|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | | Escuela Politécnica Superior  Ingeniería Informática  Prácticas de Sistemas Informáticos 2 | | | |
| **Grupo** | **2401** | **Práctica** | 1b | **Fecha** | 19/03/2021 |
| **Alumno/a** | | De las Heras Moreno, Martín | | | |
| **Alumno/a** | | Valderrábano Zamorano, Santiago Manuel | | | |

# Práctica 1b: Arquitectura de JAVA EE

## Cuestión número 1:

Las librerías son:

***import java.sql.Connection;***

***import java.sql.PreparedStatement;***

***import java.sql.ResultSet;***

***import java.sql.SQLException;***

***import java.sql.Statement;***

***import java.util.ArrayList;***

***import javax.ejb.Local;***

Las única anotación que hay es ***@Local*** encima de la clase VisaDAOLocal, que indica que la clase va a estar en el mismo emplazamiento que el cliente.

## Ejercicio 1:

Introducimos en VisaDAOBean las siguientes modificaciones como se indica en el enunciado:

***import javax.ejb.Stateless;***

***@Stateless(mappedName="VisaDAOBean")***

Aquí indicamos que la clase VisaDAO además de extender a DBTester, implementa la interfaz VisaDAOLocal:

***public class VisaDAOBean extends DBTester implements VisaDAOLocal {***

y además borramos el constructor por defecto y modificamos el método getPagos para que devuelva un array y así coincida con la definición VisaDAOLocal de la siguiente forma:

public PagoBean[] getPagos(String idComercio)

***PagoBean[] ret = null;***

***ret = new PagoBean[pagos.size()];***

***ret = pagos.toArray(ret);***

***return ret;***

## Ejercicio número 2:

En este ejercicio realizamos las siguientes modificaciones en **todos los servelets**:

1. Añadimos los nuevos imports:

***import javax.ejb.EJB;***

***import ssii2.visa.\*;***

(Ya que esto incluye VisaDAOLocal)

1. Añadimos el atributo para acceder al EJB Local:

***@EJB(name="VisaDAOBean", beanInterface=VisaDAOLocal.class)***

***private VisaDAOLocal dao;***

1. Comentamos los siguientes imports:

***// import javax.xml.ws.BindingProvider;***

***//*** ***import javax.xml.ws.WebServiceRef***

***// import javax.xml.ws.WebServiceException;***

1. Comentamos las siguientes líneas:

***// String rutaServicio = getServletContext().getInitParameter("rutaServicio");***

***//***

***// BindingProvider bp = (BindingProvider) dao;***

***// bp.getRequestContext().put(BindingProvider.ENDPOINT\_ADDRESS\_PROPERTY, rutaServicio);***

## Cuestión número 2:

El archivo application.xml se encuentra en ./conf/application/META-INF/application.xml. En este fichero se encuentra la información necesaria para el despliegue de la aplicación.

Para ver el contenido de los archivos .ear .jar y .war, ejecuutamos los siguientes comandos:

*jar -tvf dist/P1-ejb.ear* ***//*** *jar -tvf dist/client/P1-ejb-cliente.war*

*Texto

Descripción generada automáticamente*

*jar -tvf dist/server/P1-ejb.jar*

*Texto

Descripción generada automáticamente*

## Ejercicio número 3:

## Para desplegar nuestra aplicación, debemos indicar en los ficheros .properties donde se encuentran alojados BD, cliente y sevidor.

## En build.properties indicamos que tanto as.host.client como as.host.server son igual a 10.1.7.2 ya que tanto cliente como servidor se encontrarán en el mismo servidor. Esto se debe a que hora estamos accediendo a la interfaz local del Enterprise JavaBean (VisaDAOLocal).

## En postgresql.properties indicamos donde se encuentra la base de datos (db.host) y donde se encuentra la interfaz de acceso a la ella (db.client.host). Según está descrito en el esquema del enunciado, la base de datos se encuentra en db.host = 10.1.7.1 y la interfaz de acceso en db.client.host = 10.1.7 2

## Ejercicio número 4:

## En primer lugar, comprobamos que se ha desplegado correctamente todo. Para ello accedemos a la interfaz de acceso a la BD que se encuentra en <http://10.1.7.2:4848> como hemos definido en el ejercicio anterior.

## Interfaz de usuario gráfica, Texto, Aplicación Descripción generada automáticamente

## Después, comprobamos el correcto funcionamiento de la aplicación:

## Desde pago.html

***Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente***

## Interfaz de usuario gráfica Descripción generada automáticamente

## 

## Desde testbd.jsp:

## Interfaz de usuario gráfica, Aplicación Descripción generada automáticamente

## Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico Descripción generada automáticamente

## Captura de pantalla de un celular Descripción generada automáticamente

## Ejercicio número 5:

## Para este ejercicio vamos a implementar un cliente remoto en EJB. Para ello, como se indica en el enunciado, usaremos como material de partida P1-ejb para implementar el servidor remoto. En primer lugar, ejecutamos el comando cp P1-ejb –pr P1-ejb-servidor-remoto. A continuación, realizamos los siguientes cambios escritos en el enunciado:

## Copiar el fichero VisaDAOLocal.java al fichero VisaDAORemote.java.

## Cambiar, en VisaDAORemote.java, el nombre de la interfaz a VisaDAORemote y cambiar la anotación @Local por @Remote.

## Añadir import javax.ejb.Remote; y quitar import javax.ejb.Local

* Hacer que VisaDAOBean implemente ambas interfaces, la local y la remota.

## Por último, tenemos que serializar tanto TarjetaBean con PagoBean. Para ello, modificamos los ficheros .java correspondientes añadiendo:

## import java.io.serializable

## public class TarjetaBean implements Serializable

## public class PagoBean implements Serializable

## Ejercicio número 6:

## Cuestión número 8: