		<b>Escuela Politécnica Superior</b> <b>Ingeniería Informática</b> <b>Prácticas de Sistemas Informáticos 2</b>			
<b>Grupo</b>	<b>M311</b>	<b>Práctica</b>	1A	<b>Fecha</b>	27/02/2017
<b>Alumno/a</b>		Garcia de Lara Parreño, Oscar			
<b>Alumno/a</b>		Glont, Costinel Alexandru			

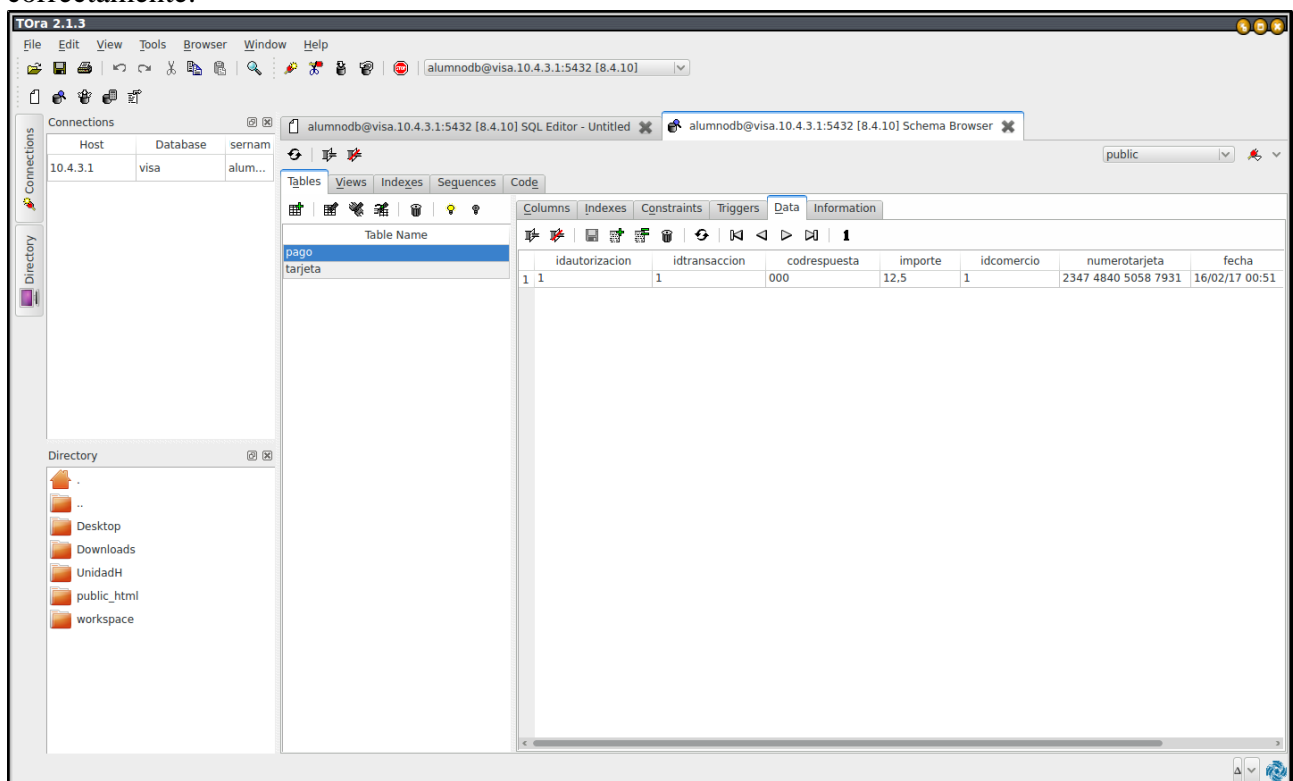
## Práctica 1A: Título

### Ejercicio número 1:

Prepare e inicie una máquina virtual a partir de la plantilla si2srv con: 1GB de RAM asignada, 2 CPUs. A continuación:

- Modifique los ficheros que considere necesarios en el proyecto para que se despliegue tanto la aplicación web como la base de datos contra la dirección asignada a la pareja de prácticas.
- Realice un pago contra la aplicación web empleando el navegador en la ruta <http://10.X.Y.Z:8080/P1> Conéctese a la base de datos (usando el cliente Tora por ejemplo) y obtenga evidencias de que el pago se ha realizado.
- Acceda a la página de pruebas extendida, <http://10.X.Y.Z:8080/P1/testbd.jsp>. Compruebe que la funcionalidad de listado de y borrado de pagos funciona correctamente. Elimine el pago anterior.

Lo primero que hemos hecho ha sido desplegar la maquina de la P0, despues hemos modificado los archivos Build.properties y postgresql.properties poniendo la ip de nuestra maquina virtual 10.4.3.1 Accedemos a <http://10.4.3.1:8080/P1> e introducimos un id para la transicion, un id para el comercio y un valor del pago. Despues accedemos a Tora para ver en la base de datos un valor de tarjeta de credito valido. Y comprobamos nuevamente con Tora que el pago se ha quedado registrado correctamente.



