		Escuela Politécnica Superior Ingeniería Informática Prácticas de Sistemas Informáticos 2			
Grupo	2312	Práctica	1-B	Fecha	2/03/2015
Alumno/a		García Teodoro, Roberto			
Alumno/a		García Roque, Mario			

Práctica 1-B: Título

Cuestión número 1:

Editar el archivo VisaDAOLocal.java y comprobar la definición de dicha interfaz. Anote sus comentarios en la memoria.

Están definidos los métodos que se usaron en la práctica anterior para gestionar los pagos de la tarjeta.

Ejercicio número 1:

Introduzca las siguientes modificaciones en el bean VisaDAOBean para convertirlo en un EJB stateless con interfaz local:

Cambiamos la clase para que se adapte a la estructura de VisaDAOLocal.java

```
import javax.ejb.Stateless;

/**
 * @author jaime
 */
@Stateless(mappedName="VisaDAOBean")
public class VisaDAOBean extends DBTester implements VisaDAOLocal {

    private boolean debug = false;
```

y compilamos el servidor:

```
e280635@11-6-66-8:~/Desktop/P1-ejb$ ant compilar-servidor
Buildfile: /home/alumnos/e280635/Desktop/P1-ejb/build.xml

montar-jerarquia:

compilar-servidor:
    [javac] Compiling 3 source files to /home/alumnos/e280635/Desktop/P1-ejb/build/server

BUILD SUCCESSFUL
Total time: 1 second
e280635@11-6-66-8:~/Desktop/P1-ejb$ █
```

Ejercicio número 2:

Modificar el servlet `ProcesaPago` para que acceda al EJB local. Para ello, modificar el archivo `ProcesaPago.java` de la siguiente manera:

Modificamos y compilamos el cliente.

```
e280635@11-6-66-8:~/Desktop/P1-ejb$ ant compilar-cliente
Buildfile: /home/alumnos/e280635/Desktop/P1-ejb/build.xml

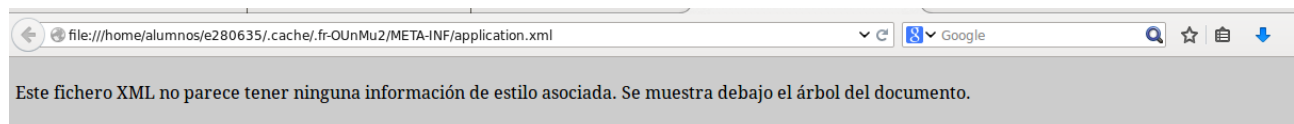
montar-jerarquia:

compilar-cliente:
[javac] Compiling 10 source files to /home/alumnos/e280635/Desktop/P1-ejb/build/client/WEB-INF/classes

BUILD SUCCESSFUL
Total time: 1 second
e280635@11-6-66-8:~/Desktop/P1-ejb$ ant empaquetar-cliente
```

Cuestión número 2:

Editar el archivo `application.xml` y comprobar su contenido. Verifique el contenido de todos los archivos `.jar` / `.war` / `.ear` que se han construido hasta el momento (empleando el comando `jar -tvf`). Anote sus comentarios en la memoria.



Este fichero XML no parece tener ninguna información de estilo asociada. Se muestra debajo el árbol del documento.

```
- <application version="5" xsi:schemaLocation="http://java.sun.com/xml/ns/javaee http://java.sun.com/xml/ns/javaee/application_5.xsd">
  <display-name>P1-ejb</display-name>
  - <module>
    <ejb>P1-ejb.jar</ejb>
  </module>
  - <module>
    - <web>
      <web-uri>P1-ejb-cliente.war</web-uri>
      <context-root>/P1-ejb-cliente</context-root>
    </web>
  </module>
</application>
```

```
e280635@11-6-66-8:~/Desktop/P1-ejb/dist$ jar -tvf P1-ejb.ear
0 Tue Feb 17 10:47:52 CET 2015 META-INF/
103 Tue Feb 17 10:47:50 CET 2015 META-INF/MANIFEST.MF
508 Sat Feb 11 23:33:00 CET 2012 META-INF/application.xml
20929 Tue Feb 17 10:45:18 CET 2015 P1-ejb-cliente.war
7016 Tue Feb 17 10:31:28 CET 2015 P1-ejb.jar
e280635@11-6-66-8:~/Desktop/P1-ejb/dist$
```

Ejercicio número 3:

Preparar los PCs con el esquema descrito y realizar el despliegue de la aplicación:

- Editar el archivo build.properties para que las propiedades as.host.client y as.host.server contengan la dirección IP del servidor de aplicaciones.
- Editar el archivo postgresql.properties para la propiedad db.client.host y db.host contengan las direcciones IP adecuadas para que el servidor de aplicaciones se conecte al postgresql, ambos estando en servidores diferentes.

