		Escuela Politécnica Superior Ingeniería Informática Práctica de Sistemas Informáticos 2			
Grupo	2323	Práctica	3	Fecha	6/03/2023
Alumno/a		Hidalgo, Gamborino, Sergio			
Alumno/a		Ibáñez, González, Miguel			

Práctica 3: Seguridad y disponibilidad

Ejercicio 1

Preparar 3 máquinas virtuales desde cero (a partir de la VM en moodle) con acceso SSH entre ellas. Esta tarea es necesaria para la correcta gestión del cluster que definiremos en el próximo apartado. Las VMs las denominaremos:

si2srv01: Dirección IP 10.X.Y.1, 768MB RAM
 si2srv02: Dirección IP 10.X.Y.2, 512MB RAM
 si2srv03: Dirección IP 10.X.Y.3, 512MB RAM

RECUERDE RANDOMIZAR LAS DIRECCIONES MAC DE CADA COPIA ANTES DE INTENTAR USAR EL NODO.

En la primera máquina (10.X.Y.1), generaremos el par de claves con DSA. A continuación importaremos la clave pública en cada uno de los otros dos nodos (10.X.Y.2 y 10.X.Y.3). Probaremos a acceder por SSH desde .1 a .2 y .3, comprobando que no requiere la introducción de la clave. Obtener una evidencia del inicio remoto de sesión mediante la salida detallada (`ssh -v si2@10.X.Y.2` y `ssh -v si2@10.X.Y.3`). Anote dicha salida en la memoria de prácticas.

Revisar y comentar la salida del mandato `ssh`. Una vez realizado este punto, detendremos las tres máquinas virtuales y obtendremos una copia de las mismas a algún medio externo (USB) para los consiguientes apartados de esta práctica. También es recomendable que preserve los directorios `.ssh` de cada uno de los nodos

Conexión con si2@10.6.4.2

```
File Virtual Machine Help
debug1: ssh_rsa_verify: signature correct
debug1: SSH2_MSG_NEWKEYS sent
debug1: expecting SSH2_MSG_NEWKEYS
debug1: SSH2_MSG_NEWKEYS received
debug1: SSH2_MSG_SERVICE_REQUEST sent
debug1: SSH2_MSG_SERVICE_ACCEPT received
debug1: Authentications that can continue: publickey,password
debug1: Next authentication method: publickey
debug1: Trying private key: /home/si2/.ssh/identity
debug1: Offering public key: /home/si2/.ssh/id_rsa
debug1: Server accepts key: pkalg ssh-rsa blen 277
debug1: read PEM private key done: type RSA
debug1: Authentication succeeded (publickey).
debug1: channel 0: new [client-session]
debug1: Requesting no-more-sessions@openssh.com
debug1: Entering interactive session.
debug1: Sending environment.
debug1: Sending env LANG = C
Linux si2srv02 2.6.32-33-generic #72-Ubuntu SMP Fri Jul 29 21:08:3
6 GNU/Linux
Ubuntu 10.04.3 LTS

Welcome to Ubuntu!
 * Documentation:  https://help.ubuntu.com/
New release 'precise' available.
Run 'do-release-upgrade' to upgrade to it.

Last login: Wed May  3 13:43:46 2023
Loading es
si2@si2srv02:~$
```

Conexión con si2@10.6.4.3

```
net8: debug1: ssh_rsa_verify: signature correct
debug1: SSH2_MSG_NEWKEYS sent
debug1: expecting SSH2_MSG_NEWKEYS
debug1: SSH2_MSG_NEWKEYS received
debug1: SSH2_MSG_SERVICE_REQUEST sent
debug1: SSH2_MSG_SERVICE_ACCEPT received
debug1: Authentications that can continue: publickey,password
debug1: Next authentication method: publickey
debug1: Trying private key: /home/si2/.ssh/identity
debug1: Offering public key: /home/si2/.ssh/id_rsa
debug1: Server accepts key: pkalg ssh-rsa blen 277
debug1: read PEM private key done: type RSA
debug1: Authentication succeeded (publickey).
debug1: channel 0: new [client-session]
debug1: Requesting no-more-sessions@openssh.com
debug1: Entering interactive session.
debug1: Sending environment.
debug1: Sending env LANG = C
Linux si2srv03 2.6.32-33-generic #72-Ubuntu SMP Fri Jul 29
6 GNU/Linux
Ubuntu 10.04.3 LTS

Welcome to Ubuntu!
 * Documentation:  https://help.ubuntu.com/
New release 'precise' available.
Run 'do-release-upgrade' to upgrade to it.

Last login: Wed May  3 14:23:29 2023
Loading es
si2@si2srv03:~$
```

Como se puede ver en ambas capturas, se confirma la autenticidad de la conexión (al ser 10.6.4.1 el único poseedor de la clave correspondiente a la clave pública).

Ejercicio 2:

Realizar los pasos del apartado 4 con el fin de obtener una configuración válida del cluster SI2Cluster, con la topología indicada de 1 DAS y 2 nodos SSH de instancias. Inicie el cluster. Liste las instancias del cluster y verifique que los pids de los procesos Java (JVM) correspondientes2 están

Creemos el Nodo01 en el servidor con ip 10.6.4.2 con el comando “asadmin --user admin --passwordfile /opt/SI2/passwordfile create-node-ssh --sshuser si2 --nodehost 10.6.4.2 --nodedir /opt/glassfish4 Node01”

```
ELITE-V2:~/glassfish4/glassfish/domains/domain1$ ls
seregio@seregio-B550-AORUS-ELITE-V2: ~/Escritorio/cositas...
si2@si2srv01:~$ asadmin --user admin --passwordfile /opt/SI2/passwordfile create-node-ssh --sshuser si2 --nodehost 10.6.4.2 --nodedir /opt/glassfish4 Node01
Command create-node-ssh executed successfully.
si2@si2srv01:~$
```

Repetimos para crear el Nodo02 en el servidor con ip 10.6.4.3 con el comando “asadmin --user admin --passwordfile /opt/SI2/passwordfile create-node-ssh --sshuser si2 --nodehost 10.6.4.3 --nodedir /opt/glassfish4 Node02”

```
seregio@seregio-B550-AORUS-ELITE-V2: ~/Escritorio/cositas...
si2@si2srv01:~$ asadmin --user admin --passwordfile /opt/SI2/passwordfile create-node-ssh --sshuser si2 --nodehost 10.6.4.2 --nodedir /opt/glassfish4 Node01
Command create-node-ssh executed successfully.
si2@si2srv01:~$ asadmin --user admin --passwordfile /opt/SI2/passwordfile create-node-ssh --sshuser si2 --nodehost 10.6.4.3 --nodedir /opt/glassfish4 Node02
Command create-node-ssh executed successfully.
si2@si2srv01:~$
```

Se indica como output del comando que ambos han ido bien, y esto queda confirmado al ejecutar el comando “asadmin --user admin --passwordfile /opt/SI2/passwordfile list-nodes” para listarlos.

```
si2@si2srv01:~$ asadmin --user admin --passwordfile /opt/SI2/passwordfile list-nodes
localhost-domain1 CONFIG localhost
Node01 SSH 10.6.4.2
Node02 SSH 10.6.4.3
Command list-nodes executed successfully.
si2@si2srv01:~$
```

No obstante, se hará también un ping a cada uno para comprobar la conexión ssh.