Despues accedemos <http://10.4.3.1:8080/P1/testbd.jsp> y registramos varios pagos para hacer mejor

comprobacion del listamos poniendo el id 2, borramos los del comercio antes referido y nuevamente lo listamos para comprobar que se han borrado correctamente.

Sistema de Pago con tarjeta - Mozilla Firefox

Curso: SISTEMAS IN...Microsoft Word - si2-pra...Microsoft PowerPoint - si...info.txtMail :: InboxSistema de Pago con ta...Edit Application

10.4.3.1:8080/P1/getpagosSearch

## Pago con tarjeta

Lista de pagos del comercio 2

idTransaccion	Importe	codRespuesta	idAutorizacion
1	12.699999809265137	000	2
2	37.0	000	4

[Volver al comercio](#)

Prácticas de Sistemas Informáticos II

Sistema de Pago con tarjeta - Mozilla Firefox

Curso: SISTEMAS IN...Microsoft Word - si2-pra...Microsoft PowerPoint - si...info.txtMail :: InboxSistema de Pago con ta...Edit Application

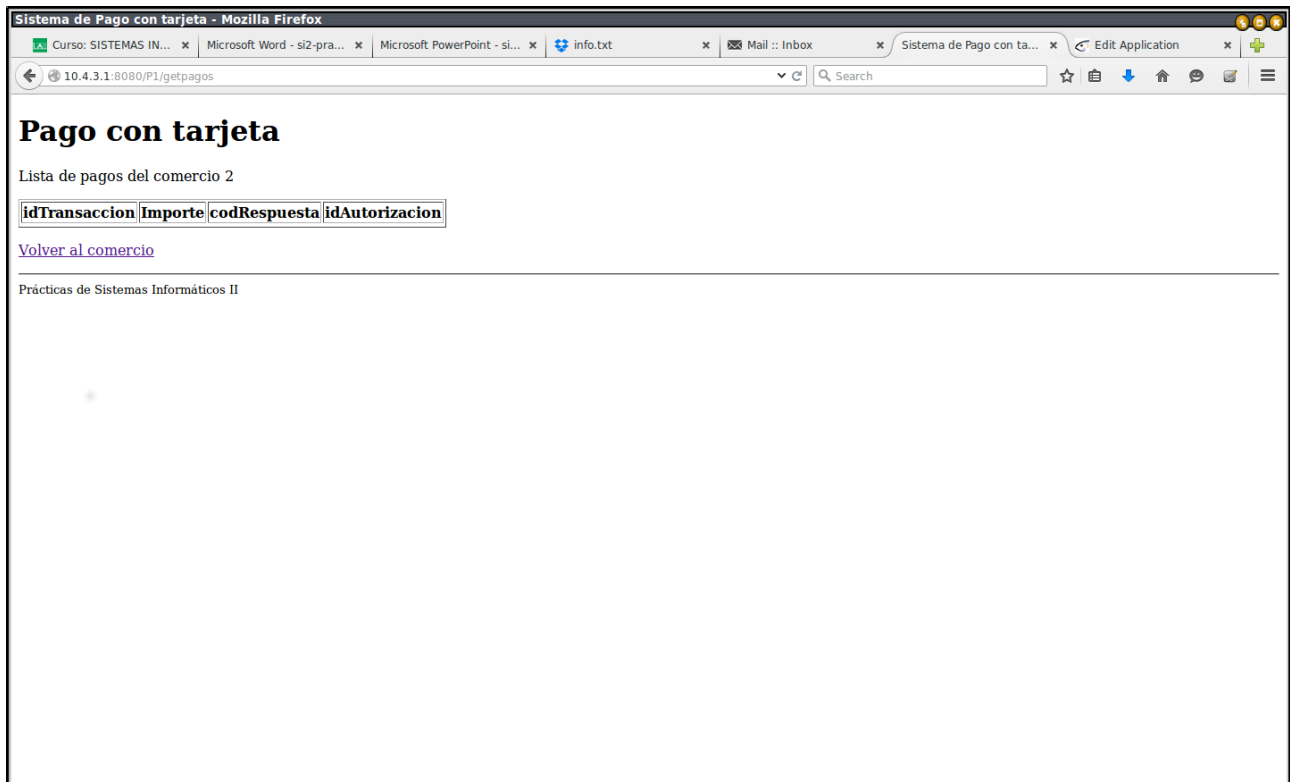
10.4.3.1:8080/P1/delpagosSearch

## Pago con tarjeta

Se han borrado 2 pagos correctamente para el comercio 2

[Volver al comercio](#)