Desplegar la aplicación de empresa
ant desplegar

The screenshot shows the GlassFish Server Open Source Edition web console. The browser address bar displays `https://10.5.7.2:4848/common/index.jsf`. The console header shows the user as 'admin', domain as 'domain1', and server as '10.5.7.2'. The left sidebar contains a tree view of the server structure, with 'P1-ejb' selected under 'Applications'. The main content area is titled 'Edit Application' and shows the configuration for the 'P1-ejb' application. The configuration includes fields for Name, Status, Virtual Servers, Java Web Start, Location, Deployment Order, Libraries, and Description. Below the configuration fields is a table titled 'Modules and Components (9)'.

Module Name	Engines	Component Name	Type	Action
P1-ejb-cliente.war	[web]	-----	-----	Launch
P1-ejb-cliente.war		default	Servlet	
P1-ejb-cliente.war		jsp	Servlet	
P1-ejb-cliente.war		DelPagos	Servlet	
P1-ejb-cliente.war		ProcesaPago	Servlet	
P1-ejb-cliente.war		GetPagos	Servlet	
P1-ejb-cliente.war		ComienzaPago	Servlet	
P1-ejb.jar	[ejb, weld]	-----	-----	
P1-ejb.jar		VisaDAOBean	StatelessSessionBean	

Como vemos se despliega correctamente.

Ejercicio número 4:

Comprobar el correcto funcionamiento de la aplicación mediante llamadas directas a través de las páginas `pago.html` y `testbd.jsp` (sin `directconnection`) . Realice un pago. Lístelo. Elimínelo. Téngase en cuenta que la aplicación se habrá desplegado bajo la ruta `/P1-ejb-cliente`

10.5.7.2:8080/P1-ejb-cliente/procesapago

Pago con tarjeta

Pago realizado con éxito. A continuación se muestra el comprobante del mismo:

idTransaccion: 1
idComercio: 2
importe: 3.0
codRespuesta: 000
idAutorizacion: 1

[Volver al comercio](#)

10.5.7.2:8080/P1-ejb-cliente/getpagos

Pago con tarjeta

Lista de pagos del comercio 2

idTransaccion	Importe	codRespuesta	idAutorizacion
1	3.0	000	1

[Volver al comercio](#)

Prácticas de Sistemas Informáticos II

```
e281377@11-5-66-7: ~
Archivo Editar Ver Buscar Terminal Ayuda
idautorizacion | idtransaccion | codrespuesta | importe | idcomercio |
numerotarjeta | fecha
-----+-----+-----+-----+-----+
1 | 1 | 000 | 3 | 2 |
1111 2222 3333 4444 | 2015-02-24 01:29:56.393061
(1 row)
(END)
```

Hacemos la transacción.

10.5.7.2:8080/P1-ejb-cliente/delpagos

Pago con tarjeta

Se han borrado 1 pagos correctamente para el comercio 2

[Volver al comercio](#)

Prácticas de Sistemas Informáticos II

```
e281377@11-5-66-7: ~
Archivo Editar Ver Buscar Terminal Ayuda
idautorizacion | idtransaccion | codrespuesta | importe | idcomercio | numerota
rjeta | fecha
-----+-----+-----+-----+-----+
(0 rows)
(END)
```

Borramos el pago.

Ejercicio número 5:

Modificar la aplicación VISA para soportar el campo saldo:

La única forma de probar que esto está hecho es verlo en el código o copiar aquí muchas líneas y como el formato del código se pierde al traerlo a un documento pues esta parte es mejor comprobarla en el código. Realmente la forma de probarlo es el ejercicio 6 así que a él nos remitimos.

Ejercicio número 6:

Desplegar y probar la nueva aplicación creada.

- Probar a realizar pagos correctos. Comprobar que disminuye el saldo de las tarjetas sobre las que realice operaciones. Añadir a la memoria las evidencias obtenidas.



