		Escuela Politécnica Superior Ingeniería Informática Prácticas de Sistemas Informáticos 2			
Grupo	2401	Práctica	1B	Fecha	09/03/2018
Alumno		Alcover, Couso, Roberto			
Alumno		Aced, Fuentes, Emilio Samuel			

Práctica 1B: Arquitectura JAVA EE (Segund Parte)

Cuestión número 1:

Abrir el archivo VisaDAOLocal.java y comprobar la definición de dicha interfaz. Anote en la memoria comentarios sobre las librerías Java EE importadas y las anotaciones utilizadas:
package ssii2.visa;

```
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;
```

@Local

Ejercicio número 1:

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

Hacer que la clase implemente la interfaz local y convertirla en un EJB stateless mediante la anotación Stateless.

Eliminar el constructor por defecto de la clase.

Ajustar los métodos getPagos() a la interfaz definida en VisaDAOLocal.

Ver código modificado en el fichero: P1-ejb/src/server/ssii2/visa//VisaDAOLocal.java.

Ejercicio número 2:

Modificar el servlet `ProcesaPago` para que acceda al EJB local. Para ello, modificar el archivo `ProcesaPago.java` de la siguiente manera: En la sección de preproceso, añadir las siguientes importaciones de clases que se van a utilizar:

Se deberán eliminar estas otras importaciones que dejan de existir en el proyecto:

Añadir como atributo de la clase el objeto proxy que permite acceder al EJB local, con su correspondiente anotación que lo declara como tal:

```
...
@EJB(name="VisaDAOBean", beanInterface=VisaDAOLocal.class) private VisaDAOLocal dao;
```

```
...
```

En el cuerpo del servlet, eliminar la declaración de la instancia del antiguo webservice `VisaDAOWS`, así como el código necesario para obtener la referencia remota
Eliminar también las referencias a `BindingProvider`.

Importante: Esta operación deberá ser realizada para todos los servlets del proyecto que hagan uso del antiguo `VisaDAOWS`. Verifique también posibles errores de compilación y ajustes necesarios en el código al cambiar la interfaz del antiguo `VisaDAOWS` (en particular, el método `getPagos()`).

Todas estas modificaciones se han realizado en los ficheros:

```
P1-ejb/src/client/sii2/controlador/DelPagos.java
P1-ejb/src/client/sii2/controlador/GetPagos.java
P1-ejb/src/client/sii2/controlador/ProcesaPago.java
```

Cuestión número 2:

Abrir 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

P1-ejb.ear:

```
0 Tue Feb 27 17:33:48 CET 2018 META-INF/
 104 Tue Feb 27 17:33:46 CET 2018 META-INF/MANIFEST.MF
 508 Sat Feb 11 23:33:00 CET 2012 META-INF/application.xml
20972 Tue Feb 27 17:29:38 CET 2018 P1-ejb-cliente.war
7208 Tue Feb 27 17:17:44 CET 2018 P1-ejb.jar
```

P1-ejb-cliente.war:

```
0 Tue Feb 27 17:29:40 CET 2018 META-INF/
 104 Tue Feb 27 17:29:38 CET 2018 META-INF/MANIFEST.MF
   0 Tue Feb 27 17:29:38 CET 2018 WEB-INF/
   0 Tue Feb 27 17:25:26 CET 2018 WEB-INF/classes/
   0 Tue Feb 27 17:29:08 CET 2018 WEB-INF/classes/ssii2/
   0 Tue Feb 27 17:29:08 CET 2018 WEB-INF/classes/ssii2/controlador/
   0 Tue Feb 27 17:29:08 CET 2018 WEB-INF/classes/ssii2/filtros/
   0 Tue Feb 27 17:29:08 CET 2018 WEB-INF/classes/ssii2/visa/
   0 Tue Feb 27 17:29:08 CET 2018 WEB-INF/classes/ssii2/visa/error/
   0 Tue Feb 27 17:10:06 CET 2018 WEB-INF/lib/
   0 Tue Feb 27 17:29:38 CET 2018 error/
2844 Tue Feb 27 17:27:48 CET 2018 WEB-INF/classes/ssii2/controlador/ComienzaPago.class
1513 Tue Feb 27 17:27:48 CET 2018 WEB-INF/classes/ssii2/controlador/DelPagos.class
```

