

Sistemas Informáticos 2

Práctica 1b

Cristina Kasner Tourné Guillermo Guridi Mateos

3 de marzo de 2015

Cuestión 1

VisaDAOLocal.java

En este archivo se especifican los métodos que va a utilizar la interfaz `VisaDAOLocal`.

Vemos que tiene la etiqueta `@Local`, que significa:

- Debe ejecutarse en la misma Máquina Virtual Java del EJB al que accede
- Puede ser un componente web u otro EJB
- Para el cliente local, la localización del EJB al que accede no es transparente

Ejercicio 1

Para ajustar los métodos a la interfaz hemos quitado la etiqueta de `synchronize` ya que no se utilizan en los métodos especificados en `VisaDAOLocal`.

También hemos cambiado el retorno del método `getPagos`, antes devolvía `ArrayList` y ahora devuelve `PagoBean []` para respetar las declaraciones de `VisaDAOLocal`.

Ejercicio 2

Hemos eliminado de los *servlets* todo lo referente a los *Web Services* (que estaban en los respectivos métodos `ProcessRequest`) ya que ahora hacemos uso del EJB que hace uso de la interfaz `VisaDAOLocal`.

En `getPagos` hemos vuelto a cambiar el tipo de retorno para que coincida con la declaración que hay en `VisaDAOLocal`.

Cuestión 2

Vemos que en el archivo *application.xml* hay dos módulos:

- Uno que contiene los EJBs, y en general el servidor.
- En el otro se declara la parte web de la aplicación indicando el `.war` del cliente, y la dirección en la que será desplegada (`/P1-ejb-cliente`)

Cuando ejecutamos el comando `jar -tvf` vemos que nos muestra la tabla de contenidos del archivo al que se lo hemos aplicado.

Vamos a explicar el caso del `.ear`.

Contiene 4 archivos:

- **application.xml**, del que ya hemos hablado
- **MANIFEST.MF**, que contiene información sobre la versión de `ant` usada para la empaquetación.
- **P1-ejb-cliente.war**, que es el cliente empaquetado.
- **P1-ejb.jar**, que es el servidor empaquetado.

Ejercicio 3

PANTALLAZO GLASSFISH

Ejercicio 5

- Modificar `TarjetaBean.java`

Además de añadir saldo como atributo hemos añadido los métodos `getSaldo` y `setSaldo`.

- Modificar `VisaDAOBean.java`

Hemos creado las dos consultas y hemos modificado el método `RealizaPago` para que ejecute las dos consultas anteriores usando los *prepared statements*.

Al contrario que las otras consultas que se ejecutan en `RealizaPago`, estas no dan la opción de no usar el *prepared*.

Para la parte de lanzar la excepción en caso de que algo falle, hemos un `String` que explica la razón por la que ha fallado y se los mandamos al cliente junto con la `EJBException`.

- Modificar `ProcesaPago`

Hemos capturado la excepción con `try/catch`

Además de añadir saldo como atributo hemos añadido los métodos `getSaldo` y `setSaldo`.

Apéncices