 20 feb 2024 10:11:37.540 0
jakarta.enterprise.logging.stderr 20 feb 2024 10:11:37.536 0
jakarta.enterprise.logging.stderr 20 feb 2024 10:11:37.527 0
jakarta.enterprise.resource.resourceadapter.com.sun.enterprise.resource.pool 20 feb 2024 10:11:37.509 0
jakarta.enterprise.resource.resourceadapter.com.sun.enterprise.connectors.service 20 feb 2024 10:11:37.494 0
jakarta.enterprise.logging.stdout 20 feb 2024 10:11:37.485 0

Como de la terminal:

```
[2024-02-20T10:11:37.485009+01:00] [GF 7.0.9] [INFO] [] [jakarta.enterprise.logging.stdout] [tid: _ThreadID=26 _ThreadName=http-listener-1(2)] [levelValue: 800] [[
[LOG INFO]:Solicitado el registro del voto.]]
[2024-02-20T10:11:37.494856+01:00] [GF 7.0.9] [WARNING] [] [jakarta.enterprise.resource.resourceadapter.com.sun.enterprise.connectors.service] [tid: _ThreadID=26 _ThreadName=http-listener-1(2)] [levelValue: 900] [[
Probably the pool org.glassfish.resourcebase.resources.api.PoolInfo@eb0889af[jndiName=VotoPool, applicationName=null, moduleName=null] is not yet initialized (lazy-loading), trying to check ...]]
[2024-02-20T10:11:37.509071+01:00] [GF 7.0.9] [INFO] [] [jakarta.enterprise.resource.resourceadapter.com.sun.enterprise.resource.pool] [tid: _ThreadID=26 _ThreadName=http-listener-1(2)] [levelValue: 800] [[
Created connection pool and added it to PoolManager: Pool [org.glassfish.resourcebase.resources.api.PoolInfo@eb0889af[jndiName=VotoPool, applicationName=null, moduleName=null]] PoolSize=0 FreeResources=0 QueueSize=0 matching=off validation=off]]
[2024-02-20T10:11:37.527147+01:00] [GF 7.0.9] [SEVERE] [] [jakarta.enterprise.logging.stderr] [tid: _ThreadID=26 _ThreadName=http-listener-1(2)] [levelValue: 1000] [[
[directConnection=true] select * from censo where numeroDNI=? and nombre=? and fechaNacimiento=? and codigoAutorizacion=?]]
[2024-02-20T10:11:37.536523+01:00] [GF 7.0.9] [SEVERE] [] [jakarta.enterprise.logging.stderr] [tid: _ThreadID=26 _ThreadName=http-listener-1(2)] [levelValue: 1000] [[
[directConnection=true] insert into voto(idCircunscripcion, idMesaElectoral, idProcesoElectoral, nombreCandidatoVotado, numeroDNI) values (?, ?, ?, ?, ?)]]
[2024-02-20T10:11:37.540578+01:00] [GF 7.0.9] [SEVERE] [] [jakarta.enterprise.logging.stderr] [tid: _ThreadID=26 _ThreadName=http-listener-1(2)] [levelValue: 1000] [[
[directConnection=true] select idVoto, codRespuesta, marcaTiempo from voto where idProcesoElectoral = ? and numeroDNI = ?]]
[2024-02-20T10:11:37.543933+01:00] [GF 7.0.9] [INFO] [] [jakarta.enterprise.logging.stdout] [tid: _ThreadID=26 _ThreadName=http-listener-1(2)] [levelValue: 800] [[
[LOG INFO]: Voto registrado correctamente!]]
si2@si2srv01:~$ _
```

Vemos que ambas incluyen información adicional de las acciones que se llevan a cabo durante el proceso del registro del voto. Donde se ve como se hace la connection pool y luego todas las consultas necesarias. Haciendo

también dos log info, el que informa que se ha solicitado el registro de voto y el que informa que se ha realizado con éxito.

Ejercicio 6.

```
public class VotoDAOWS extends DBTester {
    public boolean compruebaCenso(CensoBean censo) {
        // Unos comentarios en la linea de isPrepared()
        PreparedStatement pstmt = null;

        try {
            // Crear una conexion u obtenerla del pool
            con = getConnection();

            // Se busca la ocurrencia de la informacion del censo en la tabla
            if (isPrepared() == true) {
                String select = SELECT_CENSO_QRY;
                errorLog(select);
                pstmt = con.prepareStatement(select);
                pstmt.setString(1, censo.getNumeroDNI());
                pstmt.setString(2, censo.getNombre());
                pstmt.setString(3, censo.getFechaNacimiento());
                pstmt.setString(4, censo.getCodigoAutorizacion());
                rs = pstmt.executeQuery();
            } else {
                stmt = con.createStatement();
                qry = getQryCompruebaCenso(censo);
                errorLog(qry);
                rs = stmt.executeQuery(qry);
            }

            /* Si hay siguiente registro, la info sobre el censo es valida OK */
            ret = rs.next();

            // Comprobamos año censo actual
            if (rs.getString("anioCenso").compareTo(String.valueOf(LocalDate.now().getYear())) != 0) {
                errorLog("El censo no está actualizado!");
                ret = false;
            }

            } catch (Exception ee) {
                errorLog(ee.toString());
                ret = false;
            } finally {
                try {
                    if (rs != null) {
                        rs.close(); rs = null;
                    }
                    if (stmt != null) {
                        stmt.close(); stmt = null;
                    }
                    if (pstmt != null) {
                        pstmt.close(); pstmt = null;
                    }
                    if (con != null) {
                        closeConnection(con); con = null;
                    }
                } catch (SQLException e) {
                }
            }
        }

        return ret;
    }
}
```

```

/**
 * @return
 */
@WebMethod
public synchronized VotoBean registraVoto(VotoBean voto) {
    Connection con = null;
    Statement stmt = null;
    ResultSet rs = null;
    boolean ret = false;
    String codRespuesta = "999"; // En principio, voto invalido

    PreparedStatement pstmt = null;

    // Comprobamos campos voto

    if (voto.getIdCircunscripcion() == null || voto.getIdMesaElectoral() == null ||
        voto.getIdProcesoElectoral() == null || voto.getNombreCandidatoVotado() == null ||
        voto.getCenso() == null || voto.getCenso().getNumeroDNI() == null) {
        errorLog("El voto tiene campos vacíos!");
        return null;
    }

    // Registrar el voto en la base de datos

    try {

        // Obtener conexión
        con = getConnection();

        // Insertar en la base de datos el voto

        if (isPrepared() == true) {
            String insert = INSERT_VOTO_QRY;
            errorLog(insert);
            pstmt = con.prepareStatement(insert);
            pstmt.setString(1, voto.getIdCircunscripcion());
            pstmt.setString(2, voto.getIdMesaElectoral());
            pstmt.setString(3, voto.getIdProcesoElectoral());
            pstmt.setString(4, voto.getNombreCandidatoVotado());
            pstmt.setString(5, voto.getCenso().getNumeroDNI());
            ret = false;

            if (!pstmt.execute() && pstmt.getUpdateCount() == 1) {
                ret = true;
            }
        } else {

            stmt = con.createStatement();
            String insert = getQryInsertVoto(voto);
            errorLog(insert);
            ret = false;

            if (!stmt.execute(insert) && stmt.getUpdateCount() == 1) {
                ret = true;
            }
        }
    }

    // Obtener id.voto
}

```

```

// Obtener id.voto

if (ret) {

    if (isPrepared() == true) {
        String select = SELECT_VOTO_TRANSACCION_QRY;
        errorLog(select);
        pstmt = con.prepareStatement(select);
        pstmt.setString(1, voto.getIdProcesoElectoral());
        pstmt.setString(2, voto.getCenso().getNumeroDNI());
        rs = pstmt.executeQuery();
    } else {
        String select = getQryBuscaVotoTransaccion(voto);
        errorLog(select);
        rs = stmt.executeQuery(select);
    }

    if (rs.next()) {

        // Completamos la información que se genera al insertar el
        // voto en la base de datos

        voto.setIdVoto(String.valueOf(rs.getInt("idVoto")));
        voto.setCodigoRespuesta(rs.getString("codRespuesta"));
        voto.setMarcaTiempo(rs.getString("marcaTiempo"));

    } else {
        ret = false;
    }
}

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

return (ret)?voto:null;
}

```

```

    /**
     * @return the debug
     */
    @WebMethod
    public boolean isDebug() {
        return debug;
    }

    /**
     * @param debug the debug to set
     */
    @WebMethod
    public void setDebug(boolean debug) {
        this.debug = debug;
    }

```

```

    /**
     * TODO: Metodos isPrepared() y setPrepared()
     */
    /** ****
     * @WebMethod
     public boolean isPrepared() {
         return prepared;
     }
     * @WebMethod
     public void setPrepared(boolean prepared) {
         this.prepared = prepared;
     }
     */

    @Override
    @WebMethod
    public boolean isDirectConnection() {
        return super.isDirectConnection(); // Llama al método isDirectConnection() de la superclase DBTester
    }

    @Override
    @WebMethod
    public void setDirectConnection(boolean directConnection) {
        super.setDirectConnection(directConnection);
    }

```

¿Por qué se ha de alterar el parámetro de retorno del método registraVoto() para que devuelva el voto modificado en lugar de un boolean?

- Porque es una representación más completa de lo que queremos devolver que es el voto modificado.

Ejercicio 7.

```
~ /Doc/i/t/s/s/p/P/P1-ws ant desplegar-servicio
Buildfile: /home/daniel/Documents/ing.informatica/tercero/segundo/si2/practica/P1/P1-ws/build.xml

desplegar-servicio:
[exec] Application deployed with name P1-ws-ws.
[exec] Command deploy executed successfully.

BUILD SUCCESSFUL
Total time: 5 seconds
```

Eclipse GlassFish

Common Tasks Domain server (Admin Server) Clusters Standalone Instances Notes Applications P1-ws-ws Lifecycle Modules Monitoring Data Resources Concurrent Resources Connectors JDBC JMS Resources JNDI JavaMail Sessions Resource Adapter Configurations default-config server-config

General Descriptor Edit Application Modify an existing application or module.

Name: P1-ws-ws Status: Virtual Servers: server

Context Root: P1-ws-ws Path relative to server's base URL. If empty, takes the default context path of a web application.

Implicit CDI Implicit discovery of CDI beans

Location: \${com.sun.aas.instanceRoot}/applications/P1-ws-ws/ Deployment Order: 100 A number that determines the loading order of the application at server startup. Lower numbers are loaded first. The default is 100.