Al Node01 con el comando: “asadmin --user admin --passwordfile /opt/SI2/passwordfile ping-node-ssh Node01” y al Node02 con el comando: “asadmin --user admin --passwordfile /opt/SI2/passwordfile ping-node-ssh Node02”

```
si2@si2srv01:~$ asadmin --user admin --passwordfile /opt/SI2/passwordfile ping-node-ssh Node01
Successfully made SSH connection to node Node01 (10.6.4.2)
Command ping-node-ssh executed successfully.
si2@si2srv01:~$ asadmin --user admin --passwordfile /opt/SI2/passwordfile ping-node-ssh Node02
Successfully made SSH connection to node Node02 (10.6.4.3)
Command ping-node-ssh executed successfully.
si2@si2srv01:~$
```

En el menú de navegación la opción Nodes, podremos ver los dos nodos creados

Nodes (3)		
<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="button" value="New..."/> <input type="button" value="Delete"/> <input type="button" value="Delete and Uninstall"/>
Select	Name	Node Host
<input type="checkbox"/>	Node01	10.6.4.2
<input type="checkbox"/>	Node02	10.6.4.3
<input type="checkbox"/>	localhost-domain1	localhost

Tras esto, se crea el cluster con el comando “asadmin create-cluster SI2Cluster”

```
seregio@seregio-B550-AORUS-ELITE-V2: ~/Escritorio/cos
seregio@si2srv01:~$ asadmin create-cluster SI2Cluster
Enter admin user name> admin
Enter admin password for user "admin">
Command create-cluster executed successfully.
seregio@si2srv01:~$
```

Y se puede listar con “asadmin list-clusters” (no está iniciado aunque sí creado)

```
si2@si2srv01:~$ asadmin list-clusters
Enter admin user name> admin
Enter admin password for user "admin">
SI2Cluster not running
Command list-clusters executed successfully.
si2@si2srv01:~$
```

Seguido se comprueba que todos los nodos se conocen, mirando el archivo “/etc/hosts”:

En 10.6.4.1:

```
GNU nano 2.2.2
10.6.4.1 si2srv01
10.6.4.2 si2srv02
10.6.4.3 si2srv03
127.0.0.1 localhost
```

En 10.6.4.2:

```
GNU nano 2.2.2
10.6.4.1 si2srv01
10.6.4.2 si2srv02
10.6.4.3 si2srv03
127.0.0.1 localhost
```

En 10.6.4.3:

```
GNU nano 2.2.2
10.6.4.1 si2srv01
10.6.4.2 si2srv02
10.6.4.3 si2srv03
127.0.0.1 localhost
```

Ahora habrá que crear una instancia asociada a cada nodo:

Comando de la instancia 1: “asadmin --user admin --passwordfile /opt/SI2/passwordfile create-instance --cluster SI2Cluster --node Node01 Instance01”

```
si2@si2srv01:~$ asadmin --user admin --passwordfile /opt/SI2/passwordfile create-instance --cluster SI2Cluster --node Node01 Instance01
Command create-instance-filesystem executed successfully.
Port Assignments for server instance Instance01:
OSGI_SHELL_TELNET_PORT=26666
JAVA_DEBUGGER_PORT=29009
JMS_PROVIDER_PORT=27676
HTTP_LISTENER_PORT=28080
IIOP_SSL_LISTENER_PORT=23820
ASADMIN_LISTENER_PORT=24848
IIOP_SSL_MUTUALAUTH_PORT=23920
JMX_SYSTEM_CONNECTOR_PORT=28686
HTTP_SSL_LISTENER_PORT=28181
IIOP_LISTENER_PORT=23700
The instance, Instance01, was created on host 10.6.4.2
Command create-instance executed successfully.
```

Comando de la instancia 2: “asadmin --user admin --passwordfile /opt/SI2/passwordfile create-instance --cluster SI2Cluster --node Node02 Instance02”

```
si2@si2srv01:~$ asadmin --user admin --passwordfile /opt/SI2/passwordfile create
-instance --cluster SI2Cluster --node Node02 Instance02
Command create-instance-filesystem executed successfully.
Port Assignments for server instance Instance02:
OSGI_SHELL_TELNET_PORT=26666
JAVA_DEBUGGER_PORT=29009
JMS_PROVIDER_PORT=27676
HTTP_LISTENER_PORT=28080
IIOP_SSL_LISTENER_PORT=23820
ASADMIN_LISTENER_PORT=24848
IIOP_SSL_MUTUALAUTH_PORT=23920
JMX_SYSTEM_CONNECTOR_PORT=28686
HTTP_SSL_LISTENER_PORT=28181
IIOP_LISTENER_PORT=23700
The instance, Instance02, was created on host 10.6.4.3
Command create-instance executed successfully.
```

Y como se ha realizado para pasos anteriores, se listan para asegurar que han sido creadas, se usa el comando “asadmin --user admin --passwordfile /opt/SI2/passwordfile list-instances -l”

```
Command create-instance executed successfully.
si2@si2srv01:~$ asadmin --user admin --passwordfile /opt/SI2/passwordfile list-i
nstances -l
Name          Host      Port  Pid  Cluster      State
Instance01    10.6.4.2  24848  --   SI2Cluster    not running
Instance02    10.6.4.3  24848  --   SI2Cluster    not running
Command list-instances executed successfully.
```

Por último se inicia el cluster con el comando “asadmin --user admin --passwordfile /opt/SI2/passwordfile start-cluster SI2Cluster”

```
seregio@sere...si2@si2srv01:~$ asadmin --user admin --passwordfile /opt/SI2/passwordfile start
cluster SI2Cluster
seregio@sere...Command start-cluster executed successfully.
```

Y hasta aquí la creación e inicialización del cluster, tras esto, queda configurarlo a través del archivo SI2Cluster-config, que puede ser modificado en la consola de administrador del servidor de glassfish:

Configurations

Manage configurations, and view the target server instances or clusters using the configurations.

Configurations (3)

New...

Delete

Select	Name	Instance
<input type="checkbox"/>	SI2Cluster-config	Instance01 Running Instance02 Running
<input type="checkbox"/>	default-config	
<input type="checkbox"/>	server-config	server Running

Se han realizado los cambios pertinentes en JVM Options:

JVM Options

Manage JVM options for the server. Values containing one or more spaces must be enclosed in double quotes ("value string").

Configuration Name: SI2Cluster-config

Options (29)	
<div><div><div></div><div></div></div><div><div>Add JVM Option</div><div>Delete</div></div></div>	
Select	Value
<input type="checkbox"/>	-Xms128m
<input type="checkbox"/>	-XX:MaxPermSize=96m
<input type="checkbox"/>	-server
<input type="checkbox"/>	-Djava.awt.headless=true
<input type="checkbox"/>	-Djdk.corba.allowOutputStreamSubclass=true
<input type="checkbox"/>	-XX:+UnlockDiagnosticVMOptions
<input type="checkbox"/>	-Djava.endorsed.dirs=\${com.sun.aas.installRoot}/modules/endorsed\${path.separator}\${com.sun.aas.installRoot}/lib/endorsed
<input type="checkbox"/>	-Djava.security.policy=\${com.sun.aas.instanceRoot}/config/server.policy
<input type="checkbox"/>	-Djava.security.auth.login.config=\${com.sun.aas.instanceRoot}/config/login.conf
<input type="checkbox"/>	-Dcom.sun.enterprise.security.httpsOutboundKeyAlias=s1as
<input type="checkbox"/>	-Djavax.net.ssl.keyStore=\${com.sun.aas.instanceRoot}/config/keystore.jks
<input type="checkbox"/>	-Djavax.net.ssl.trustStore=\${com.sun.aas.instanceRoot}/config/cacerts.jks
<input type="checkbox"/>	-Djava.ext.dirs=\${com.sun.aas.javaRoot}/lib/ext\${path.separator}\${com.sun.aas.javaRoot}/jre/lib/ext\${path.separator}\${com.sun.aas.in
<input type="checkbox"/>	-Djdbc.drivers=org.apache.derby.jdbc.ClientDriver
<input type="checkbox"/>	-DANTLR_USE_DIRECT_CLASS_LOADING=true
<input type="checkbox"/>	-Dcom.sun.enterprise.config.config_environment_factory_class=com.sun.enterprise.config.serverbeans.AppserverConfigEnvironmentf
<input type="checkbox"/>	-XX:NewRatio=2
<input type="checkbox"/>	-Xmx128m