1365 Tue Feb 27 17:29:08 CET 2018 WEB-INF/classes/ssii2/controlador/GetPagos.class
4919 Tue Feb 27 17:29:08 CET 2018 WEB-INF/classes/ssii2/controlador/ProcesaPago.class
1894 Tue Feb 27 17:25:26 CET 2018 WEB-INF/classes/ssii2/controlador/ServletRaiz.class
2608 Tue Feb 27 17:29:08 CET 2018 WEB-INF/classes/ssii2/filtros/CompruebaSesion.class
3170 Tue Feb 27 17:29:08 CET 2018 WEB-INF/classes/ssii2/visa/ValidadorTarjeta.class
616 Tue Feb 27 17:29:08 CET 2018 WEB-INF/classes/ssii2/visa/error/ErrorVisa.class
198 Tue Feb 27 17:29:08 CET 2018 WEB-INF/classes/ssii2/visa/error/ErrorVisaCVV.class
209 Tue Feb 27 17:29:08 CET 2018

WEB-INF/classes/ssii2/visa/error/ErrorVisaFechaCaducidad.class

207 Tue Feb 27 17:29:08 CET 2018 WEB-INF/classes/ssii2/visa/error/ErrorVisaFechaEmision.class
201 Tue Feb 27 17:29:08 CET 2018 WEB-INF/classes/ssii2/visa/error/ErrorVisaNumero.class
202 Tue Feb 27 17:29:08 CET 2018 WEB-INF/classes/ssii2/visa/error/ErrorVisaTitular.class
6174 Tue Feb 27 17:29:38 CET 2018 WEB-INF/web.xml
455 Tue Feb 27 17:29:38 CET 2018 borradoerror.jsp
501 Tue Feb 27 17:29:38 CET 2018 borradook.jsp
509 Tue Feb 27 17:29:38 CET 2018 cabecera.jsp
283 Tue Feb 27 17:29:38 CET 2018 error/muestraerror.jsp
2729 Tue Feb 27 17:29:38 CET 2018 formdatosvisa.jsp
1257 Tue Feb 27 17:29:38 CET 2018 listapagos.jsp
1178 Tue Feb 27 17:29:38 CET 2018 pago.html
1142 Tue Feb 27 17:29:38 CET 2018 pagoexito.jsp
104 Tue Feb 27 17:29:38 CET 2018 pie.html
5011 Tue Feb 27 17:29:38 CET 2018 testbd.jsp

```
e321075@localhost:~/Desktop/P1-ejb$ jar -tvf dist/P1-ejb.ear
0 Tue Feb 27 17:33:48 CET 2018 META-INF/
104 Tue Feb 27 17:33:46 CET 2018 META-INF/MANIFEST.MF
508 Sat Feb 11 23:33:00 CET 2012 META-INF/application.xml
20972 Tue Feb 27 17:29:38 CET 2018 P1-ejb-cliente.war
7208 Tue Feb 27 17:17:44 CET 2018 P1-ejb.jar
e321075@localhost:~/Desktop/P1-ejb$ jar -tvf dist/client/P1-ejb-cliente.war
0 Tue Feb 27 17:29:40 CET 2018 META-INF/
104 Tue Feb 27 17:29:38 CET 2018 META-INF/MANIFEST.MF
0 Tue Feb 27 17:29:38 CET 2018 WEB-INF/
0 Tue Feb 27 17:25:26 CET 2018 WEB-INF/classes/
0 Tue Feb 27 17:29:08 CET 2018 WEB-INF/classes/ssii2/
0 Tue Feb 27 17:29:08 CET 2018 WEB-INF/classes/ssii2/controlador/
0 Tue Feb 27 17:29:08 CET 2018 WEB-INF/classes/ssii2/filtros/
0 Tue Feb 27 17:29:08 CET 2018 WEB-INF/classes/ssii2/visa/
0 Tue Feb 27 17:29:08 CET 2018 WEB-INF/classes/ssii2/visa/error/
0 Tue Feb 27 17:10:06 CET 2018 WEB-INF/lib/
0 Tue Feb 27 17:29:38 CET 2018 error/
2844 Tue Feb 27 17:27:48 CET 2018 WEB-INF/classes/ssii2/controlador/ComienzaPago.class
1513 Tue Feb 27 17:27:48 CET 2018 WEB-INF/classes/ssii2/controlador/DelPagos.class
1365 Tue Feb 27 17:29:08 CET 2018 WEB-INF/classes/ssii2/controlador/GetPagos.class
4919 Tue Feb 27 17:29:08 CET 2018 WEB-INF/classes/ssii2/controlador/ProcesaPago.class
1894 Tue Feb 27 17:25:26 CET 2018 WEB-INF/classes/ssii2/controlador/ServletRaiz.class
2608 Tue Feb 27 17:29:08 CET 2018 WEB-INF/classes/ssii2/filtros/CompruebaSesion.class
3170 Tue Feb 27 17:29:08 CET 2018 WEB-INF/classes/ssii2/visa/ValidadorTarjeta.class
616 Tue Feb 27 17:29:08 CET 2018 WEB-INF/classes/ssii2/visa/error/ErrorVisa.class
198 Tue Feb 27 17:29:08 CET 2018 WEB-INF/classes/ssii2/visa/error/ErrorVisaCVV.class
209 Tue Feb 27 17:29:08 CET 2018 WEB-INF/classes/ssii2/visa/error/ErrorVisaFechaCaducidad.class
207 Tue Feb 27 17:29:08 CET 2018 WEB-INF/classes/ssii2/visa/error/ErrorVisaFechaEmision.class
201 Tue Feb 27 17:29:08 CET 2018 WEB-INF/classes/ssii2/visa/error/ErrorVisaNumero.class
202 Tue Feb 27 17:29:08 CET 2018 WEB-INF/classes/ssii2/visa/error/ErrorVisaTitular.class
6174 Tue Feb 27 17:29:38 CET 2018 WEB-INF/web.xml
455 Tue Feb 27 17:29:38 CET 2018 borradoerror.jsp
501 Tue Feb 27 17:29:38 CET 2018 borradook.jsp
509 Tue Feb 27 17:29:38 CET 2018 cabecera.jsp
283 Tue Feb 27 17:29:38 CET 2018 error/muestraerror.jsp
2729 Tue Feb 27 17:29:38 CET 2018 formdatosvisa.jsp
1257 Tue Feb 27 17:29:38 CET 2018 listapagos.jsp
1178 Tue Feb 27 17:29:38 CET 2018 pago.html
1142 Tue Feb 27 17:29:38 CET 2018 pagoexito.jsp
104 Tue Feb 27 17:29:38 CET 2018 pie.html
5011 Tue Feb 27 17:29:38 CET 2018 testbd.jsp
```

P1-ejb.jar:

```
e321075@localhost:~/Desktop/P1-ejb$ jar -tvf dist/server/P1-ejb.jar
0 Tue Feb 27 17:17:44 CET 2018 META-INF/
104 Tue Feb 27 17:17:42 CET 2018 META-INF/MANIFEST.MF
0 Tue Feb 27 17:17:26 CET 2018 ssii2/
0 Tue Feb 27 17:17:26 CET 2018 ssii2/visa/
0 Tue Feb 27 17:17:26 CET 2018 ssii2/visa/dao/
255 Tue Feb 27 17:17:44 CET 2018 META-INF/sun-ejb-jar.xml
1464 Tue Feb 27 17:17:26 CET 2018 ssii2/visa/PagoBean.class
856 Tue Feb 27 17:17:26 CET 2018 ssii2/visa/TarjetaBean.class
593 Tue Feb 27 17:17:26 CET 2018 ssii2/visa/VisaDAOLocal.class
2051 Tue Feb 27 17:17:26 CET 2018 ssii2/visa/dao/DBTester.class
7105 Tue Feb 27 17:17:26 CET 2018 ssii2/visa/dao/VisaDAOBean.class
```

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

Pago realizado por testbd.jsp

Pago con tarjeta

Proceso de un pago

Id Transacción:	<input type="text" value="10"/>
Id Comercio:	<input type="text" value="1"/>
Importe:	<input type="text" value="12"/>
Numero de visa:	<input type="text" value="2347 4840 5058 7931"/>
Titular:	<input type="text" value="Gabriel Avila Locke"/>
Fecha Emisión:	<input type="text" value="11/09"/>
Fecha Caducidad:	<input type="text" value="01/20"/>
CVV2:	<input type="text" value="207"/>
Modo debug:	<input type="radio"/> True <input type="radio"/> False
Direct Connection:	<input type="radio"/> True <input type="radio"/> False
Use Prepared:	<input type="radio"/> True <input type="radio"/> False
<input type="button" value="Pagar"/>	