Libraries:

Description:

Modules and Components (3)

Module Name	Engines	Component Name	Type	Action
P1-ws-ws	[web]	-----	-----	Launch
P1-ws-ws		default	Servlet	
P1-ws-ws		jsp	Servlet	

Eclipse GlassFish

Tree Common Tasks Domain server (Admin Server) Clusters Standalone Instances Notes Applications P1-ws-ws Lifecycle Modules Monitoring Data Resources Concurrent Resources Connectors JDBC JMS Resources JNDI JavaMail Sessions Resource Adapter Configurations default-config server-config

Web Service Endpoint Information

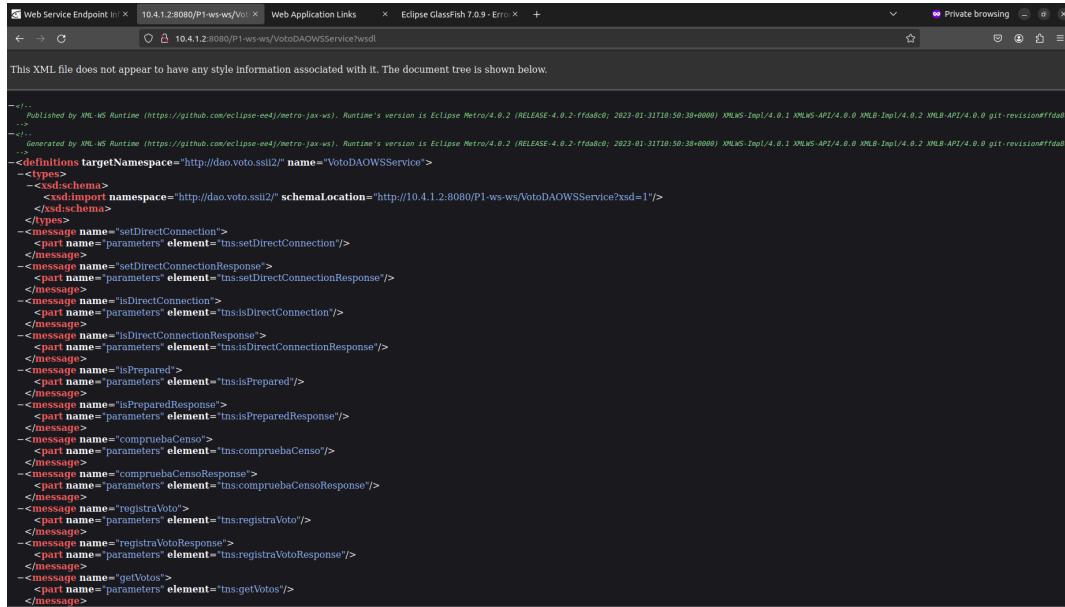
View details about a web service endpoint.

Application Name: P1-ws-ws Tester: /P1-ws-ws/VotoDAOWebService?Tester WSDL: /P1-ws-ws/VotoDAOWebService?wsdl Endpoint Name: VotoDAOWebService Service Name: VotoDAOWebService Port Name: VotoDAOWebPort Deployment Type: 109 Implementation Type: SERVLET Implementation Class Name: ssii2.voto.dao.VotoDAOWebService Endpoint Address URI: /P1-ws-ws/VotoDAOWebService Namespace: http://dao.voto.ssii2/

¿En qué fichero están definidos los tipos de datos intercambiados con el webservice?

- [/P1-ws-ws/VotoDAOWSService?wsdl](http://10.4.1.2:8080/P1-ws-ws/VotoDAOWSService?wsdl)

¿Qué tipos de datos predefinidos se usan?



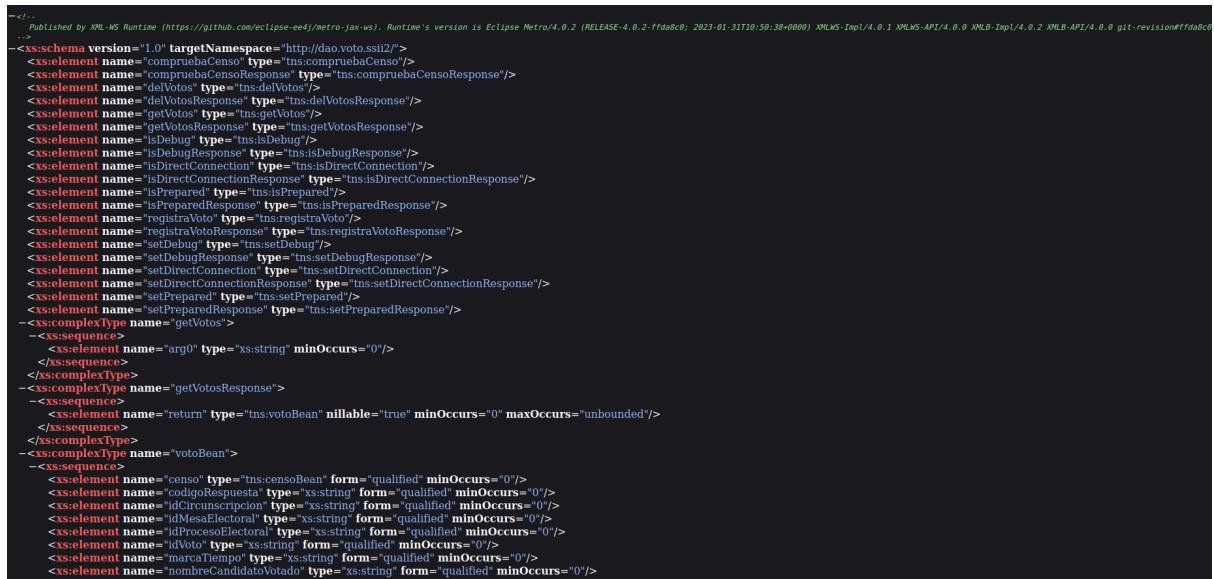
This XML file does not appear to have any style information associated with it. The document tree is shown below.

```
<!-- Published by XML-WS Runtime (https://github.com/eclipse-eed/metro-jax-ws). Runtime's version is Eclipse Metro/4.0.2 (RELEASE-4.0.2-ffda8c0; 2023-01-31T10:50:38+0000) XMLWS-Impl/4.0.1 XMLWS-API/4.0.0 XMLB-Impl/4.0.2 XMLB-API/4.0.0 git-revision#ffda8c0 -->
<!-- Generated by XML-WS Runtime (https://github.com/eclipse-eed/metro-jax-ws). Runtime's version is Eclipse Metro/4.0.2 (RELEASE-4.0.2-ffda8c0; 2023-01-31T10:50:38+0000) XMLWS-Impl/4.0.1 XMLWS-API/4.0.0 XMLB-Impl/4.0.2 XMLB-API/4.0.0 git-revision#ffda8c0 -->
<definitions targetNamespace="http://dao.voto.ssii2/" name="VotoDAOWSService">
  <types>
    <xsd:schema>
      <xsd:import namespace="http://dao.voto.ssii2/" schemaLocation="http://10.4.1.2:8080/P1-ws-ws/VotoDAOWSService?xsd=1"/>
    </xsd:schema>
  </types>
  <message name="setDirectConnection">
    <part name="parameters" element="tns:setDirectConnection"/>
  </message>
  <message name="setDirectConnectionResponse">
    <part name="parameters" element="tns:setDirectConnectionResponse"/>
  </message>
  <message name="isDirectConnection">
    <part name="parameters" element="tns:isDirectConnection"/>
  </message>
  <message name="isDirectConnectionResponse">
    <part name="parameters" element="tns:isDirectConnectionResponse"/>
  </message>
  <message name="isPrepared">
    <part name="parameters" element="tns:isPrepared"/>
  </message>
  <message name="isPreparedResponse">
    <part name="parameters" element="tns:isPreparedResponse"/>
  </message>
  <message name="compruebaCenso">
    <part name="parameters" element="tns:compruebaCenso"/>
  </message>
  <message name="compruebaCensoResponse">
    <part name="parameters" element="tns:compruebaCensoResponse"/>
  </message>
  <message name="registraVoto">
    <part name="parameters" element="tns:registraVoto"/>
  </message>
  <message name="registraVotoResponse">
    <part name="parameters" element="tns:registraVotoResponse"/>
  </message>
  <message name="getVotos">
    <part name="parameters" element="tns:getVotos"/>
  </message>
</definitions>
```