Pago con tarjeta

Pago realizado con éxito. A continuación se muestra el comprobante del mismo:

idTransaccion: 1
idComercio: 1
importe: 999.0
codRespuesta: 000
idAutorizacion: 1

[Volver al comercio](#)

Prácticas de Sistemas Informáticos II

Realizamos un pago con la cuenta Jose Garcia con un id de transacción y de comercio de 1 y un importe de 999, la cuenta tiene un saldo de 1000, por lo tanto no nos debe saltar ningún error ya que el saldo la transacción se puede llevar a cabo porque tiene suficiente dinero y además al ser la primera operación que se realiza no puede coincidir con otra transacción anterior.

```
visa=# select * from pago;
 idautorizacion | idtransaccion | codrespuesta | importe | idcomercio | numerotarjeta | fecha
-----+-----+-----+-----+-----+-----+-----
1 | 1 | 000 | 999 | 1 | 1111 2222 3333 4444 | 2015-02-26 09:50:22.907218
(1 row)
```

Como vemos el pago se realiza correctamente.

```
visa=# select * from tarjeta where codigoverificacion = '123';
 numerotarjeta | titular | validadesde | validahasta | codigoverificacion | saldo
-----+-----+-----+-----+-----+-----
1111 2222 3333 4444 | Jose Garcia | 11/08 | 11/15 | 123 | 1
(1 row)
```

Y en efecto el saldo de la cuenta cambia.

- Realice una operación con identificador de transacción y de comercio duplicados. Compruebe que el saldo de la tarjeta especificada en el pago no se ha variado.

Proceso de un pago

Id Transacción:

Id Comercio:

Importe:

Numero de visa:

Titular:

Fecha Emisión:

Fecha Caducidad:

CVV2:

Modo debug: ☐ True ☐ False

Direct Connection: ☐ True ☐ False

Use Prepared: ☒ True ☐ False

Pago con tarjeta

Pago incorrecto

Prácticas de Sistemas Informáticos II

En este caso hacemos una transacción con un id de transacción y de comercio repetido, ya que existe un pago con los mismos datos la transacción no se realiza y los datos de la cuenta no cambian.

```
visa=# select * from tarjeta where codigoverificacion = '123';
      numerotarjeta | titular | validadesde | validahasta | codigoverificacion | saldo
-----+-----+-----+-----+-----+-----
 1111 2222 3333 4444 | Jose Garcia | 11/08 | 11/15 | 123 | 1
(1 row)
```

Para probarlo mas a fondo hemos hecho un pago con los id de transacción y de comercio correctos pero con la cantidad del pago superior al saldo, es decir que no se pueda realizar el pago ya que la cuenta no tiene suficiente dinero. Y en efecto:

← 10.5.7.2:8080/P1-ejb-cliente/procesapago

Pago con tarjeta

No hay suficiente dinero

Prácticas de Sistemas Informáticos II

Ejercicio número 7:

En la máquina virtual donde se encuentra el servidor de aplicaciones (10.X.Y.2), declare manualmente la factoría de conexiones empleando la consola de administración, tal y como se adjunta en la Figura 4

The screenshot shows the GlassFish Server Open Source Edition administration console. The browser address bar displays `https://10.5.7.2:4848/common/index.jsf`. The page title is "New JMS Connection Factory". The left sidebar shows a tree view with "Resources" expanded, and "JMS Resources" > "Connection Factories" selected. The main content area contains the following configuration fields:

- General Settings**
 - JNDI Name: `jms/VisaConnectionFactory`
 - Resource Type: `javax.jms.QueueConnectionFactory`
 - Description: `Factoría de conexiones a la cola de pagos`
 - Status: ☒ Enabled
- 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
 - On Any Failure: ☐ Close All Connections
 - Transaction Support: `Level`
 - Connection Validation: ☐ Required

Ejercicio número 8:

En la máquina virtual donde se encuentra el servidor de aplicaciones (10.X.Y.2), declare manualmente la conexión empleando la consola de administración, tal y como se adjunta en la Figura 5.

The screenshot shows the GlassFish Server Open Source Edition administration console. The browser address bar displays `https://10.5.7.2:4848/common/index.jsf`. The page title is "Edit JMS Destination Resource". The left sidebar shows a tree view with "Resources" expanded, and "JMS Resources" > "Destination Resources" selected. The main content area contains the following configuration fields:

- General Settings**
 - JNDI Name: `jms/VisaPagosQueue`
 - Physical Destination Name: `VisaPagosQueue`
 - Resource Type: `javax.jms.Queue`
 - Deployment Order: `100`
 - Description: `Cola de pagos VISA`
 - Status: ☒ Enabled

Ejercicio número 9:

Modifique el fichero sun-ejb-jar.xml para que el MDB conecte adecuadamente a su connection factory

Básicamente es añadir esta línea dentro del build:jms/VisaConnectionFactory

Incluya en la clase VisaCancelacionJMSBean:

- Consulta SQL necesaria para actualizar el código de respuesta a valor 999, de aquella autorización existente en la tabla de pagos cuyo idAutorizacion coincida con lo recibido por el mensaje.

update pago set codRespuesta=999 where idAutorizacion=?