Pago con tarjeta

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

idTransaccion: 10
idComercio: 1
importe: 12.0
codRespuesta: 000
idAutorizacion: 3

[Volver al comercio](#)

Prácticas de Sistemas Informáticos II


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

Módulos de la aplicación:

Modules and Components (9)			
Module Name	Engines	Component Name	Type
P1-ejb-cliente.war	[web]	-----	-----
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

Realizamos un pago:



The screenshot shows a web browser window with the address bar displaying "10.1.1.1:8080/P1-ejb-cliente-remoto/procesapago". The page title is "Pago con tarjeta". The main content area displays a confirmation message: "Pago realizado con éxito. A continuación se muestra el comprobante del mismo:". Below this, the following transaction details are listed: idTransaccion: 1, idComercio: 2, importe: 1.0, codRespuesta: 000, and idAutorizacion: 10. A link labeled "Volver al comercio" is provided. At the bottom, the footer text "Prácticas de Sistemas Informáticos II" is visible.

← → ↻ ⓘ 10.1.1.1:8080/P1-ejb-cliente-remoto/procesapago

Pago con tarjeta

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

idTransaccion: 1
idComercio: 2
importe: 1.0
codRespuesta: 000
idAutorizacion: 10

[Volver al comercio](#)

Prácticas de Sistemas Informáticos II

Ejercicio número 5:

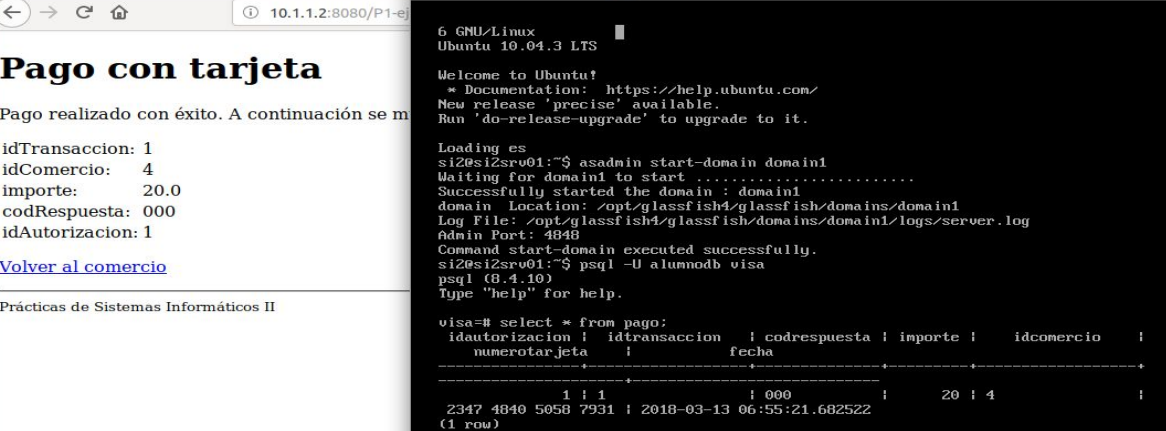
Realizar los cambios indicados en P1-ejb-servidor-remoto y preparar los PCs con el esquema de máquinas virtuales indicado. Compilar, empaquetar y desplegar de nuevo la aplicación P1-ejb como servidor de EJB remotos de forma similar a la realizada en el Ejercicio 3 con la Figura 2 como entorno de despliegue. Esta aplicación tendrá que desplegarse en la máquina virtual del PC2

En el código.

Ejercicio número 6:

Realizar los cambios comentados en la aplicación P1-base para convertirla en P1-ejb-cliente-remoto. Compilar, empaquetar y desplegar de nuevo la aplicación en otra máquina virtual distinta a la de la aplicación servidor, es decir, esta aplicación cliente estará desplegada en la MV del PC1 tal y como se muestra en el diagrama de despliegue de la Figura 2.