¿Cuáles son los tipos de datos nuevos que se definen?

Dentro de la sección `<types>`, se importa un esquema XML (XSD) desde la ubicación "<http://10.4.1.2:8080/P1-ws-ws/VotoDAOWSService?xsd=1>". Este esquema contendrá las definiciones de los tipos de datos utilizados en el servicio web.

Aquí podemos observar la primera parte de este archivo:



```
<!-- Published by XML-WS Runtime (https://github.com/eclipse-eed/metro-jax-ws). Runtime's version is Eclipse Metro/4.0.2 (RELEASE-4.0.2-ffda8c0; 2023-01-31T10:50:38+0000) XMLWS-Impl/4.0.1 XMLWS-API/4.0.0 XMLB-Impl/4.0.2 XMLB-API/4.0.0 git-revision#ffda8c0 -->
<xs:schema version="1.1" xmlns="http://dao.voto.ssii2/">
  <xs:element name="compruebaCenso" type="tns:compruebaCenso"/>
  <xs:element name="compruebaCensoResponse" type="tns:compruebaCensoResponse"/>
  <xs:element name="delVotos" type="tns:delVotos"/>
  <xs:element name="getVotos" type="tns:getVotos"/>
  <xs:element name="getVotosResponse" type="tns:getVotosResponse"/>
  <xs:element name="isDebug" type="tns:isDebug"/>
  <xs:element name="isDebugResponse" type="tns:isDebugResponse"/>
  <xs:element name="isDirectConnection" type="tns:isDirectConnection"/>
  <xs:element name="isDirectConnectionResponse" type="tns:isDirectConnectionResponse"/>
  <xs:element name="isPrepared" type="tns:isPrepared"/>
  <xs:element name="isPreparedResponse" type="tns:isPreparedResponse"/>
  <xs:element name="registraVoto" type="tns:registraVoto"/>
  <xs:element name="registraVotoResponse" type="tns:registraVotoResponse"/>
  <xs:element name="setDebug" type="tns:setDebug"/>
  <xs:element name="setDirectConnection" type="tns:setDirectConnection"/>
  <xs:element name="setDirectConnectionResponse" type="tns:setDirectConnectionResponse"/>
  <xs:element name="setPrepared" type="tns:setPrepared"/>
  <xs:element name="setPreparedResponse" type="tns:setPreparedResponse"/>
  <xs:complexType name="getVotos">
    <xs:sequence>
      <xs:element name="arg1" type="xs:string" minOccurs="0"/>
    </xs:sequence>
  </xs:complexType>
  <xs:complexType name="getVotosResponse">
    <xs:sequence>
      <xs:element name="return" type="tns:votoBean" nillable="true" minOccurs="0" maxOccurs="unbounded"/>
    </xs:sequence>
  </xs:complexType>
  <xs:complexType name="votoBean">
    <xs:sequence>
      <xs:element name="censo" type="tns:censoBean" form="qualified" minOccurs="0"/>
      <xs:element name="codigoRespuesta" type="xs:string" form="qualified" minOccurs="0"/>
      <xs:element name="idCircunscripcion" type="xs:string" form="qualified" minOccurs="0"/>
      <xs:element name="idMesaElectoral" type="xs:string" form="qualified" minOccurs="0"/>
      <xs:element name="idProcesoElectoral" type="xs:string" form="qualified" minOccurs="0"/>
      <xs:element name="idVoto" type="xs:string" form="qualified" minOccurs="0"/>
      <xs:element name="marcaTiempo" type="xs:string" form="qualified" minOccurs="0"/>
      <xs:element name="nombreCandidatoVotado" type="xs:string" form="qualified" minOccurs="0"/>
    </xs:sequence>
```

censoBean: incluye campos como "codigoAutorizacion", "fechaNacimiento", "nombre", y "numeroDNI", todos ellos de tipo cadena (xs:string).