Prácticas de Sistemas Informáticos II



### ***Ejercicio número 2:***

La clase VisaDAO implementa los dos tipos de conexión descritos anteriormente. Sin embargo, la configuración de la conexión utilizando la conexión directa es incorrecta. Se pide completar la información necesaria en dicha clase para llevar a cabo la conexión directa de forma correcta. Para ello habrá que fijar los atributos de dicha clase a los valores correctos. En particular, el nombre del driver JDBC a utilizar, el JDBC connection string que se debe corresponder con el servidor postgresql, y el nombre de usuario y la contraseña. Es necesario consultar el apéndice 10 para ver los detalles de cómo se obtiene una conexión de forma correcta. Una vez completada la información, acceda a la página de pruebas extendida, <http://10.X.Y.Z:8080/P1/testbd.jsp> y pruebe a realizar un pago utilizando la conexión directa y pruebe a listarlo y eliminarlo. Adjunte en la memoria evidencias de este proceso.

Para poder realizarlo hemos modificado el archivo DBTester.java, y hemos cambiado las variables globales

- ⑩ JDBC\_DRIVER = "org.postgresql.Driver";
- ⑩ JDBC\_CONNSTRING="jdbc:postgresql://10.4.3.1:5432/visa";
- ⑩ JDBC\_USER="alumnodb";
- ⑩ JDBC\_PASSWORD="";

Sistema de Pago con tarjeta - Mozilla Firefox

Curso: SISTEMAS IN...Microsoft Word - si2-pra...Microsoft PowerPoint - si...Inicio - DropboxMail :: InboxSistema de Pago con ta...General Information

10.4.3.1:8080/P1/testbd.jspSearch

# Pago con tarjeta

## Proceso de un pago

Id Transacción:

2

Id Comercio:

1

Importe:

55

Numero de visa:

0694 4853 5696 2092

Titular:

Restituta Torres Coll

Fecha Emisión:

08/09

Fecha Caducidad:

03/20

CVV2:

547

Modo debug:

☒ True ☐ False

Direct Connection:

☒ True ☐ False

Use Prepared:

☒ True ☐ False

Pagar

## Consulta de pagos

Id Comercio:

GetPagos

## Borrado de pagos

Id Comercio:

DelPagos

Mozilla Firefox seems slow... to... start.

Learn How to Speed It UpDon't Tell Me Again

Sistema de Pago con tarjeta - Mozilla Firefox

Curso: SISTEMAS IN...Microsoft Word - si2-pra...Microsoft PowerPoint - si...Inicio - DropboxMail :: InboxSistema de Pago con ta...General Information

10.4.3.1:8080/P1/procesapagoSearch

# Pago con tarjeta

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

idTransaccion: 2

idComercio: 1

importe: 55.0

codRespuesta: 000

idAutorizacion: 2

[Volver al comercio](#)

Prácticas de Sistemas Informáticos II

Mozilla Firefox seems slow... to... start.

Learn How to Speed It UpDon't Tell Me Again

Sistema de Pago con tarjeta - Mozilla Firefox

Curso: SISTEMAS IN... Microsoft Word - si2-pra... Microsoft PowerPoint - si... Inicio - Dropbox Mail :: Inbox Sistema de Pago con ta... General Information

10.4.3.1:8080/P1/getpagos Search

## Pago con tarjeta

Lista de pagos del comercio 1

idTransaccion	Importe	codRespuesta	idAutorizacion
1	12.0	000	1
2	55.0	000	2

[Volver al comercio](#)

---

Prácticas de Sistemas Informáticos II

Mozilla Firefox seems slow... to... start. Learn How to Speed It Up Don't Tell Me Again

Sistema de Pago con tarjeta - Mozilla Firefox

Curso: SISTEMAS IN... Microsoft Word - si2-pra... Microsoft PowerPoint - si... Inicio - Dropbox Mail :: Inbox Sistema de Pago con ta... General Information

10.4.3.1:8080/P1/delpagos Search

## Pago con tarjeta

Se han borrado 2 pagos correctamente para el comercio 1

[Volver al comercio](#)

---

Prácticas de Sistemas Informáticos II

Mozilla Firefox seems slow... to... start. Learn How to Speed It Up Don't Tell Me Again

### *Ejercicio número 3:*

**Examinar el archivo `postgresql.properties` para determinar el nombre del recurso JDBC correspondiente al DataSource y el nombre del pool. Acceda a la Consola de Administración. Compruebe que los recursos JDBC y pool de conexiones han sido correctamente creados.**

**Realice un Ping JDBC a la base de datos. Anote en la memoria de la práctica los valores para los parámetros Initial and Minimum Pool Size, Maximum Pool Size, Pool Resize Quantity, Idle Timeout, Max Wait Time. Comente razonadamente qué impacto considera que pueden tener estos parámetros en el rendimiento de la aplicación.**

db.pool.name=VisaPool

db.jdbc.resource.name=jdbc/VisaDB

Para hacer el ping entramos en JDBC/JDBC connection Pools/VisaPool y damos al boton ping

- ⑩ **Initial and Minimum Pool Size:** 8 Conexiones
- ⑩ **Maximum Pool Size:** 32 Conexiones
- ⑩ **Pool Resize Quantity:** 2 Conexiones
- ⑩ **Idle Timeout:** 300 s
- ⑩ **Max Wait Time:** 60000 ms

El impacto es porque la operación de crear una conexión a la base de datos es muy costosa, por tanto tener un numero inicial muy bajo cuando se van tener muchas mas conexiones seria ineficiente, incluso tener un numero mínimo muy alto que nunca se alcanza también seria muy ineficiente al principio de la aplicación. El numero máximo si es demasiado alto, cargaría de conexiones simultaneas el servidor y este no podría atenderlas con un rendimiento aceptable, si es muy bajo se crearía una cola de espera que también bajaría el rendimiento general. El timeout evita que alguien monopolice la conexión un numero muy bajo estaríamos conectando y desconectando muy periódicamente. El max wait muy alto soloparía con el timeout y seria poco útil, muy bajo podría cortar consultas de conexiones mas lentas.

**JDBC Resources**

JDBC resources provide applications with a means to connect to a database.

Resources (3)

Select	JNDI Name	Logical JNDI Name	Enabled	Connection Pool	Description
<input type="checkbox"/>	jdbc/VisaDB		✓	VisaPool	
<input type="checkbox"/>	jdbc/_TimerPool		✓	TimerPool	
<input type="checkbox"/>	jdbc/_default	java:comp/DefaultDataSource	✓	DerbyPool	

JDBC Connection Pools - Mozilla Firefox

Curso: SISTEMAS IN...Microsoft Word - si2-pra...Microsoft PowerPoint - si...Inicio - DropboxMail :: InboxSistema de Pago con ta...JDBC Connection Pools

https://10.4.3.1:4848/common/index.jsf

Search

HomeAbout...LogoutHelp

User: admin | Role: domain1 | Server: 10.4.3.1

GlassFish™ Server Open Source Edition

Total # of available updates : 48

Tree

Common Tasks

Domain

server (Admin Server)

Clusters

Standalone Instances

Nodes

Applications

P1

Lifecycle Modules

Monitoring Data

Resources

Concurrent Resources

Connectors

JDBC

JDBC Resources

JDBC Connection Pools

JMS Resources

JNDI

JavaMail Sessions

Resource Adapter Configs

Configurations

default-config

server-config

Update Tool

JDBC Connection Pools

To store, organize, and retrieve data, most applications use relational databases. Java EE applications access relational databases through the JDBC API. Before an application can access a database, it must get a connection.

Pools (3)

New...Delete

Select	Pool Name	Resource Type	Classname	Description
<input type="checkbox"/>	DerbyPool	javax.sql.DataSource	org.apache.derby.jdbc.ClientDataSource	
<input type="checkbox"/>	VisaPool	javax.sql.ConnectionPoolDataSource	org.postgresql.ds.PGConnectionPoolDataSource	
<input type="checkbox"/>	TimerPool	javax.sql.XADataSource	org.apache.derby.jdbc.EmbeddedXADataSource	

Mozilla Firefox seems slow... to... start.

Learn How to Speed It UpDon't Tell Me Again

Edit JDBC Connection Pool - Mozilla Firefox

Curso: SISTEMAS IN...Microsoft Word - si2-pra...Microsoft PowerPoint - si...Inicio - DropboxMail :: InboxSistema de Pago con ta...Edit JDBC Connecio...

https://10.4.3.1:4848/common/index.jsf

Search

HomeAbout...LogoutHelp

User: admin | Role: domain1 | Server: 10.4.3.1

GlassFish™ Server Open Source Edition

Total # of available updates : 48

Tree

Common Tasks

Domain

server (Admin Server)

Clusters

Standalone Instances

Nodes

Applications

P1

Lifecycle Modules

Monitoring Data

Resources

Concurrent Resources

Connectors

JDBC

JDBC Resources

jdbc/VisaDB

jdbc/TimerPool

jdbc/\_default

JDBC Connection Pools

DerbyPool

VisaPool

TimerPool

JMS Resources

JNDI

JavaMail Sessions

Resource Adapter Configs

Configurations

GeneralAdvancedAdditional Properties

Ping Succeeded

Edit JDBC Connection Pool

SaveCancel

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

Load DefaultsFlushPing

\* Indicates required field

General Settings

Pool Name: VisaPool

Resource Type: javax.sql.ConnectionPoolDataSource

Must be specified if the datasource class implements more than 1 of the interface.

Datasource Classname: org.postgresql.ds.PGConnectionPoolDataSource

Vendor-specific classname that implements the DataSource and/or XADataSource APIs

Driver Classname:

Vendor-specific classname that implements the java.sql.Driver interface.