Conectarse a la aplicación cliente y probar a realizar un pago. Comprobar los resultados e incluir en la memoria evidencias de que el pago ha sido realizado de forma correcta



The screenshot shows a terminal window with a dark background. The top part of the terminal shows the Ubuntu login banner and a series of commands and outputs: "6 GNU/Linux", "Ubuntu 10.04.3 LTS", "Welcome to Ubuntu!", "Loading es", "si2@si2srv01:~\$ asadmin start-domain domain1", "Waiting for domain1 to start", "Successfully started the domain : domain1", "domain Location: /opt/glassfish4/glassfish/domains/domain1", "Log File: /opt/glassfish4/glassfish/domains/domain1/logs/server.log", "Admin Port: 4848", "Command start-domain executed successfully.", "si2@si2srv01:~\$ psql -U alumodb visa", "psql (9.4.10)", "Type 'help' for help.", "visa=# select * from pago:", "idautorizacion | idtransaccion | codrespuesta | importe | idcomercio |", "numerotarjeta | fecha", "1 | 1 | 000 | 20 | 4", "2347 4840 5058 7931 | 2018-03-13 06:55:21.682522", "(1 row)".

← → ↻ ⓘ 10.1.1.2:8080/P1-e

Pago con tarjeta

Pago realizado con éxito. A continuación se m

idTransaccion: 1
idComercio: 4
importe: 20.0
codRespuesta: 000
idAutorizacion: 1

[Volver al comercio](#)

Prácticas de Sistemas Informáticos II

Cuestión número 7:

Modificar la aplicación VISA para soportar el campo saldo: Archivo TarjetaBean.java: Añadir el atributo saldo y sus métodos de acceso: private double saldo; Archivo VisaDAOBean.java:

Importar la definición de la excepción EJBException que debe lanzar el servlet para indicar que se debe realizar un rollback: import javax.ejb.EJBException;

Declarar un prepared statement para recuperar el saldo de una tarjeta de la base de datos.

Declarar un prepared statement para insertar el nuevo saldo calculado en la base de datos.

Modificar el método realizaPago con las siguientes acciones:

- o Recuperar el saldo de la tarjeta a través del prepared statement declarado anteriormente.

- o Comprobar si el saldo es mayor o igual que el importe de la operación. Si no lo es, retornar denegando el pago (idAutorizacion= null y pago retornado=null)

- o Si el saldo es suficiente, decrementarlo en el valor del importe del pago y actualizar el registro de la tarjeta para reflejar el nuevo saldo mediante el prepared statement declarado anteriormente.

- o Si lo anterior es correcto, ejecutar el proceso de inserción del pago y obtención del idAutorizacion, tal como se realizaba en la práctica anterior (este código ya debe estar programado y no es necesario modificarlo).

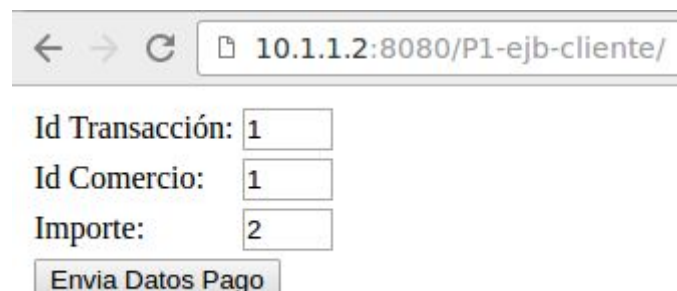
- o En caso de producirse cualquier error a lo largo del proceso (por ejemplo, si no se obtiene el idAutorizacion porque la transacción está duplicada), lanzar una excepción EJBException para retornar al cliente.

Modificar el servlet ProcesaPago para que capture la posible interrupción EJBException lanzada por realizaPago, y, en caso de que se haya lanzado, devuelva la página de error media En el código

Cuestión número 8:

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. 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.

Flujo del pago



← → ↻ 10.1.1.2:8080/P1-ejb-cliente/

Id Transacción:

Id Comercio:

Importe:

← → ↻ 10.1.1.2:8080/P1-ejb-cliente/comienzapago

Pago con tarjeta

Numero de visa: 2347 4840 5058 7931
Titular: Gabriel Avila Locke
Fecha Emisión: 11/09
Fecha Caducidad: 01/20
CVV2: 207

Id Transacción: 1
Id Comercio: 1
Importe: 2.0

Prácticas de Sistemas Informáticos II

Tora 2.1.3
File Edit View Tools Editor Window Help
alumnodb@visa.10.1.1.1:5432 [8.4.10]

Connecti...
Host
10.1.1.1

Directory
Result Execution plan Visualize Logging

```
select * from tarjeta where numerotarjeta = '2347 4840 5058 7931'
```