votoBean: Incluye campos como "censo" (que es un objeto de tipo censoBean), "codigoRespuesta", "idCircunscripcion", "idMesaElectoral", "idProcesoElectoral", "idVoto", "marcaTiempo" y "nombreCandidatoVotado", todos ellos de tipo cadena (xs:string).

setDebug: Incluye un campo "arg0" que indica si se activa o desactiva la depuración, representado como un booleano (xs:boolean).

registraVoto: Incluye un campo "arg0" que contiene un objeto de tipo votoBean.

getVotosResponse: que representa una respuesta que devuelve una lista de objetos "votoBean"

delVotosResponse se define un elemento que contiene un valor (xs:int)

¿Qué etiqueta está asociada a los métodos invocados en el webservice?

La etiqueta operation

¿Qué etiqueta describe los mensajes intercambiados en la invocación de los métodos del webservice?

La etiqueta message

```
--<message name="setDirectConnection">
  <part name="parameters" element="tns:setDirectConnection"/>
</message>
--<message name="setDirectConnectionResponse">
  <part name="parameters" element="tns:setDirectConnectionResponse"/>
</message>
--<message name="isDirectConnection">
  <part name="parameters" element="tns:isDirectConnection"/>
</message>
--<message name="isDirectConnectionResponse">
  <part name="parameters" element="tns:isDirectConnectionResponse"/>
</message>
--<message name="isPrepared">
  <part name="parameters" element="tns:isPrepared"/>
</message>
--<message name="isPreparedResponse">
  <part name="parameters" element="tns:isPreparedResponse"/>
</message>
```

¿En qué etiqueta se especifica el protocolo de comunicación con el webservice?

```
-<portType name="VotoDAOWS">
  -<operation name="setDirectConnection">
    <input wsam:Action="http://dao.voto.ssii2/VotoDAOWS/setDirectConnectionRequest" message="tns:setDirectConnection"/>
    <output wsam:Action="http://dao.voto.ssii2/VotoDAOWS/setDirectConnectionResponse" message="tns:setDirectConnectionResponse"/>
  </operation>
  -<operation name="isDirectConnection">
    <input wsam:Action="http://dao.voto.ssii2/VotoDAOWS/isDirectConnectionRequest" message="tns:isDirectConnection"/>
    <output wsam:Action="http://dao.voto.ssii2/VotoDAOWS/isDirectConnectionResponse" message="tns:isDirectConnectionResponse"/>
  </operation>
  -<operation name="isPrepared">
    <input wsam:Action="http://dao.voto.ssii2/VotoDAOWS/isPreparedRequest" message="tns:isPrepared"/>
    <output wsam:Action="http://dao.voto.ssii2/VotoDAOWS/isPreparedResponse" message="tns:isPreparedResponse"/>
  </operation>
  -<operation name="compruebaCenso">
    <input wsam:Action="http://dao.voto.ssii2/VotoDAOWS/compruebaCensoRequest" message="tns:compruebaCenso"/>
    <output wsam:Action="http://dao.voto.ssii2/VotoDAOWS/compruebaCensoResponse" message="tns:compruebaCensoResponse"/>
  </operation>
  -<operation name="registraVoto">
    <input wsam:Action="http://dao.voto.ssii2/VotoDAOWS/registraVotoRequest" message="tns:registraVoto"/>
    <output wsam:Action="http://dao.voto.ssii2/VotoDAOWS/registraVotoResponse" message="tns:registraVotoResponse"/>
  </operation>
  -<operation name="getVotos">
    <input wsam:Action="http://dao.voto.ssii2/VotoDAOWS/getVotosRequest" message="tns:getVotos"/>
    <output wsam:Action="http://dao.voto.ssii2/VotoDAOWS/getVotosResponse" message="tns:getVotosResponse"/>
  </operation>
  -<operation name="delVotos">
    <input wsam:Action="http://dao.voto.ssii2/VotoDAOWS/delVotosRequest" message="tns:delVotos"/>
    <output wsam:Action="http://dao.voto.ssii2/VotoDAOWS/delVotosResponse" message="tns:delVotosResponse"/>
  </operation>
  -<operation name="setPrepared">
    <input wsam:Action="http://dao.voto.ssii2/VotoDAOWS/setPreparedRequest" message="tns:setPrepared"/>
    <output wsam:Action="http://dao.voto.ssii2/VotoDAOWS/setPreparedResponse" message="tns:setPreparedResponse"/>
  </operation>
  -<operation name="isDebug">
    <input wsam:Action="http://dao.voto.ssii2/VotoDAOWS/isDebugRequest" message="tns:isDebug"/>
    <output wsam:Action="http://dao.voto.ssii2/VotoDAOWS/isDebugResponse" message="tns:isDebugResponse"/>
  </operation>
  -<operation name="setDebug">
    <input wsam:Action="http://dao.voto.ssii2/VotoDAOWS/setDebugRequest" message="tns:setDebug"/>
    <output wsam:Action="http://dao.voto.ssii2/VotoDAOWS/setDebugResponse" message="tns:setDebugResponse"/>
  </operation>
```

Dentro de la etiqueta operation se especifica como se relacionan los mensajes de entrada y salido junto con las acciones SOAP asociadas a estas operaciones

¿En qué etiqueta se especifica la URL a la que deberá acceder un cliente para acceder al webservice?

```
-<service name="VotoDAOSService">
  -<port name="VotoDAOWSPort" binding="tns:VotoDAOWSBinding">
    <soap:address location="http://10.4.1.2:8080/P1-ws-ws/VotoDAOSService"/>
  </port>
</service>
</definitions>
```

La URL a la que un cliente deberá acceder para interactuar con el servicio web se especifica dentro de la etiqueta <soap:address> esta etiqueta se encuentra dentro de la sección <service> que describe el servicio web.