En este caso el id de autorización toma el valor del campo inMessage, es decir, se nos pasa como argumento.

- Consulta SQL necesaria para rectificar el saldo de la tarjeta que realizó el pago

update tarjeta set saldo = ? where numeroTarjeta = ?

Aparte de esta consulta se ejecutan 3 consultas mas para que esta tenga sentido:

1. Una que dado el idAutorizacion nos devuelve el numero de la tarjeta.
2. Otra que dado el numero de tarjeta nos da su saldo.
3. Y la ultima que dado el idAutorizacion nos devuelve el pago que se ha hecho.

El sentido de esto es que con el numero de tarjeta que nos devuelve la primera, obtenemos el saldo de la cuenta mediante la segunda consulta y la suma del saldo y el importe (que obtenemos en la tercera) es igual al saldo que debe tener la tarjeta del cliente una vez se halla cancelado el pago con éxito.

- Método onMessage() que implemente ambas actualizaciones. Para ello tome de ejemplo el código SQL de ejercicios anteriores, de modo que se use un prepared statement que haga bind del idAutorizacion para cada mensaje recibido.

Básicamente este método ejecuta las 2 consultas anteriores para que la cancelación de la operación tenga sentido.

Ejercicio número 10:

Implemente ambos métodos en el cliente proporcionado. Deje comentado el método de acceso por JNDI. Indique en la memoria de prácticas qué ventajas podrían tener uno u otro método.

La principal ventaja de usar JMS es que no conlleva una sobrecarga en el rendimiento de la aplicación ya que se lleva a cabo en tiempo de compilación. Por otro lado esto exige saber de antemano el nombre del recurso.

Por otro lado JNDI tiene la ventaja de que es más flexible ya que soporta distintas implementaciones a la hora de buscar el recurso que se desea.

Ejercicio número 11:

Compruebe en la consola de administración del Glassfish que, efectivamente, los recursos se han creado automáticamente. Revise el fichero `jms.xml` y anote en la memoria de prácticas cuál es el comando equivalente para crear una cola JMS usando la herramienta `asadmin`.



Por lo que se lee del fichero `jms.xml` habría que ejecutar el el comando `asadmin` con los comandos que se muestran en la imagen, `--user ...`, `--passwordfile ...`, etc.

Ejercicio número 12:

Modifique el cliente, `VisaQueueMessageProducer.java`, implementando el envío de `args[0]` como mensaje de texto (consultar los apéndices). Ejecute el cliente en el PC del laboratorio mediante el comando:

Escribimos los 2 comando que nos piden, el primero manda un mensaje a la cola, al estar esta parada no se ejecutar, pero utilizando `-browse` podemos ver los datos que están en la cola de mensajes para que sean procesados.

```
rober@rober-PC:~/Escritorio/P1-jms$ /home/rober/src/glassfish4.0/glassfish/bin/appclient -targetserver 10.5.7.2 -client dist/clientjms/P1-jms-clientjms.jar idAutorizacion
mar 02, 2015 6:00:02 PM org.hibernate.validator.internal.util.Version <clinit>
INFO: HV000001: Hibernate Validator 5.0.0.Final
mar 02, 2015 6:00:02 PM com.sun.messaging.jms.ra.ResourceAdapter start
INFORMACIÓN: MQJMSRA_RA1101: GlassFish MQ JMS Resource Adapter: Versión: 5.0 (Build 14-e) Compilación: April 12 2013 0104
mar 02, 2015 6:00:02 PM com.sun.messaging.jms.ra.ResourceAdapter start
INFORMACIÓN: MQJMSRA_RA1101: GlassFish MQ JMS Resource Adapter starting: broker is REMOTE, connection mode is TCP
mar 02, 2015 6:00:02 PM com.sun.messaging.jms.ra.ResourceAdapter start
INFORMACIÓN: MQJMSRA_RA1101: GlassFish MQ JMS Resource Adapter Started:REMOTE
rober@rober-PC:~/Escritorio/P1-jms$ /home/rober/src/glassfish4.0/glassfish/bin/appclient -targetserver 10.5.7.2 -client dist/clientjms/P1-jms-clientjms.jar -browse
mar 02, 2015 6:00:17 PM org.hibernate.validator.internal.util.Version <clinit>
INFO: HV000001: Hibernate Validator 5.0.0.Final
mar 02, 2015 6:00:17 PM com.sun.messaging.jms.ra.ResourceAdapter start
INFORMACIÓN: MQJMSRA_RA1101: GlassFish MQ JMS Resource Adapter: Versión: 5.0 (Build 14-e) Compilación: April 12 2013 0104
mar 02, 2015 6:00:17 PM com.sun.messaging.jms.ra.ResourceAdapter start
INFORMACIÓN: MQJMSRA_RA1101: GlassFish MQ JMS Resource Adapter starting: broker is REMOTE, connection mode is TCP
mar 02, 2015 6:00:17 PM com.sun.messaging.jms.ra.ResourceAdapter start
INFORMACIÓN: MQJMSRA_RA1101: GlassFish MQ JMS Resource Adapter Started:REMOTE
Mensajes en cola:
idAutorizacion
rober@rober-PC:~/Escritorio/P1-jms$
```