#	numerotarjeta	titular	validadesde	validahasta	codigoverificacion	saldo
1	2347 4840 5058 7931	Gabriel Avila Locke	11/09	01/20	207	1000

← → ↻ 10.1.1.2:8080/P1-ejb-cliente/procesapago

Pago con tarjeta

Pago realizado con éxito. A continuación:

idTransaccion: 1
idComercio: 1
importe: 2.0
codRespuesta: 000
idAutorizacion: 1

[Volver al comercio](#)

Prácticas de Sistemas Informáticos II

Tora 2.1.3
File Edit View Tools Editor Window Help
alumnodb@visa.10.1.1.1:5432 [8.4.10]

Connecti...
Host
10.1.1.1

Directory
Result Execution plan Visualize Logging

```
select * from tarjeta where numerotarjeta = '2347 4840 5058 7931'
```

#	numerotarjeta	titular	validadesde	validahasta	codigoverificacion	saldo
1	2347 4840 5058 7931	Gabriel Avila Locke	11/09	01/20	207	998

Flujo de pago incorrecto:

← → ↻ 10.1.1.2:8080/P1-ejb-cliente/pago.html

Id Transacción:
Id Comercio:
Importe:

← → ↻ 10.1.1.2:8080/P1-ejb-cliente/comienzapago

Pago con tarjeta

Numero de visa: 2347 4840 5058 7931
Titular: Gabriel Avila Locke
Fecha Emisión: 11/09
Fecha Caducidad: 01/20
CVV2: 207

Id Transacción: 1
Id Comercio: 1
Importe: 5.0

Tora 2.1.3
File Edit View Tools Editor Window Help
alumnodb@visa.10.1.1.1:5432 [8.4.10]

Connecti...
Host
10.1.1.1

Directory
Result Execution plan Visualize Logging

#	numerotarjeta	titular	validadesde	validahasta	codigoverificacion
1	2347 4840 5058 7931	Gabriel Avila Locke	11/09	01/20	207

10.1.1.2:8080/P1-ejb-cliente/procesapago

Pago con tarjeta

Pago incorrecto

Prácticas de Sistemas Informáticos II

Tora 2.1.3
File Edit View Tools Editor Window Help

alumnodb@visa.10.1.1.1:5432 [8.4.10]

alumnodb@visa.10.1.1.1:5432 SQL Editor - *Untitled

select * from tarjeta where numerotarjeta = '2347 4840 5058 7931'

Result Execution plan Visualize Logging

#	numerotarjeta	titular	validadesde	validahasta	codigoverificacion	saldo
1	2347 4840 5058 7931	Gabriel Avila Locke	11/09	01/20	207	998

Cuestión número 9:

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.

Captura de nuestro Glassfish con los valores requeridos.

Home About...

User: admin Role: domain1 Server: 10.1.1.2

GlassFish™ Server Open Source Edition

Total # of available updates : 1

Common Tasks

- Domain
 - server (Admin Server)
 - Clusters
 - Standalone Instances
- Nodes
- Applications
 - Lifecycle Modules
 - Monitoring Data
- Resources
 - Concurrent Resources
 - Connectors
 - JDBC
 - JMS Resources
 - Connection Factories
 - Destination Resources
 - JNDI
 - JavaMail Sessions
 - Resource Adapter Configs
- Configurations
 - default-config
 - server-config
- Update Tool

New JMS Connection Factory

The creation of a new Java Message Service (JMS) connection factory also creates a connector connection pool for the factory and a c

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
Minimum and initial number of connections maintained in the pool

Maximum Pool Size: 32 Connections
Maximum number of connections that can be created to satisfy client requests

Pool Resize Quantity: 2 Connections
Number of connections to be removed when pool idle timeout expires

Idle Timeout: 300 Seconds
Maximum time that connection can remain idle in the pool

Max Wait Time: 60000 Milliseconds
Amount of time caller waits before connection timeout is sent

On Any Failure: ☐ Close All Connections
Close all connections and reconnect on failure, otherwise reconnect only when used

Transaction Support:
Level of transaction support. Overwrite the transaction support attribute in the Resource Adapter in

Connection Validation: ☐ Required
Validate connections, allow server to reconnect in case of failure

Additional Properties (0)

Add Property Delete Properties

Select	Name	Value	Description
No items found.			

Ejercicio número 10:

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

Declaración manual de la conexión.

Edit JMS Destination Resource

Editing a Java Message Service (JMS) destination resource also modifies the associated admin object resource.