Ejercicio 8.

```
package ssii2.controlador;

import jakarta.enterprise.context.SessionScoped;
import jakarta.inject.Named;
import jakarta.inject.Inject;
import java.io.Serializable;
import jakarta.faces.context.*;
import java.util.List;
import java.util.ArrayList;

import ssii2.voto.*;
import ssii2.interaccion.*;
import ssii2.servicio.VotoDAOWS;
import ssii2.servicio.VotoDAOWSService;

import jakarta.xml.ws.BindingProvider;
```

```
public String enviarVoto() {

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

    VotoDAOWSService service = new VotoDAOWSService();
    VotoDAOWS dao = service.getVotoDAOWSPort();

    try {
        // Traducimos la información del voto al formato del servicio web
        ssii2.servicio.VotoBean votoParaServicio = traducirVotoParaServicio(this.voto);

        // Llamamos al método registraVoto() del servicio web
        ssii2.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("¡Voto registrado correctamente!");
        }
        return "respuesta";
    } catch (Exception e) {
        // Manejamos la excepción
        e.printStackTrace();
        return "error";
    }
}
```

Se instancia un objeto VotoDaoWSService que representa el servicio web
Se obtiene el stun VotoDaoWS a partir del objeto VotoDAOWSService
mediante el metodo getVotoDAOWSPort()

Se llama al método traducirVotoParaServicio() para convertir el objeto
VotoBean del cliente al formato esperado por el servidor
ssii2.servicio.VotoBean

Invocación del metodo registraVoto del stub VotoDAOWS pasando como parametro el objeto votoParaServicio

Despues se obtiene el objeto VotoBean devuelto por el servicio web que contiene los valores actualizados del voto

se utiliza try-catch para manejar las excepciones, en caso de excepción se maneja imprimiendo la pila de llamadas en la consola para depuración y retornando una página de error

Ejercicio 9.

```
<!-- Nuevo parametro de inicializacion para la URL del servicio remoto -->
<context-param>
    <param-name>servicioRemotoURL</param-name>
    <param-value>http://10.4.1.2:8080/P1-ws-ws/VotoDAOWebService</param-value>
</context-param>
```

Ejercicio 10.

```
public String borrarVotos() {
    try {
        VotoDAOWebService service = new VotoDAOWebService();
        VotoDAOWS dao = service.getVotoDAOWebServicePort();

        // Llamada al método del servicio web para borrar votos
        int votosBorrados = dao.delVotos(this.voto.getIdProcesoElectoral());

        // Almacenamos el número de votos borrados en la sesión para mostrarlo en la página de éxito
        FacesContext.getCurrentInstance().getExternalContext().getSessionMap()
            .put("numVotosBorrados", String.valueOf(votosBorrados));

        return "borradook";
    } catch (Exception e) {
        e.printStackTrace();
        return "error";
    }
}
```

```

public String consultarVotos() {
    try {
        VotoDAOService service = new VotoDAOService();
        VotoDAO dao = service.getVotoDAOSPort();

        // Llamada al método del servicio web para consultar votos
        List<ssii2.servicio.VotoBean> votos = dao.getVotos(this.voto.getIdProcesoElectoral());

        // Convertimos la lista de votos del servicio web a un array de VotoBean[]
        ArrayList<VotoBean> votosList = new ArrayList<>();
        for (ssii2.servicio.VotoBean v : votos) {
            VotoBean votoBean = new VotoBean();
            votoBean.setIdVoto(v.getIdVoto());
            votoBean.setMarcaTiempo(v.getMarcaTiempo());
            votoBean.setCodigoRespuesta(v.getCodigoRespuesta());
            votoBean.setIdCircunscripcion(v.getIdCircunscripcion());
            votoBean.setIdMesaElectoral(v.getIdMesaElectoral());
            votoBean.setIdProcesoElectoral(v.getIdProcesoElectoral());
            votoBean.setNombreCandidatoVotado(v.getNombreCandidatoVotado());

            votosList.add(votoBean);
        }
        VotoBean[] votosArray = votosList.toArray(new VotoBean[votosList.size()]);

        // Almacenamos los votos en la sesión para mostrarlos en la página listavotos.xhtml
        FacesContext.getCurrentInstance().getExternalContext().getSessionMap()
            .put("votosObtenidos", votosArray);

        return "listadoVotos";
    } catch (Exception e) {
        e.printStackTrace();
        return "error";
    }
}

```

Ejercicio 11.

Primero realizamos la importación manual del WSDL del servicio desde la URL de dicho servicio

```

~/.Doc/i/t/s/s/p/P1-ws || wget http://10.4.1.2:8080/P1-ws-ws/VotoDAOSService?WSDL -O VotoDAOSService.wsdl

--2024-02-24 12:46:13-- http://10.4.1.2:8080/P1-ws-ws/VotoDAOSService?WSDL
Connecting to 10.4.1.2:8080... connected.
HTTP request sent, awaiting response... 200 OK
Length: unspecified [text/xml]
Saving to: 'VotoDAOSService.wsdl'

VotoDAOSService.w      [ <>           ]   7,69K  --.-KB/s   in 0s

2024-02-24 12:46:13 (315 MB/s) - 'VotoDAOSService.wsdl' saved [7877]

```

Luego se generan las clases javas necesarias en el paquete ssii2.servicio y las guardamos en el directorio build/client/WEB-INF/classes, utilizando la información del archivo WSDL descargado anteriormente.

```

wsimport -d build/client/WEB-INF/classes -p ssii2.servicio VotoDAOService.wsdl

parsing WSDL...

Generating code...

Compiling code...
Note: /home/daniel/Documents/ing.informatica/tercero/segundo/si2/practica/P1/P1-
ws/build/client/WEB-INF/classes/ssii2/servicio/VotoDAOService.java uses or ove-
rrides a deprecated API.
Note: Recompile with -Xlint:deprecation for details.

```

```

ls ~/Doc/i/t/s/s/p/P1-w/b/c/W/c/s/servicio
CensoBean.class          ObjectFactory.class
CompruebaCenso.class     package-info.class
CompruebaCensoResponse.class RegistraVoto.class
DelVotos.class           RegistraVotoResponse.class
DelVotosResponse.class   SetDebug.class
GetVotos.class           SetDebugResponse.class
GetVotosResponse.class   SetDirectConnection.class
IsDebug.class            SetDirectConnectionResponse.class
IsDebugResponse.class    SetPrepared.class
IsDirectConnection.class SetPreparedResponse.class
IsDirectConnectionResponse.class VotoBean.class
IsPrepared.class         VotoDAOWS.class
IsPreparedResponse.class VotoDAOService.class

```

Estas clases son generadas porque wsimport analiza el archivo WSDL del servicio web y genera representaciones Java de los elementos definidos en ese WSDL. Y se usan para interactuar con el servicio web, permitiendo al cliente realizar llamadas al servicio y procesar las respuestas en su aplicación Java.

Ejercicio 12.

```

<target name="generar-stubs" depends="montar-jerarquia" description="Genera los stubs del cliente a partir del archivo WSDL">
    <!-- TODO - Implementar llamada wsimport -->
    <exec executable="wsimport">
        <arg line="-d ${build.client}/WEB-INF/classes" />
        <arg line="-p ${paquete}.servicio" />
        <arg line="${wsdl.url}" />
    </exec>
    <delete file="${build}/${tmpvisclientjar}" />
    <jar jarfile="${build}/${tmpvisclientjar}" >
        <fileset dir="${build.client}/WEB-INF/classes" />
    </jar>
    <move file="${build}/${tmpvisclientjar}" todir="${build.client}/WEB-INF/lib" />
</target>

```

Ejercicio 13.

```
~/Doc/I/t/s/s/p/P1-ws ant cliente 22:35:57
Buildfile: /home/daniel/Documents/ing.informatica/tercero/segundo/si2/practica/P1/P1-ws/build.xml

cliente:
montar-jerarquia:
generar-stubs:
[exec] parsing WSDL...
[exec]
[exec] Generating code...
[exec]
[exec] Compiling code...
[exec] Note: /home/daniel/Documents/ing.informatica/tercero/segundo/si2/practica/P1/P1-ws/build/client/WEB-INF/classes/ssil2/servicio/VotoDAOWSService.java uses or overrides a deprecated API.
[exec] Note: Recompile with -Xlint:deprecation for details.
[jar] Building jar: /home/daniel/Documents/ing.informatica/tercero/segundo/si2/practica/P1/P1-ws/build/P1-ws-clientstubs.jar
[move] Moving 1 file to /home/daniel/Documents/ing.informatica/tercero/segundo/si2/practica/P1/P1-ws/build/client/WEB-INF/lib

montar-jerarquia:
compilar-cliente:
[javac] Compiling 4 source files to /home/daniel/Documents/ing.informatica/tercero/segundo/si2/practica/P1/P1-ws/build/client/WEB-INF/classes

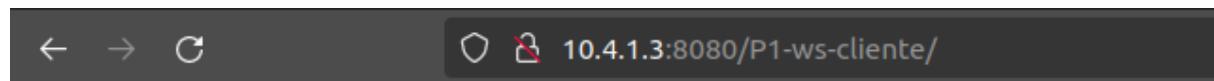
preparar-web-inf-cliente:
[copy] Copying 8 files to /home/daniel/Documents/ing.informatica/tercero/segundo/si2/practica/P1/P1-ws/build/client

empaquetar-cliente:
[jar] Building jar: /home/daniel/Documents/ing.informatica/tercero/segundo/si2/practica/P1/P1-ws/dist/Client/P1-ws-cliente.war

desplegar-cliente:
[exec] Application deployed with name P1-ws.
[exec] Command deploy executed successfully.

BUILD SUCCESSFUL
Total time: 3 seconds
```

Despliegue realizado con éxito



Complete la información sobre el voto:

Id Mesa electoral:

Id Circunscripción:

Id Proceso electoral:

Nombre candidato/a votado/a:



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:

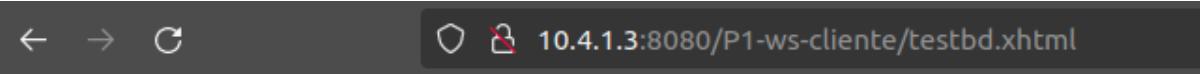
Código Autorización:

Debug: Si No

Prepared Statements: Si No

Conexión directa al a BD: Si No

Registraremos un voto y comprobamos su correcto registro.