Ping: ☐ Enabled

When enabled, the pool is pinged during creation or reconfiguration to identify and warn of any erroneous values for its attributes

Deployment Order: 100

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

Description:

Pool Settings

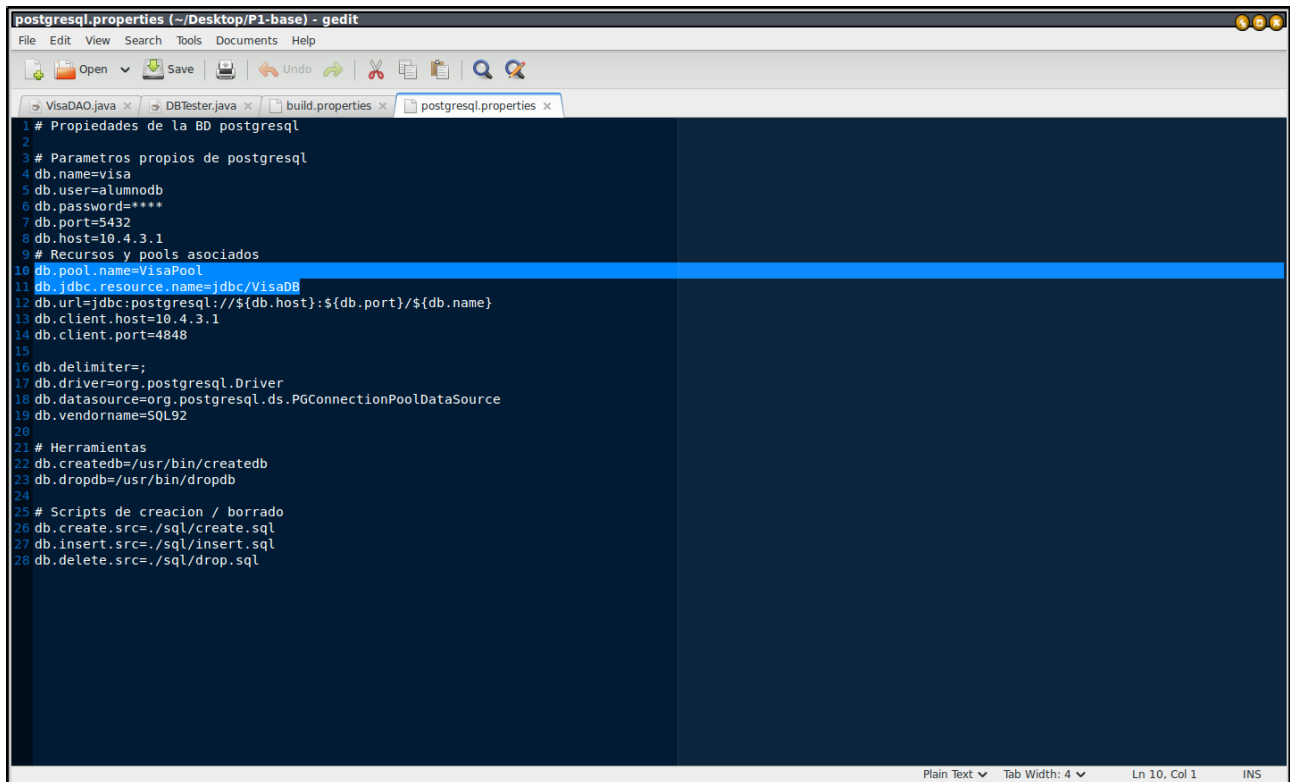
Initial and Minimum Pool Size: 8 Connections

Minimum and initial number of connections maintained in the pool

Maximum Pool Size: 32 Connections

Mozilla Firefox seems slow... to... start.

Learn How to Speed It UpDon't Tell Me Again



```
1 # Propiedades de la BD postgresql
2
3 # Parametros propios de postgresql
4 db.name=visa
5 db.user=alumnodb
6 db.password=****
7 db.port=5432
8 db.host=10.4.3.1
9 # Recursos y pools asociados
10 db.pool.name=VisaPool
11 db.jdbc.resource.name=jdbc/VisaDB
12 db.url=jdbc:postgresql://${db.host}:${db.port}/${db.name}
13 db.client.host=10.4.3.1
14 db.client.port=4848
15
16 db.delimiter=;
17 db.driver=org.postgresql.Driver
18 db.datasource=org.postgresql.ds.PGConnectionPoolDataSource
19 db.vendorname=SQL92
20
21 # Herramientas
22 db.createdb=usr/bin/createdb
23 db.dropdb=usr/bin/dropdb
24
25 # Scripts de creacion / borrado
26 db.create.src=./sql/create.sql
27 db.insert.src=./sql/insert.sql
28 db.delete.src=./sql/drop.sql
```

#### ***Ejercicio número 4:***

**Localice los siguientes fragmentos de código SQL dentro del proyecto proporcionado (P1-base) correspondientes a los siguientes procedimientos:**

##### **- Consulta de si una tarjeta es válida.**

```
String qry = "select * from tarjeta "
            + "where numeroTarjeta=" + tarjeta.getNumero()
            + " and titular=" + tarjeta.getTitular()
            + " and validaDesde=" + tarjeta.getFechaEmision()
            + " and validaHasta=" + tarjeta.getFechaCaducidad()
            + " and codigoVerificacion=" + tarjeta.getCodigoVerificacion() + "";
```

```
SELECT_TARJETA_QRY =
    "select * from tarjeta " +
    "where numeroTarjeta=? " +
    " and titular=? " +
    " and validaDesde=? " +
    " and validaHasta=? " +
    " and codigoVerificacion=? ";
```

##### **- Ejecución del pago.**

```
String qry = "insert into pago("
            + "idTransaccion,"
            + "importe,idComercio,"
            + "numeroTarjeta)"
            + " values ("
```



```
+ "" + pago.getIdTransaccion() + ","
+ pago.getImporte() + ","
+ "" + pago.getIdComercio() + ","
+ "" + pago.getTarjeta().getNumero() + ""
+ ")";
```

```
String INSERT_PAGOS_QRY =
    "insert into pago(" +
    "idTransaccion,importe,idComercio,numeroTarjeta)" +
    " values (?, ?, ?, ?)";
```

### ***Ejercicio número 5:***

**Edite el fichero VisaDAO.java y localice el método errorLog. Compruebe en qué partes del código se escribe en log utilizando dicho método. Realice un pago utilizando la página testbd.jsp con la opción de debug activada. Visualice el log del servidor de aplicaciones y compruebe que dicho log contiene información adicional sobre las acciones llevadas a cabo en VisaDAO.java.**

Se usa en los catch para informar de las excepciones y también se usa para informar, aunque llamándose error suena un poco raro, las query que se van a lanzar, o sea por donde está ejecutando la aplicación.

### ***Ejercicio número 6:***

**¿Por qué se ha de alterar el parámetro de retorno del método realizaPago() para que devuelva el pago el lugar de un boolean?**

Necesitas cambiar de boolean a PagoBean porque ahora necesitas devolver el objeto pago que contiene la información del id y null en caso de no ser posible

### ***Ejercicio número 7:***

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

Es un fichero xml wsdl, que está en aplicación P1-ws-ws

**¿Qué tipos de datos predefinidos se usan?**

Xs:string, xs:boolean, xs:int y xs:Double

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

Los definidos en xsd:schema en el otro fichero te define los métodos, los atributos de clases como PagoBean y los argumentos de los métodos

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

Es la etiqueta Message

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

Es la etiqueta PortType

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

Es la etiqueta Binding

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

En la etiqueta Service dentro contiene la etiqueta soap:address

### ***Ejercicio número 9:***

**Modifique la llamada al servicio para que la ruta al servicio remoto se obtenga del fichero de**

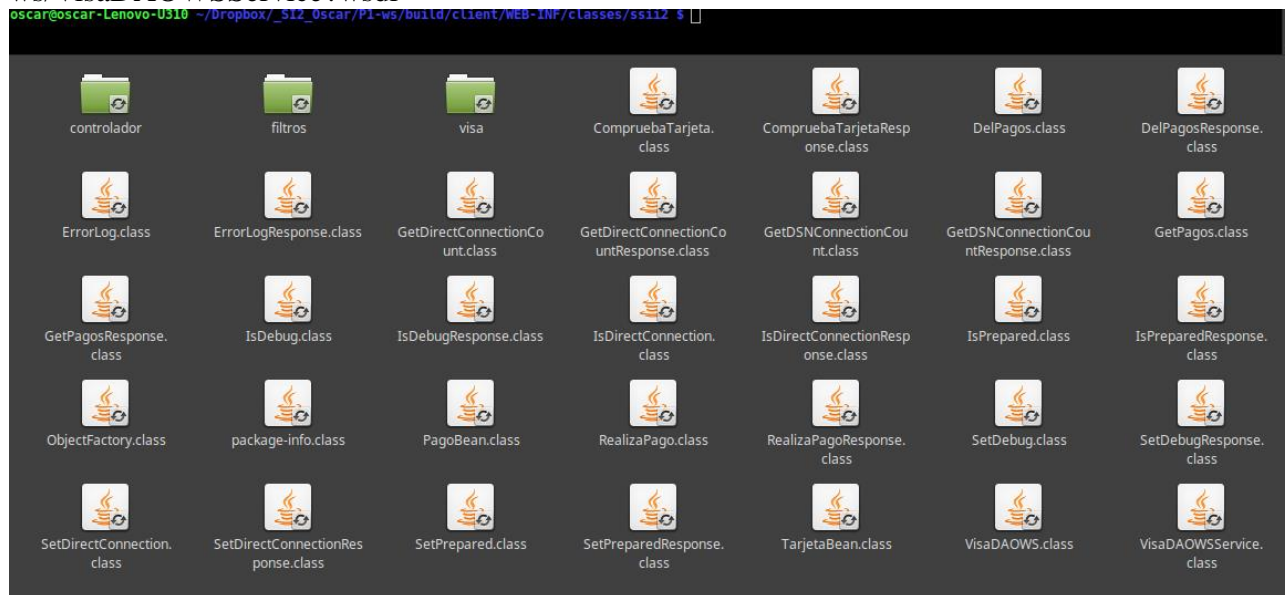
configuración web.xml. Para saber cómo hacerlo consulte el apéndice 15.1 para más información y edite el fichero web.xml y analice los comentarios que allí se incluyen.