Como se puede apreciar añadimos a la cola de mensajes el mensaje "idAutorización" (si vale no es un número pero va igual). Y como se ve al poner `-browse` en la cola de mensajes se encuentra "idAutorización".

A continuación, volver a habilitar la ejecución del MDB y realizar los siguientes pasos:

- Realice un pago con la aplicación web

Pago con tarjeta

Proceso de un pago

Id Transacción:

Id Comercio:

Importe:

Numero de visa:

Titular:

Fecha Emisión:

Fecha Caducidad:

CVV2:

Modo debug: ☐ True ☐ False

Direct Connection: ☐ True ☐ False

Use Prepared: ☐ True ☐ False

Básicamente realizamos un pago con la tarjeta de Jose Garcia

- Obtenga evidencias de que se ha realizado

Pago con tarjeta

Pago realizado con éxito. A continuación se muestra el comprobante del mismo:

idTransaccion: 1
idComercio: 2
importe: 3.0
codRespuesta: 000
idAutorizacion: 1

[Volver al comercio](#)

Prácticas de Sistemas Informáticos II

```
rober@rober-PC: ~
idautorizacion | idtransaccion | codrespuesta | importe | idcomercio | numerotarjeta | fecha
-----+-----+-----+-----+-----+-----+-----
(1 row)
1 | 1 | 000 | 3 | 2 | 1111 2222 3333 4444 | 2015-03-02 12:52:54.345106
```

Y en efecto como se puede observar el pago se ha realizado correctamente, ya que nos aparece tanto en la ventana como en la base de datos

- Cancelelo con el cliente

```
rober@rober-PC:~/Escritorio/P1-jms$ /home/rober/src/glassfish4.0/glassfish/bin/appclient
-targetserver 10.5.7.2 -client dist/clientjms/P1-jms-clientjms.jar 1
mar 02, 2015 9:58:33 PM org.hibernate.validator.internal.util.Version <clinit>
INFO: HV000001: Hibernate Validator 5.0.0.Final
mar 02, 2015 9:58:34 PM com.sun.messaging.jms.ra.ResourceAdapter start
INFORMACIÓN: MQJMSRA_RA1101: GlassFish MQ JMS Resource Adapter: Versión: 5.0 (Build 14-
e) Compilación: April 12 2013 0104
mar 02, 2015 9:58:34 PM com.sun.messaging.jms.ra.ResourceAdapter start
INFORMACIÓN: MQJMSRA_RA1101: GlassFish MQ JMS Resource Adapter starting: broker is REMOTE
, connection mode is TCP
mar 02, 2015 9:58:34 PM com.sun.messaging.jms.ra.ResourceAdapter start
INFORMACIÓN: MQJMSRA_RA1101: GlassFish MQ JMS Resource Adapter Started:REMOTE
rober@rober-PC:~/Escritorio/P1-jms$
```

Para cancelarlo simplemente introducimos el id de la transacción que hemos llevado a cabo y se borrara.

- Obtenga evidencias de que se ha cancelado y de que el saldo se ha rectificado

```
visa=# select * from pago;
idautorizacion | idtransaccion | codrespuesta | importe | idcomercio | numerotarjeta | fecha
-----+-----+-----+-----+-----+-----+-----
(1 row)
1 | 1 | 999 | 3 | 2 | 1111 2222 3333 4444 | 2015-03-02 12:52:54.345106
```

```
visa=# select * from tarjeta where codigoverificacion='123';
numerotarjeta | titular | validadesde | validahasta | codigoverificacion | saldo
-----+-----+-----+-----+-----+-----
(1 row)
1111 2222 3333 4444 | Jose Garcia | 11/08 | 11/15 | 123 | 1000
```

visa=#

Como se puede observar tanto el “codrespuesta” se ha modificado y el saldo de la cuenta vuelve a ser el mismo por tanto la operación se ha realizado con éxito..