[Load Defaults](#)

JNDI Name: jms/VisaPagosQueue

Physical Destination Name * VisaPagosQueue

Destination name in the Message Queue broker. If the destination does not exist, it

Resource Type: * javax.jms.Queue

Deployment Order: 100

Specifies the loading order of the resource at server startup. Lower numbers are loa

Description: Cola de pagos VISA

Status: ☒ Enabled

Additional Properties (0)

[Add Property](#)

[Delete Properties](#)

Select	Name	Value	Description
No items found.			

Ejercicio número 11:

Modifique el fichero sun-ejb-jar.xml para que el MDB conecte adecuadamente a su connection factory Incluya en la clase VisaCancelacionJMSBean:

Ver la ruta: build/mdb/META-INF/sun-ejb-jar.xml

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.

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

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 binddel idAutorizacion para cada mensaje recibido.

Ver la ruta: P1-jms/src/mdb/ssii2/visa/VisaCancelacionJMSBean.java

Ejercicio número 12:

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.

Acceder en tiempo de ejecución al recurso nos libera de tener que modificar el código del cliente en caso que el mismo cambie de ubicación lo nos otorga transparencia.

Por otro lado, el acceso estático tiene una programación más sencilla.

Ejercicio número 13:

Automatice la creación de los recursos JMS (cola y connection factory) en el build.xml y jms.xml. Para ello, indique en jms.properties los nombres de ambos y el Physical Destination Name de la cola de acuerdo a los valores asignados en los ejercicios 7 y 8. Recuerde también asignar las direcciones IP adecuadas a las variables as.host.mdb (build.properties) y as.host.server (jms.properties).

Borre desde la consola de administración de Glassfish la connectionFactory y la cola creadas manualmente y ejecute:

```
cd P1-jms
ant todo
```

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.

Fichero jms.xml:

```
<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>
```

Comando:

```
asadmin --user admin --passwordfile passwordfile --host 10.1.1.1 --port 4848 create-jms-resource
--restype javax.jms.QueueConnectionFactory --enabled=true --property Name=Visa
jms/VisaPagosQueue
```

Ejercicio número 14:

Importante: Detenga la ejecución del MDB con la consola de administración para poder realizar satisfactoriamente el siguiente ejercicio (check de 'Enabled' en Applications/P1-jms-mdb y guardar los cambios).

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:

```
/usr/local/glassfish-4.1.1/glassfish/bin/appclient -targetserver 10.X.Y.Z - client  
dist/clientjms/P1-jms-clientjms.jar idAutorizacion
```

Donde 10.X.Y.Z representa la dirección IP de la máquina virtual en cuyo servidor de aplicaciones se encuentra desplegado el MDB.

Para garantizar que el comando funcione correctamente es necesario fijar la variable (web console->Configurations->server-config->JavaMessageService->JMSHosts->default_JMS_host) que toma el valor "localhost" por la dirección IP de dicha máquina virtual.

El cambio se puede llevar a cabo desde la consola de administración. Será necesario reiniciar el servidor de aplicaciones para que surja efecto.

Verifique el contenido de la cola ejecutando:

```
/usr/local/glassfish-4.1.1/glassfish/bin/appclient -targetserver 10.X.Y.Z - client  
dist/clientjms/P1-jms-clientjms.jar -browse Indique la salida del comando e inclúyala en la  
memoria de prácticas.
```

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

- Realice un pago con la aplicación web
- Obtenga evidencias de que se ha realizado
- Cancelelo con el cliente
- Obtenga evidencias de que se ha cancelado y de que el saldo se ha rectificado

Al realizar este ejercicio en los laboratorios surge un error indicando que no es posible resolver el nombre del host local a una dirección IP. Esto se debe a que no hay una entrada con dicho nombre en el fichero /etc/hosts asociado a una dirección IP. Como dicho fichero no se puede editar, la solución es ejecutar el cliente de colas de mensajes desde la máquina virtual 1, para que se conecte a la máquina virtual 2.

Basta con copiar el .jar del cliente a la máquina virtual, iniciar sesión de forma remota. Indicar donde se encuentra la versión 8 de java exportando la variable JAVA_HOME y ejecutar el cliente de colas con appclient desde la máquina virtual 1.