Tras esto se para y se vuelve a arrancar el servidor para que se efectúen los cambios:

```
si2@si2srv01:~$ asadmin stop-cluster SI2Cluster
Enter admin user name> admin
Enter admin password for user "admin">
Command stop-cluster executed successfully.
si2@si2srv01:~$ asadmin start-cluster SI2Cluster
Enter admin user name> admin
Enter admin password for user "admin">
Command start-cluster executed successfully.
```

Tras ejecutar "ps -aefl | grep java"

En el nodo 1: (pid 3537)

```
si2@si2srv02:~$ ps -aefl | grep java
0 S si2 3537 1 2 80 0 - 137606 futex_ 13:20 ? 00:00:11 /usr/lib/jvm/java-8-oracle/bin/java -cp /opt/glassfish4/glassfish/modules/glassfish.jar -XX:+UnlockDiagnosticVMOptions -XX:NewRatio=2 -XX:MaxPermSize=96m -Xmx128m -Xms128m -server -javaagent:/opt/glassfish4/glassfish/lib/monitor/flashlight-agent.jar -Djavax.net.ssl.trustStore=/opt/glassfish4/Node01/Instance01/config/cacerts.jks -Djdk.corba.allowOutputStreamSubclass=true -Dfelix.fileinstall.dir=/opt/glassfish4/glassfish/modules/autostart/ -Dorg.glassfish.additionalOSGiBundlesToStart=org.apache.felix.shell,org.apache.felix.gogo.runtime,org.apache.felix.gogo.shell,org.apache.felix.gogo.command,org.apache.felix.fileinstall
```

En el nodo 2: (pid 3154)

```
si2@si2srv03:~$ ps -aefl | grep java
0 S si2 3154 1 7 80 0 - 135612 futex_ 13:20 ? 00:00:33 /usr/lib/jvm/java-8-oracle/bin/java -cp /opt/glassfish4/glassfish/modules/glassfish.jar -XX:+UnlockDiagnosticVMOptions -XX:NewRatio=2 -XX:MaxPermSize=96m -Xmx128m -Xms128m -server -javaagent:/opt/glassfish4/glassfish/lib/monitor/flashlight-agent.jar -Djavax.net.ssl.trustStore=/opt/glassfish4/Node02/Instance02/config/cacerts.jks -Djdk.corba.allowOutputStreamSubclass=true -Dfelix.fileinstall.dir=/opt/glassfish4/glassfish/modules/autostart/ -Dorg.glassfish.additionalOSGiBundlesToStart=org.apache.felix.shell,org.apache.felix.gogo.runtime,org.apache.felix.gogo.shell,org.apache.felix.gogo.command,org.apache.felix.fileinstall
```

Esto se verifica con el output del comando “asadmin list-instances --long=true”:

```
si2@si2srv01:~$ asadmin list-instances --long=true
Enter admin user name> admin
Enter admin password for user "admin">
Name          Host      Port    Pid    Cluster    State
Instance01    10.6.4.2  24848   3537   SI2Cluster  running
Instance02    10.6.4.3  24848   3154   SI2Cluster  running
Command list-instances executed successfully.
si2@si2srv01:~$
```

Ejercicio 3:

Pruebe a realizar un pago individualmente en cada instancia. Para ello, identifique los puertos en los que están siendo ejecutados cada una de las dos instancias (IPs 10.X.Y.2 y 10.X.Y.3 respectivamente). Puede realizar esa comprobación directamente desde la consola de administración, opción Applications, acción Launch, observando los Web Application Links generados. Realice un único pago en cada nodo. Verifique que el pago se ha anotado correctamente el nombre de la instancia y la dirección IP. Anote sus observaciones (puertos de cada instancia) y evidencias (captura de pantalla de la tabla de pagos).

Se copian los ficheros correspondientes en la P1-base renombrada

```
cations/j2ee_modules/P3$ cp -r /home/seregio/Escritorio/cositas/uni/3o/2o_cuatri
/si2/SI2/P3/P3-alumnos/datagen/listado.csv datagen/
seregio@seregio-B550-AORUS-ELITE-V2:~/glassfish4/glassfish/domains/domain1/appli
cations/j2ee_modules/P3$ cp -r /home/seregio/Escritorio/cositas/uni/3o/2o_cuatri
/si2/SI2/P3/P3-alumnos/sql/insert.sql sql/
```

Se han realizado las siguientes modificaciones en el código:

“ProcesaPago.java”:

```
VisaDAO dao = new VisaDAO();
//Debería coger la sesion anterior pero no lo hace
HttpSession sesion = (HttpSession)request.getSession(false);
if (sesion != null) {
    pago = (PagoBean) sesion.getAttribute(ComienzaPago.ATTR_P
}
if (pago == null) {
    pago = creaPago(request);
    boolean isdebug = Boolean.valueOf(request.getParameter("d
    //dao.setDebug(isdebug);
    dao.setDebug(debug:false); //Modo debug desactivado
```

“create.sql”:

```
-- Tabla con pagos autorizados
-- Siempre vienen precedidos por una transaccion existente
CREATE TABLE pago
(
  instancia      varchar(50) not null,
  ip             varchar(50) not null,
  -- idAutorizacion se autogenera con cada inserción
  idAutorizacion serial      not null,
  idTransaccion  char(16)    not null,
  codRespuesta   char(3)     not null default '000',
  importe        float       not null
```

“PagoBean.java”:

```
private String instancia;
private String ip;

/**
 * @return la instancia
 */
public String getInstancia() {
    return instancia;
}

/**
 * @param instancia la instancia
 */
public void setInstancia(String instancia) {
    this.instancia = instancia;
}

/**
 * @return la ip
 */
public String getIp() {
    return ip;
}

/**
 * @param ip la ip
 */
public void setIp(String ip) {
    this.ip = ip;
}
```


“ComienzaPago.java”:

```
private PagoBean creaPago(HttpServletRequest request) {
    PagoBean pago = new PagoBean();
    pago.setIdTransaccion(request.getParameter(PARAM_ID_TRANSACCION));
    pago.setIdComercio(request.getParameter(PARAM_ID_COMERCIO));
    double impd=-1.0;
    try {
        impd = Double.parseDouble(request.getParameter(PARAM_IMPORTE));
    } catch (NumberFormatException e) {
        impd = -1.0;
    } catch (NullPointerException e) {
        impd = -1.0;
    }
    pago.setImporte(impd);
    pago.setRutaRetorno(request.getParameter(PARAM_RUTA_RETORNO));

    //Setters P3
    pago.setInstancia(System.getProperty("com.sun.aas.instanceName"));
    try {
        pago.setIp(java.net.InetAddress.getLocalHost().getHostAddress());
    } catch (java.net.UnknownHostException e) {
        impd = -1.0;
    }

    return pago;
}
```