Información Sobre el Registro del Voto:

Código Respuesta: 000

Id Voto: 4

Id Proceso electoral: 1

Marca Tiempo Voto: 2024-02-25 23:05:45.915897

Número DNI: 23424434Y

Endbeaver comprobamos que se guarda correctamente en la base de datos

idvoto	idcircunscripcion	idmesaelectoral	idprocesoelectoral	nombreregistradovotado	codrespuest
4	1	1	1	daniel	000

Accedemos al listado



Lista de votos del proceso electoral 1

idVoto	idCircunscripcion	idMesaElectoral	idProcesoElectoral	Candidato/a Votado/a	Marca de Tiempo	codRespuesta
4	1	1	1	daniel	2024-02-25 23:05:45.915897	000

[Volver](#)

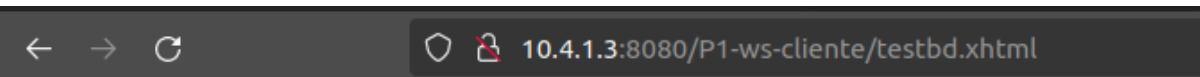
Y comprobamos que también se lista el voto correctamente

Por último borramos el voto

Borrado de Votos:

Id Proceso electoral: 1

[Borrar Votos](#)



Se han borrado correctamente 1 votos del proceso electoral con identificador 1

[Volver](#)

Y verificamos que se ha borrado correctamente

The screenshot shows the MySQL Workbench interface with the following details:

- Top Bar:** Shows connections to "10.4.1.2" and "voto", and tabs for "Properties", "Data", and "ER Diagram".
- Toolbar:** Includes icons for "Enter a SQL expression to filter results (use Ctrl+Space)" and various navigation and search functions.
- Table Structure:** The "voto" table is selected, displaying columns: "idvoto", "idcircunscripcion", "idmesaelectoral", "idprocesoelectoral", "nombrecandidatovotado", and "codrespuesta".
- Data Grid:** A large grid below the table structure displays multiple rows of data corresponding to the selected table.
- Bottom Navigation:** Includes back, forward, and search icons, along with the URL "10.4.1.3:8080/P1-ws-cliente/testbd.xhtml".

Cuestiones

Cuestión número 1:

Teniendo en cuenta el diagrama de la Figura 3, indicar las páginas xhtml y métodos, por los que se pasa para registrar un voto desde voto.xhtml, pero en el caso de uso en que se introduzca una información censal incorrecta. Es decir, de una persona que no se encuentre en la tabla de censo.

La página `voto.xhtml` valida los parámetros del nuevo voto, los registra en `VotoBean`, crea una nueva sesión de forma automática, y redirige la petición al formulario generado por `censo.xhtml`. Este formulario pide al usuario los datos del censo asociados a la persona que está registrando su voto. `censo.xhtml` los registra en `VotoBean` y redirige la solicitud a otro Managed Bean, `ControladorBean`, que se encarga de comprobar la información, pasando por los métodos (`enviarVoto` y `compruebaCenso`), y registrar el voto en la base de datos, interactuando con el modelo (`VotoDAO`). En este caso, `VotoDAO` pasa por un método llamado `compruebaCenso()`, el cual, al meter información incorrecta hace que la función devuelva error por lo que se pasará a la página `error.xhtml` que mostrará el mensaje de error correspondiente.

Cuestión número 2:

De los diferentes Managed Beans y páginas xhtml que se usan en la aplicación, ¿podría indicar cuáles son las páginas xhtml encargadas de obtener la información sobre el voto y el censo cuando se usa voto.xhtml para realizar el registro del voto, y cuáles son los Managed Beans encargados de almacenarla? ¿Qué información se obtiene en cada página xhtml sobre el voto?

Las páginas implicadas para recopilar la información del voto es voto.xhtml y para el censo se pasa a censo.xhtml. Se usan los Managed Beans "VotoBean" que almacena la información sobre el voto y "censoBean" que almacena el censo. En la página voto.xhtml se obtiene: "ID de la Mesa electoral", "ID de la Circunscripción", "ID del Proceso Electoral" y "Nombre del Candidato votado". En la página censo.xhtml se obtiene: "Número de DNI", "Nombre y Apellidos", "Fecha de Nacimiento" y "Código Autorización".

Cuestión número 3:

Respecto a la información sobre el voto que maneja la aplicación (voto en sí y censo), ¿cómo accede a ella el controlador ControladorBean?

El controlador accede a la información sobre el voto mediante la inyección de dependencias de votoBean. Con esta inyección, el controlador puede acceder a todos los métodos y propiedades públicas de votoBean, incluyendo el propio censo.

Cuestión número 4:

Enumere las diferencias que existen en la invocación de páginas XHTML, a la hora de realizar el registro del voto, cuando se utiliza la página de pruebas extendida testbd.xhtml frente a cuando se usa voto.xhtml. ¿Podría indicar por qué funciona correctamente el registro del voto cuando se usa testbd.xhtml a pesar de las diferencias observadas?

Entre las diferencias encontradas están: en testbd.xhtml se solicitan tanto los campos del voto como los del censo, además se incluyen opciones para manejar el comportamiento de la aplicación a la hora de registrar un voto y también se incluyen unas nuevas funcionalidades para borrar y consultar votos. Este registro funciona con ambas páginas ya que el controladorBean es capaz de manejar la lógica de ambas páginas y controlar los campos adicionales de testbd.xhtml, los cuales ignora cuando se accede desde voto.xhtml.