Para ello, hay que ejecutar la siguiente secuencia de comandos: Desde PC1 host: \$ scp dist/clientjms/P1-jms-clientjms.jar si2@10.X.Y.1:/tmp Desde la máquina virtual 10.X.Y.1: si2@si2srv01:~\$ export JAVA_HOME=/usr/lib/jvm/java-8-oracle/ si2@si2srv01:~\$ /opt/glassfish4/glassfish/bin/appcli

Realización del pago

Pago con tarjeta

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

idTransaccion: 10
idComercio: 10
importe: 10.0
codRespuesta: 000
idAutorizacion: 2

[Volver al comercio](#)

Salida del comando /usr/local/glassfish-4.1.1/glassfish/bin/appclient -targetserver 10.X.Y.Z - client dist/clientjms/P1-jms-clientjms.jar -browse

```
mar 12, 2018 8:07:30 PM org.hibernate.validator.internal.util.Version <clinit>  
INFO: HV000001: Hibernate Validator 5.1.2.Final  
mar 12, 2018 8:07:30 PM com.sun.messaging.jms.ra.ResourceAdapter start  
INFORMACIÓN: MQJMSRA_RA1101: GlassFish MQ JMS Resource Adapter: Version: 5.1.1  
(Build 2-c) Compile: March 17 2015 1045  
mar 12, 2018 8:07:30 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 12, 2018 8:07:30 PM com.sun.messaging.jms.ra.ResourceAdapter start  
INFORMACIÓN: MQJMSRA_RA1101: GlassFish MQ JMS Resource Adapter Started:REMOTE  
Cola de mensajes vacía!
```

Vemos que el pago se ha realizado correctamente

Pago con tarjeta

Pago realizado con éxito. A continuación

idTransaccion: 10
idComercio: 10
importe: 10.0
codRespuesta: 000
idAutorizacion: 2

[Volver al comercio](#)

Prácticas de Sistemas Informáticos II

The screenshot shows a web browser window with the URL 10.1.1.2:8080/P1-ejb-cliente/procesapago. The page displays the payment confirmation details. Overlaid on the browser is a SQL editor window titled 'Tora 2.1.3' with a connection to 'alumnodb@visa.10.1.1.1:5432 [8.4.10]'. The SQL editor shows a query: 'select * from pago , tarjeta where pago.numerotarjeta=tarjeta.numerotarjeta'. The results are displayed in a table with columns: #, idautorizacion, idtransaccion, codrespuesta, importe, idcomercio, and numerotarjeta. The results show two rows of data.

#	idautorizacion	idtransaccion	codrespuesta	importe	idcomercio	numerotarjeta
2	2	10	000	10	10	2347 4840 5058 7931
1	1	1	999	12	1	2347 4840 5058 7931

Envío del mensaje de cancelación:

```

si2@si2srv01:~$ /opt/glassfish4/glassfish/bin/appclient -targetserver 10.1.1.2 -
client /tmp/P1-jms-clientjms.jar 2
Mar 12, 2018 11:37:22 AM org.hibernate.validator.internal.util.Version <clinit>
INFO: HV000001: Hibernate Validator 5.0.0.Final
Mar 12, 2018 11:37:22 AM com.sun.messaging.jms.ra.ResourceAdapter start
INFO: MQJMSRA_RA1101: GlassFish MQ JMS Resource Adapter: Version: 5.0 (Build 1
4-e) Compile: April 12 2013 0104
Mar 12, 2018 11:37:22 AM com.sun.messaging.jms.ra.ResourceAdapter start
INFO: MQJMSRA_RA1101: GlassFish MQ JMS Resource Adapter starting: broker is REMO
TE, connection mode is TCP
Mar 12, 2018 11:37:23 AM com.sun.messaging.jms.ra.ResourceAdapter start
INFO: MQJMSRA_RA1101: GlassFish MQ JMS Resource Adapter Started:REMOTE
Comenzando la produccion
Enviando el siguiente mensaje :2

```

1.3

Edit View Tools Editor Window Help

alumnodb@visa.10.1.1.1:5432 [8.4.10] SQL Editor - *Untitled

```

1 select idautorizacion, idtransaccion, codrespuesta, saldo
2 from pago , tarjeta
3 where pago.numerotarjeta=tarjeta.numerotarjeta

```

Result Execution plan Visualize Logging

#	idautorizacion	idtransaccion	codrespuesta	saldo
1	1	1	999	1000
2	2	10	999	1000