Hemos descomentado la etiqueta <context-param> cambiando el param-value con la dirección de nuestro servidor.

### **Ejercicio número 11:**

Realice una importación manual del WSDL del servicio sobre el directorio de clases local. Anote en la memoria qué comando ha sido necesario ejecutar en la línea de comandos, qué clases han sido generadas y por qué. Téngase en cuenta que el servicio debe estar previamente desplegado.

```
wsimport -d build/cliente/WEB-INF/classes -p ssii2.visa http://10.4.3.1:8080/P1-ws-  
ws/VisaDAOWSService?wsdl
```



En la ruta build/cliente/WEB-INF/classes se han generado muchos archivos, esto sirve para que el cliente conozca las clases y metodos disponibles para poder comunicarse con el servidor.

### **Ejercicio número 12:**

Complete el target generar-stubs definido en build.xml para que invoque a wsimport (utilizar la funcionalidad de ant exec para ejecutar aplicaciones). Téngase en cuenta que:

- El raíz del directorio de salida del compilador para la parte cliente ya está definido en build.properties como \${build.cliente}/WEB-INF/classes
- El paquete Java raíz (ssii2) ya está definido como \${paquete}
- La URL ya está definida como \${wsdl.url}

```
<!-- TODO - Implementar llamada wsimport -->  
<exec executable="wsimport">  
  <arg line=" -d ${build.cliente}/WEB-INF/classes" />  
  <arg line=" -p ${paquete}" />  
  <arg line=" ${wsdl.url}" />  
</exec>
```

Hemos añadido esta línea siguiendo el ejemplo de otro exec del archivo.

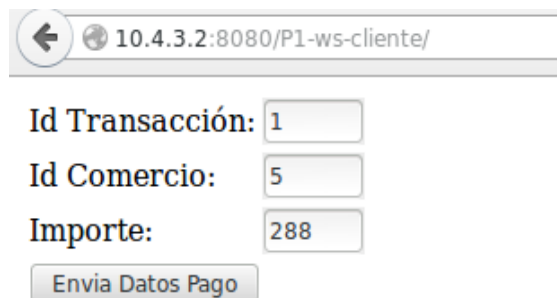
### **Ejercicio número 13:**

Realice un despliegue de la aplicación completo en dos nodos tal y como se explica en la

**Figura 8.** Habrá que tener en cuenta que ahora en el fichero build.properties hay que especificar la dirección IP del servidor de aplicaciones donde se desplegará la parte del cliente de la aplicación y la dirección IP del servidor de aplicaciones donde se desplegará la parte del servidor. Las variables as.host.client y as.host.server deberán contener esta información.

- Probar a realizar pagos correctos a través de la página testbd.jsp. Ejecutar las consultas SQL necesarias para comprobar que se realiza el pago. Anotar en la memoria práctica los resultados en forma de consulta SQL y resultados sobre la tabla de pagos.

Hemos realizado un pago desde el cliente que esta desplegado en un PC con la maquina virtual con dirección 10.4.3.2 y el server en otro pc con la maquina virtual con la dirección 10.4.3.1



← 10.4.3.2:8080/P1-ws-cliente/

Id Transacción: 1

Id Comercio: 5

Importe: 288

Envia Datos Pago



← 10.4.3.2:8080/P1-ws-cliente/comienzapago

## Pago con tarjeta

Numero de visa: 8365 5667 6698 6481

Titular: Enjuto Gonzalez Torres

Fecha Emisión: 03/09

Fecha Caducidad: 11/20

CVV2: 421

Pagar

---

Id Transacción: 1

Id Comercion: 5

Importe: 288.0

---

# Pago con tarjeta

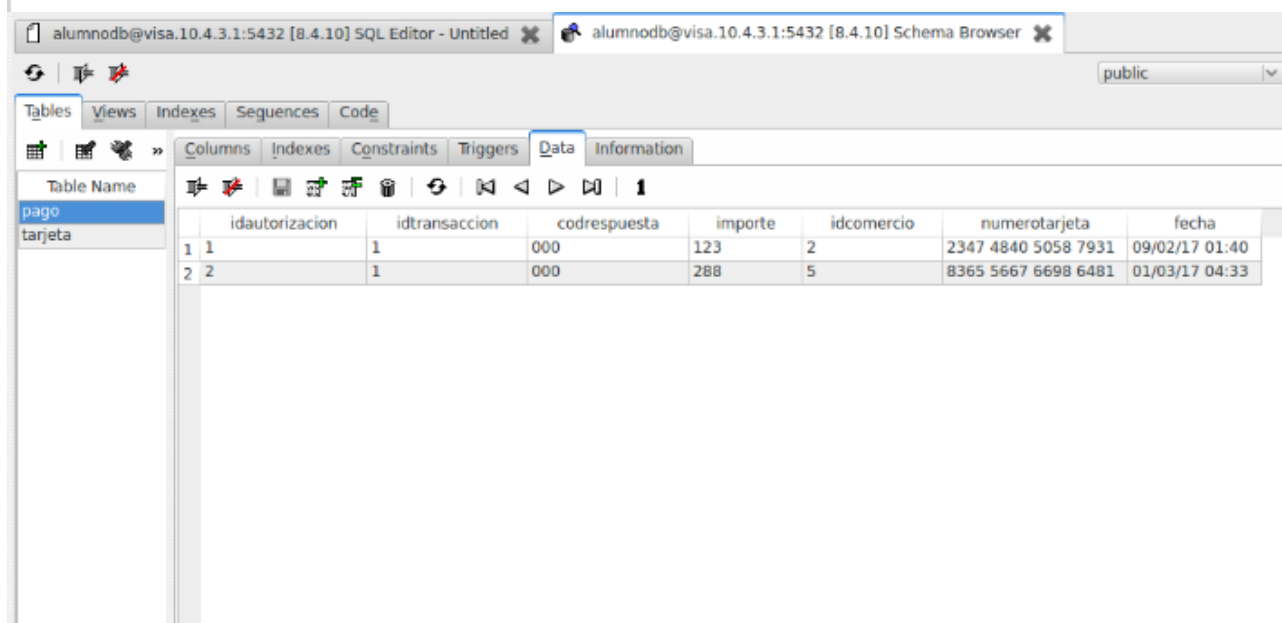
Pago realizado con éxito. A continuación se muestra el comprobante del mismo:

idTransaccion: 1  
idComercio: 5  
importe: 288.0  
codRespuesta:  
idAutorizacion:

[Volver al comercio](#)

---

Prácticas de Sistemas Informáticos II



alumnodb@visa.10.4.3.1:5432 [8.4.10] SQL Editor - Untitled

alumnodb@visa.10.4.3.1:5432 [8.4.10] Schema Browser

public

Table Name	idautorizacion	idtransaccion	codrespuesta	importe	idcomercio	numerotarjeta	fecha
pago	1	1	000	123	2	2347 4840 5058 7931	09/02/17 01:40
tarjeta	2	1	000	288	5	8365 5667 6698 6481	01/03/17 04:33

### ***Cuestión número 1:***

**Teniendo en cuenta el diagrama de la Figura 3, indicar las páginas html, jsp y servlets por los que se pasa para realizar un pago desde pago.html, pero en el caso de uso en que se introduce una tarjeta cuya fecha de caducidad ha expirado.**

Pago.html luego vas al servlet de Comienza Pago, después vas a formdatosvisa.jsp, de él vas al servlet de Procesa Pago y ahí detecta el error en el campo de la tarjeta, este te retorna a formdatosvisa.jsp, luego vas a error/muestraerror.jsp que muestra lo de pago incorrecto.

### ***Cuestión número 2:***

**De los diferentes servlets que se usan en la aplicación, ¿podría indicar cuáles son los encargados de solicitar la información sobre el pago con tarjeta cuando se usa pago.html para realizar el pago?**

Es el servlet “Comienza Pago”.

### ***Cuestión número 3:***

**Cuando se accede a pago.html para hacer el pago, ¿qué información solicita cada servlet?**

**Respecto a la información que manejan, ¿cómo la comparten? ¿dónde se almacena?**

En “Comienza Pago” es el que solicita los campos de el numero de la tarjeta, el titular, fecha de emision y caducidad y el CVV (código de verificación). Se almacena en la instancia de Pago Bean y se usa en formdatosvisa.jsp y en “Procesa Pago”.

### ***Cuestión número 4:***

**Enumere las diferencias que existen en la invocación de servlets, a la hora de realizar el pago, cuando se utiliza la página de pruebas extendida testbd.jsp frente a cuando se usa pago.html. ¿Podría indicar por qué funciona correctamente el pago cuando se usa testbd.jsp a pesar de las diferencias observadas?**

Las diferencias es que cuando se accede por pago.html los datos se piden en dos tandas, usando “Comienza Pago” para pedir los datos de la tarjeta de credito, en cambio testbd.jsp pide los datos a la vez. Ambos, uno desde “Comienza Pago” y otro desde testbd.jsp, envian los datos a “Procesa Pago” por eso funciona igualmente.