“ProcesaPago.java”:

```
private PagoBean creaPago(HttpServletRequest request) {
    PagoBean pago = new PagoBean();
    pago.setIdTransaccion(request.getParameter(PARAM_ID_TRANSACCION));
    pago.setIdComercio(request.getParameter(PARAM_ID_COMERCIO));

    double impd = -1.0;
    try {
        impd = Double.parseDouble(request.getParameter(PARAM_IMPORTE));
    } catch (NumberFormatException e) {
        impd = -1.0;
    } catch (NullPointerException e) {
        impd = -1.0;
    }

    pago.setImporte(impd);
    pago.setRutaRetorno(request.getParameter(PARAM_RUTA_RETORNO));

    //Setters P3
    pago.setInstancia(System.getProperty("com.sun.aas.instanceName"));
    try {
        pago.setIp(java.net.InetAddress.getLocalHost().getHostAddress());
    } catch (java.net.UnknownHostException e) {
        impd = -1.0;
    }

    return pago;
}
```

“VisaDao.java”:

Querys preparadas:

```
private static final String INSERT_PAGOS_QRY =
    "insert into pago(" +
    "idTransaccion,importe,idComercio,numeroTarjeta,instancia,ip)" +
    " values (?,?,?,?,?,?)";
```

```
if (isPrepared() == true) {
    String insert = INSERT_PAGOS_QRY;
    errorLog(insert);
    pstmt = con.prepareStatement(insert);
    pstmt.setString(1, pago.getIdTransaccion());
    pstmt.setDouble(2, pago.getImporte());
    pstmt.setString(3, pago.getIdComercio());
    pstmt.setString(4, pago.getTarjeta().getNumero());
    pstmt.setString(5, pago.getInstancia());
    pstmt.setString(6, pago.getIp());
    ret = false;
    if (!pstmt.execute()
        && pstmt.getUpdateCount() == 1) {
        ret = true;
    }
}
```

Querys no preparadas:

```
/**
 * getQryInsertPago
 */
String getQryInsertPago(PagoBean pago) {
    String qry = "insert into pago("
        + "idTransaccion,"
        + "importe,idComercio,"
        + "numeroTarjeta,instancia,ip)"
        + " values ("
        + "'" + pago.getIdTransaccion() + "',"
        + pago.getImporte() + ","
        + "'" + pago.getIdComercio() + "',"
        + "'" + pago.getTarjeta().getNumero() + "',"
        + "'" + pago.getInstancia() + "',"
        + "'" + pago.getIp() + "'"
        + ")";
    return qry;
}
```

“postgresql.xml”:

```
seregio@seregio-B550-A0RUS-ELITE-V2:~/glassfish4/glassfish/domains/domain1/applications/j2ee_modules/P3$ cp -r /home/seregio/Escritorio/cositas/uni/3o/2o_cuatri/si2/SI2/P3/P3-alumnos/postgresql.xml .
```

“build.properties”:

```
properties X
d.properties
# Propiedades de despliegue de aplicacion
nombre=P3
build=${basedir}/build

dist=${basedir}/dist

src=${basedir}/src

web=${basedir}/web

paquete=ssii2
war=${nombre}.war

asadmin=${as.home}/bin/asadmin
as.home=${env.J2EE_HOME}
as.lib=${as.home}/lib
as.user=admin
as.host=10.6.4.1

as.port=4848
as.passwordfile=${basedir}/passwordfile
as.target=SI2Cluster
```

“postgresql.properties”:

```
gresql.properties X
gresql.properties
# Propiedades de la BD postgresql

# Parametros propios de postgresql
db.name=visa
db.user=alumnodb
db.password=****
db.port=5432
db.host=10.6.4.1
# Recursos y pools asociados
db.pool.name=VisaPool
db.jdbc.resource.name=jdbc/VisaDB
db.url=jdbc:postgresql://${db.host}:${db.port}/${db.name}
db.client.host=10.6.4.1
db.client.port=4848

db.delimiter=;
db.driver=org.postgresql.Driver
db.datasource=org.postgresql.ds.PGConnectionPool
db.vendorname=SQL92

# Herramientas
db.createdb=/usr/bin/createdb
db.dropdb=/usr/bin/dropdb

# Scripts de creacion / borrado
db.create.src=./sql/create.sql
db.insert.src=./sql/insert.sql
db.delete.src=./sql/drop.sql
```

Como se puede ver el despliegue ha ido correctamente

```
setup-db:

montar-jerarquia:

compilar:
[javac] Compiling 17 source files to /home/seregio/glassfish4/glassfish/domains/domain1/applications/j2ee_modules/P3/build/WEB-INF/classes

preparar-web-inf:

empaquetar:
[jar] Building jar: /home/seregio/glassfish4/glassfish/domains/domain1/applications/j2ee_modules/P3/dist/P3.war

desplegar:
[exec] Application deployed with name P3.
[exec] Command deploy executed successfully.

BUILD SUCCESSFUL
Total time: 4 seconds
seregio@seregio-B550-AORUS-ELITE-V2:~/glassfish4/glassfish/domains/domain1/applications/j2ee_modules/P3$
```

Web links de cada instancia:

Web Application Links

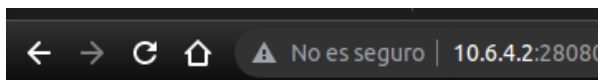
If the server or listener is not running, the link may not work. In this event, check the status of the serve

Application Name: P3

Links:

- [Instance01] <http://10.6.4.2:28080/P3>
- [Instance01] <https://10.6.4.2:28181/P3>
- [Instance02] <http://10.6.4.3:28080/P3>
- [Instance02] <https://10.6.4.3:28181/P3>

Se prueba el 10.6.4.2:



Pago con tarjeta

Numero de visa:

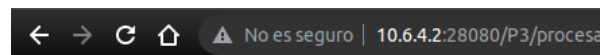
Titular:

Fecha Emisión:

Fecha Caducidad:

CVV2:

Id Transacción: 2
Id Comercion: 2
Importe: 2.0

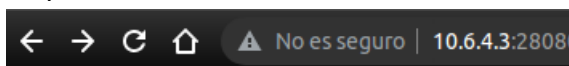


Pago con tarjeta

Pago realizado con éxito. A continuación se muestra el comprobant

idTransaccion: 2
idComercio: 2
importe: 2.0
codRespuesta: 000
idAutorizacion: 1

Se prueba el 10.6.4.2:



Pago con tarjeta

Numero de visa:

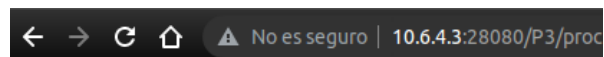
Titular:

Fecha Emisión:

Fecha Caducidad:

CVV2:

Id Transacción: 3
Id Comercion: 3
Importe: 3.0



Pago con tarjeta

Pago realizado con éxito. A continuación se muestra el comprob

idTransaccion: 3
idComercio: 3
importe: 3.0
codRespuesta: 000
idAutorizacion: 3

[Volver al comercio](#)

Se ejecuta la query “select * from pago;” en la base de datos y como se puede observar, el resultado es el esperado.

```
visa=# select * from pago;
instancia | ip      | idautorizacion | idtransaccion | codrespuesta | importe |
-----+-----+-----+-----+-----+-----+
Instance01 | 10.6.4.2 | 1 | 2 | 000 | 2 |
Instance02 | 10.6.4.3 | 3 | 3 | 000 | 3 |
(2 rows)
```

```
idcomercio | numerotarjeta | fecha
-----+-----+-----
2 | 1111 2222 3333 4444 | 2023-05-04 17:10:23.935139
3 | 1111 2222 3333 4444 | 2023-05-04 17:19:48.959971
```

Ejercicio 4:

Probar la influencia de jvmRoute en la afinidad de sesión.

1- Eliminar todas las cookies del navegador

2- Sin la propiedad jvmRoute, acceder a la aplicación P3 a través de la URL del balanceador:

http://10.X.Y.1/P3

3- Completar el pago con datos de tarjeta correctos.

4- Repetir los pagos hasta que uno falle debido a la falta de afinidad de sesión.

5- Mostrar la cookie “JSESSIONID” correspondiente a la URL del balanceador donde se vea:

Name: JSESSIONID

Content: YYYYYYYYYYYYYYYYYYYY

Domain: 10.X.Y.1 Path: /P3

6- Añadir la propiedad “jvmRoute” al cluster y rearrancar el cluster.

7- Eliminar todas las cookies del navegador.

8- Acceso a la aplicación P3 a través de la URL del balanceador: http://10.X.Y.1/P3

9- Completar el pago con datos de tarjeta correctos. Se pueden repetir los pagos y no fallarán.

10- Mostrar la cookie “JSESSIONID” correspondiente a la URL del balanceador donde se vea:

Name: JSESSIONID

Content: ZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZ

Domain: 10.X.Y.1 Path: /P3

Mostrar las pantallas y comentar: las diferencias en el contenido de las cookies respecto a jvmRoute, y cómo esta diferencia afecta a la afinidad y por qué. ¿Se podría, en general, usar el valor \${com.sun.aas.hostName} para la propiedad jvmRoute, en lugar de \${com.sun.aas.instanceName}?

Creamos el archivo de “proxy_balancer.conf”:

```
seregio@seregio-B550-AORUS-ELITE-V2: ~/Escritorio/cositas/uni/3o/2o_cuatri/si...
GNU nano 2.2.2 File: proxy_balancer.conf

ProxyRequests Off

<Proxy balancer://SI2Cluster>
  BalancerMember http://10.6.4.2:28080 route=Instance01
  BalancerMember http://10.6.4.3:28080 route=Instance02
</Proxy>

<Location /P3>
  Order allow,deny
  Allow from all
  ProxyPass balancer://SI2Cluster/P3 stickysession=JSESSIONID|jsessionid scolonpathdelim=On
  ProxyPassReverse balancer://SI2Cluster/P3
</Location>

<Location /balancer-manager>
  SetHandler balancer-manager
</Location>
```

Y se coloca en “/etc/apache2/mods-enabled” para que apache lo utilice (junto con sus módulos necesarios):

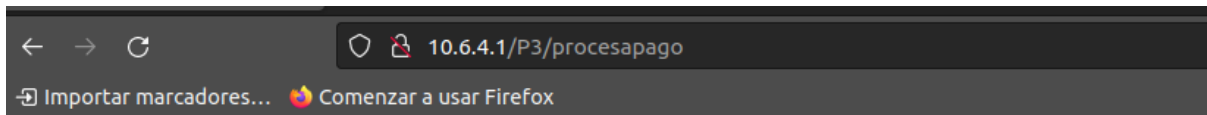
```
si2@si2srv01:/etc/apache2/mods-enabled$ cd /etc/apache2/mods-enabled
si2@si2srv01:/etc/apache2/mods-enabled$ sudo ln -sf ../mods-available/proxy_balancer.conf
si2@si2srv01:/etc/apache2/mods-enabled$ ls
alias.conf          authz_host.load    deflate.conf       mime.load          setenvif.conf
alias.load          authz_user.load    deflate.load       negotiation.conf   setenvif.load
auth_basic.load     autoindex.conf    dir.conf          negotiation.load   status.conf
authn_file.load     autoindex.load    dir.load          proxy_balancer.conf status.load
authz_default.load  cgid.conf         env.load          reqtimeout.conf
authz_groupfile.load cgid.load         mime.conf         reqtimeout.load
si2@si2srv01:/etc/apache2/mods-enabled$ sudo ln -sf ../mods-available/proxy.load
si2@si2srv01:/etc/apache2/mods-enabled$ sudo ln -sf ../mods-available/proxy_http.load
si2@si2srv01:/etc/apache2/mods-enabled$ sudo ln -sf ../mods-available/proxy_balancer.load
si2@si2srv01:/etc/apache2/mods-enabled$
```

Ahora toca crear una propiedad llamada “jvmRoute”, que modifica el identificador de sesión para saber qué instancia procesó la petición. En el cluster, dentro de “system properties”:

jvmRoute	\$(com.sun.aas.instanceName)
----------	------------------------------

Se reinicia el cluster para aplicar dichos cambios.

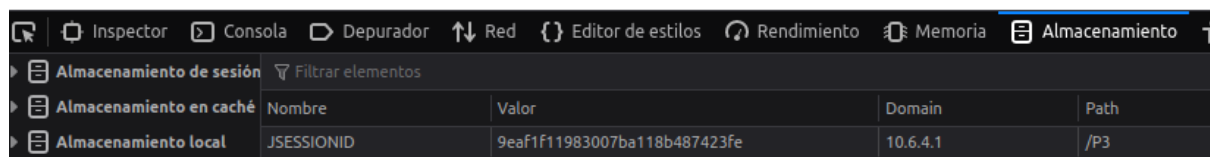
Sin el jvmRoute:



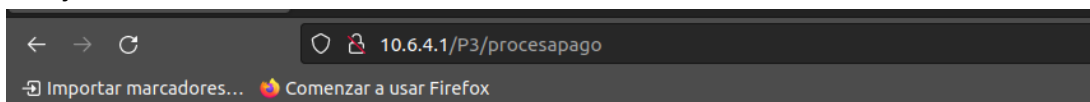
Pago con tarjeta

Pago incorrecto

Prácticas de Sistemas Informáticos II



Con jvmRoute:



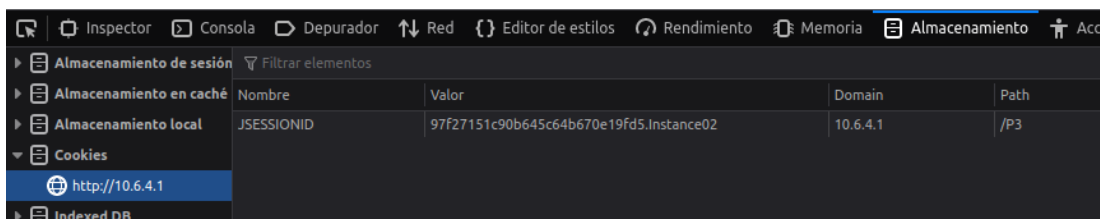
Pago con tarjeta

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

idTransaccion: 2
idComercio: 2
importe: 2.0
codRespuesta: 000
idAutorizacion: 1

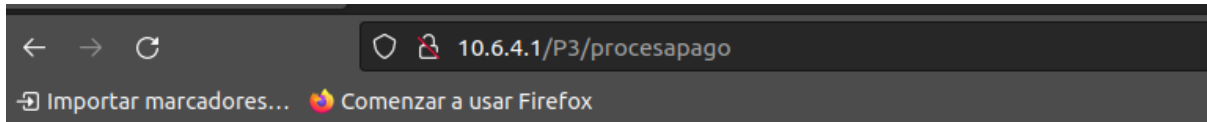
[Volver al comercio](#)

Prácticas de Sistemas Informáticos II



Sin el `jvmRoute` tiende a fallar, porque se desconoce qué instancia ha procesado la información y por tanto no se puede procesar la información. Además, al final de la cookie aparece el identificador de la instancia,

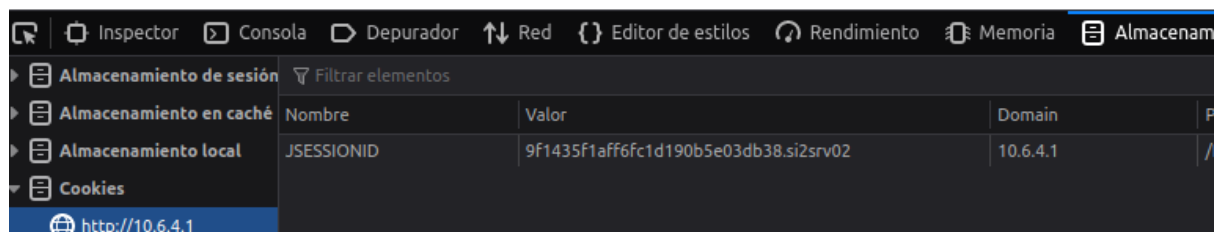
No se puede utilizar `${com.sun.aas.hostName}` pues cada host va relacionado a una instancia como ruta para el balanceador.



Pago con tarjeta

Pago incorrecto

Prácticas de Sistemas Informáticos II



Ejercicio 5: Probar el balanceo de carga y la afinidad de sesión, realizando un pago directamente contra la dirección del cluster `http://10.X.Y.1/P3` desde distintos ordenadores. Comprobar que las peticiones se reparten entre ambos nodos del cluster, y que se mantiene la sesión iniciada por cada usuario sobre el mismo nodo. Comentad la información mostrada en la página del Load Balancer Manager.

Desde el portátil:

10.6.4.1/P3/comienzapago

Parece que no ha iniciado Firefox desde hace tiempo. ¿Quiere limpiarlo para comenzar una experiencia

Pago con tarjeta

Numero de visa:

1111 2222 3333 4444

Titular:

Jose Garcia

Fecha Emisión:

11/09

Fecha Caducidad:

11/24

CVV2:

123

Pagar

Id Transacción: 2

Id Comercion: 2

Importe: 2.0

Prácticas de Sistemas Informáticos II

10.6.4.1/P3/procesapago

Parece que no ha iniciado Firefox desde hace tiempo. ¿Quiere limpiarlo para comenzar una experiencia

Pago con tarjeta

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

idTransaccion: 2

idComercio: 2

importe: 2.0

codRespuesta: 000

idAutorizacion: 1

[Volver al comercio](#)

Prácticas de Sistemas Informáticos II

http://10.6.4.1

Filtrar elementos

Almacenamiento de sesión

Almacenamiento en caché

Almacenamiento local

Cookies

Nombre	Valor	Domain
JSESSIONID	990cf7b12af31297eba990ddaf47.Instance01	10.6.4.1

Como se puede apreciar todo ha salido de manera correcta, el balanceador asignó la Instancia 1 para responder.

[OBJ] Desde el de sobremesa (de manera simultánea):

← → ↻ 10.6.4.1/P3/comienzapago

Importar marcadores... Comenzar a usar Firefox

Pago con tarjeta

Numero de visa:

Titular:

Fecha Emisión:

Fecha Caducidad:

CVV2:

Id Transacción: 1
Id Comercion: 1
Importe: 1.0

Prácticas de Sistemas Informáticos II

Inspector Console Depurador Red Editor de estilos Rendimiento Memoria Almacenamiento

Almacenamiento de sesión Filtrar elementos

Almacenamiento en caché

Nombre	Valor	Domain	Path
JSESSIONID	98fa6e7aa972005fff6dd6d6c973.Instance02	10.6.4.1	/P3

Almacenamiento local

Cookies

http://10.6.4.1

← → ↻ 10.6.4.1/P3/procesapago

Importar marcadores... Comenzar a usar Firefox

Pago con tarjeta

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

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

[Volver al comercio](#)

Prácticas de Sistemas Informáticos II

Inspector Console Depurador Red Editor de estilos Rendimiento Memoria Almacenamiento

Almacenamiento de sesión Filtrar elementos

Almacenamiento en caché

Nombre	Valor	Domain	Path
JSESSIONID	99109551a6d6a85380ba44d8486b.Instance02	10.6.4.1	/P3

Almacenamiento local

Cookies

http://10.6.4.1

En este el balanceador asignó la Instancia 2 (al estar ya la primera ocupada con una solicitud). También se ha realizado correctamente.

Load Balancer Manager for 10.6.4.1

Server Version: Apache/2.2.14 (Ubuntu)
Server Built: Nov 3 2011 03:31:27

LoadBalancer Status for balancer://si2cluster

StickySession	Timeout	FailoverAttempts	Method
JSESSIONID jsessionid	0	1	byrequests

Worker URL	Route	RouteRedir	Factor	Set	Status	Elected To	From
http://10.6.4.2:28080	Instance01		1	0	Ok	17	11K 12K
http://10.6.4.3:28080	Instance02		1	0	Ok	16	10K 12K

Apache/2.2.14 (Ubuntu) Server at 10.6.4.1 Port 80

El tráfico se ve reflejado entre las columnas to y from (además de que evidentemente el valor de elected será mayor en aquellas que han sido elegidas).

Ejercicio 6: Comprobación del proceso de fail-over. Parar la instancia del cluster que haya tenido menos elecciones hasta el momento. Para ello, identificaremos el pid (identificador del proceso java) de la instancia usando las herramientas descritas en esta práctica o el mandato ‘ps -aef | grep java’. Realizaremos un kill -9 pid en el nodo correspondiente. Vuelva a realizar peticiones y compruebe (accediendo a la página /balancer-manager y revisando el contenido de la base de datos) que el anterior nodo ha sido marcado como “erróneo” y que todas las peticiones se dirijan al nuevo servidor. Adjunte la secuencia de comandos y evidencias obtenidas en la memoria de la práctica.

La instancia 2 ha sido la menos elegida así que se ejecutará ‘ps -aef | grep java’ en la terminal de su instancia y “kill -9 pid” de dicho proceso (3610):

```
si2@si2srv03:~$ ps -aef | grep java
si2      3610      1  2 18:36 ?        00:00:44 /usr/lib/jvm/java-8-oracle/bin/java -cp /opt/gl
4/glassfish/modules/glassfish.jar -XX:+UnlockDiagnosticVMOptions -XX:NewRatio=2 -XX:MaxPermSize
mx128m -Xms128m -server -javaagent:/opt/glassfish4/glassfish/lib/monitor/Flashlight-agent.jar -
net.ssl.trustStore=/opt/glassfish4/Node02/Instance02/config/cacerts.jks -Djdk.corba.allowOutput
ubclass=true -Dfelix.fileinstall.dir=/opt/glassfish4/glassfish/modules/autostart/ -Dorg.glassfi
tionalOSGiBundlesToStart=org.apache.felix.shell.org.apache.felix.gogo.runtime.org.apache.felix.
ell.org.apache.felix.gogo.command.org.apache.felix.fileinstall -Dcom.sun.aas.installRoot=/opt/g
h4/glassfish -Dfelix.fileinstall.poll=5000 -Djava.security.policy=/opt/glassfish4/Node02/Instan
nfig/server.policy -Djava.endorsed.dirs=/opt/glassfish4/glassfish/modules/endorsed:/opt/glassfi
ssfish/lib/endorsed -Dfelix.fileinstall.bundles.startTransient=true -Dosgi.shell.telnet.maxconn
lix.fileinstall.log.level=3 -Dcom.sun.enterprise.config.config_environment_factory_class=com.su
prise.config.serverbeans.AppserverConfigEnvironmentFactory -Djavax.net.ssl.keyStore=/opt/glassf
de02/Instance02/config/keyStore.jks -Djava.security.auth.login.config=/opt/glassfish4/Node02/In
2/config/login.conf -Dfelix.fileinstall.disableConfigSave=false -Dfelix.fileinstall.bundles.new
true -Dcom.sun.aas.instanceRoot=/opt/glassfish4/Node02/Instance02 -Dosgi.shell.telnet.port=2666
h.args=-noshutdown -c noop=true -Dcom.sun.enterprise.security.httpsOutboundKeyAlias=s1as -Dosg
i.telnet.ip=127.0.0.1 -DANTLR_USE_DIRECT_CLASS_LOADING=true -Djava.awt.headless=true -Djava.ext.
sr/lib/jvm/java-8-oracle/lib/ext:/usr/lib/jvm/java-8-oracle/jre/lib/ext:/opt/glassfish4/Node02/
e02/lib/ext -Djdbc.drivers=org.apache.derby.jdbc.ClientDriver -Djava.library.path=/opt/glassfis
sfish/lib:/usr/java/packages/lib/i386:/lib:/usr/lib com.sun.enterprise.glassfish.bootstrap.ASMa
rade false -read-stdin true -asadmin-args --host,,si2srv01,,--port,,4848,,--secure=false,,,
=false,,,--echo=false,,,--interactive=false,,,start-local-instance,,,--verbose=false,,,--watchd
e,,,--debug=false,,,--nodetdir,,,/opt/glassfish4,,,--node,,,Node02,,,Instance02 -instancename In
2 -type INSTANCE -verbose false -instancedir /opt/glassfish4/Node02/Instance02 -asadmin-classpa
/glassfish4/glassfish/modules/admin-cli.jar -debug false -asadmin-classname com.sun.enterprise.
li.AdminMain
si2      3909  3902  0 19:04 pts/0    00:00:00 grep java
si2@si2srv03:~$ kill -9 3610
si2@si2srv03:~$ ps -aef | grep java
si2      3912  3902  0 19:04 pts/0    00:00:00 grep java
```

Efectivamente, tras realizar el kill y conectarse para realizar el pago, el tráfico será redirigido ala instancia 1 y la 2 aparecerá como erronea.

Load Balancer Manager for 10.6.4.1

Server Version: Apache/2.2.14 (Ubuntu)
Server Built: Nov 3 2011 03:31:27

LoadBalancer Status for balancer://si2cluster

StickySession	Timeout	FailoverAttempts	Method
JSESSIONID jsessionid 0	3		byrequests

Worker URL	Route	RouteRedir	Factor	Set	Status	Elected To	From
http://10.6.4.2:28080	Instance01		1	0	Ok	49	28K 41K
http://10.6.4.2:28181	Instance01		1	0	Ok	53	31K 0
http://10.6.4.3:28080	Instance02		1	0	Err	43	27K 30K
http://10.6.4.3:28181	Instance02		1	0	Err	38	22K 0

Apache/2.2.14 (Ubuntu) Server at 10.6.4.1 Port 80

←

→

↺

10.6.4.1/P3/comienzapago

Importar marcadores... Comenzar a usar Firefox

Pago con tarjeta

Numero de visa:

1111 2222 3333 4444

Titular:

Jose Garcia

Fecha Emisión:

11/09

Fecha Caducidad:

11/24

CVV2:

123

Pagar

Id Transacción: 1

Id Comercion: 1

Importe: 1.0

Prácticas de Sistemas Informáticos II

Inspector

Consola

Depurador

Red

Editor de estilos

Rendimiento

Almacenamiento local

Filtrar elementos

Almacenamiento de sesión

Nombre

Valor

Almacenamiento en caché

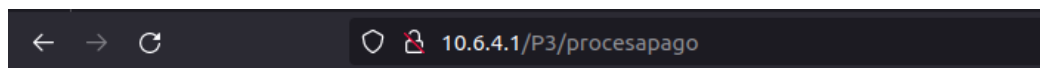
JSESSIONID

9a91f74b6f2cb054bb60b26691c3.Instance01

Ejercicio 7: Comprobación del proceso de fail-back. Inicie manualmente la instancia detenida en el comando anterior. Verificar la activación de la instancia en el gestor del balanceador. Incluir todas las evidencias en la memoria de prácticas y comentar qué sucede con los nuevos pagos. Consulte los apéndices para información detallada de comandos de gestión individual de las instancias.

Se utiliza el comando “asadmin start-instance Instance02” para volver a iniciar la instancia anteriormente terminada.

```
seregio@seregio-B550-AORUS-ELITE-V2: ~/Escritorio/cositas/uni/3o/2o_cuatri/si2/SI2/P3
si2@si2srv01:/etc/apache2/mods-enabled$ asadmin list-instances -l
Enter admin user name> admin
Enter admin password for user "admin">
Name      Host      Port    Pid    Cluster    State
Instance01 10.6.4.2  24848   3876   SI2Cluster running
Instance02 10.6.4.3  24848   --     SI2Cluster not running
Command list-instances executed successfully.
si2@si2srv01:/etc/apache2/mods-enabled$ asadmin start-instance Instance02
Enter admin user name> admin
Enter admin password for user "admin">
Waiting for Instance02 to start .....
Successfully started the instance: Instance02
instance Location: /opt/glassfish4/Node02/Instance02
Log File: /opt/glassfish4/Node02/Instance02/logs/server.log
Admin Port: 24848
Command start-local-instance executed successfully.
The instance, Instance02, was started on host 10.6.4.3
Command start-instance executed successfully.
si2@si2srv01:/etc/apache2/mods-enabled$ asadmin list-instances -l
Enter admin user name> admin
Enter admin password for user "admin">
Name      Host      Port    Pid    Cluster    State
Instance01 10.6.4.2  24848   3876   SI2Cluster running
Instance02 10.6.4.3  24848   4203   SI2Cluster running
Command list-instances executed successfully.
si2@si2srv01:/etc/apache2/mods-enabled$
```



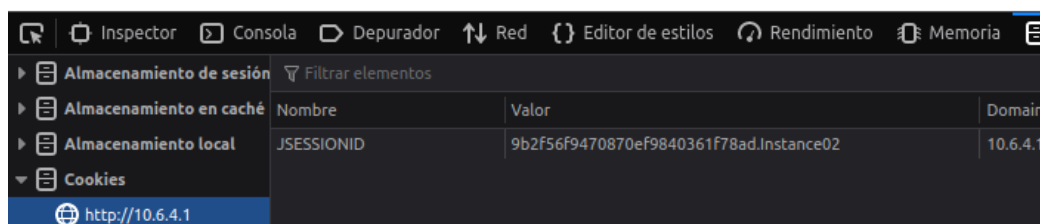
Pago con tarjeta

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

idTransaccion: 3
idComercio: 3
importe: 4.0
codRespuesta: 000
idAutorizacion: 2

[Volver al comercio](#)

Prácticas de Sistemas Informáticos II



Como era de esperar, una vez se ha vuelto a iniciar la instancia, los pagos son perfectamente realizables desde la misma y el balanceador dividirá el tráfico entre ambas instancias de nuevo.

The screenshot shows a web browser at the URL `10.6.4.1/P3/procesapago`. The page title is "Pago con tarjeta". The main content states: "Pago realizado con éxito. A continuación se muestra el comprobante del mismo:" followed by transaction details: `idTransaccion: 1`, `idComercio: 1`, `importe: 1.0`, `codRespuesta: 000`, and `idAutorizacion: 1`. A link "Volver al comercio" is present. Below the page content, the Firefox developer tools "Almacenamiento" (Storage) tab is open, showing a table of cookies:

Nombre	Valor	Domain	Path	Expires
JSESSIONID	9b18f1c41fcb0049e3c170593f34.Instance01	10.6.4.1	/P3	Sesión

A terminal window shows a table with 6 columns: `instancia`, `ip`, `idautorizacion`, `idtransaccion`, `codrespuesta`, and `importe`. The data is as follows:

instancia	ip	idautorizacion	idtransaccion	codrespuesta	importe
Instance01	10.6.4.2	1	1	000	1
Instance02	10.6.4.3	2	3	000	4

(2 rows)

A terminal window shows a table with 3 columns: `idcomercio`, `numerotarjeta`, and `fecha`. The data is as follows:

idcomercio	numerotarjeta	fecha
1	1111 2222 3333 4444	2023-05-04 19:15:41.512766
3	1111 2222 3333 4444	2023-05-04 19:17:12.800444

Capturas de la base de datos.

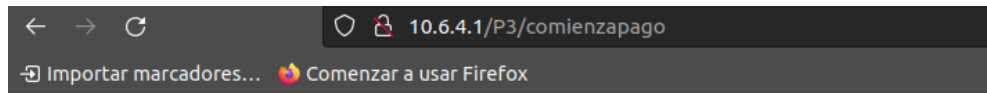
Ejercicio 8: Fallo en el transcurso de una sesión.

Desde un navegador, comenzar una petición de pago introduciendo los valores del mismo en la pantalla inicial y realizando la llamada al servlet `ComienzaPago`.

Al presentarse la pantalla de "Pago con tarjeta", leer la instancia del servidor que ha procesado la petición y detenerla.

Se puede encontrar la instancia que ha procesado la petición revisando la cookie de sesión (tiene la instancia como sufijo), el `balancer-manager` o el `server.log` de cada instancia. Completar los datos de la tarjeta de modo que el pago fuera válido, y enviar la petición. Observar la instancia del cluster que procesa el pago, y razonar las causas por las que se rechaza la petición.

Se inicia el pago (Es la instancia 2 la encargada de dicho pago)



Pago con tarjeta

Numero de visa:

Titular:

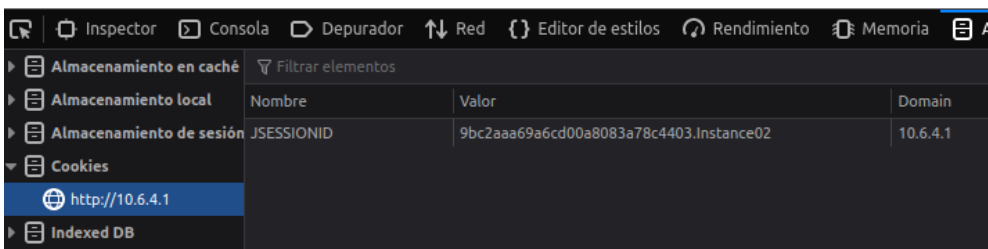
Fecha Emisión:

Fecha Caducidad:

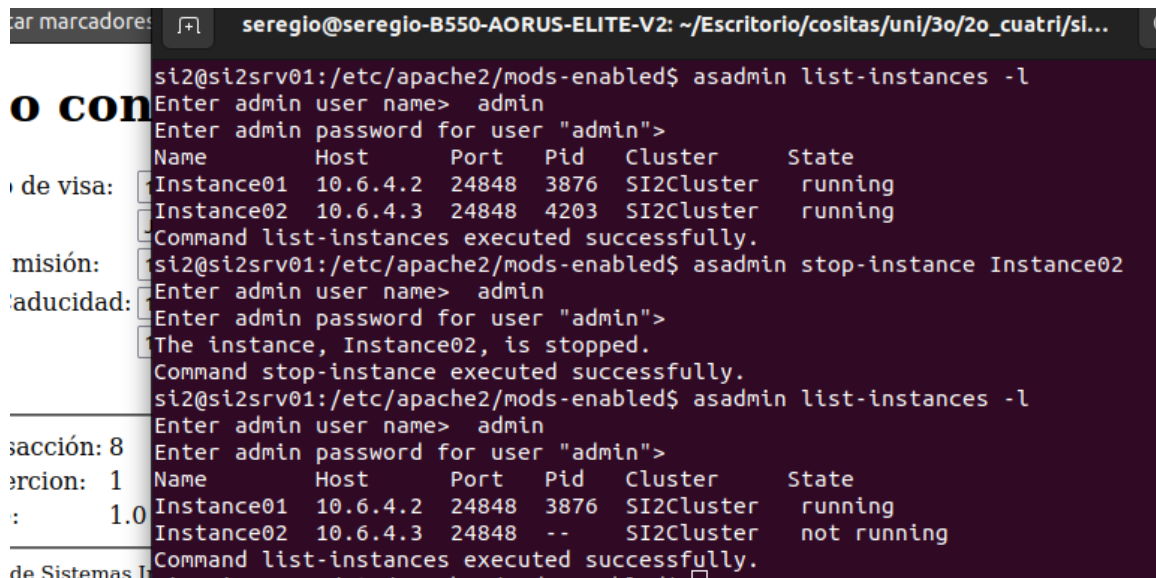
CVV2:

Id Transacción: 8
Id Comercion: 1
Importe: 1.0

Prácticas de Sistemas Informáticos II

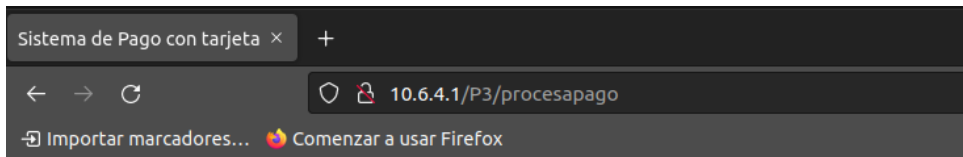


Se utilizará el comando “asadmin stop-instance Instance02”



Se rechaza la petición porque al perder la instancia que estaba atendiendo la solicitud, el balanceador pasa dicha solicitud a la otra instancia, que al leer la cookie no la reconoce

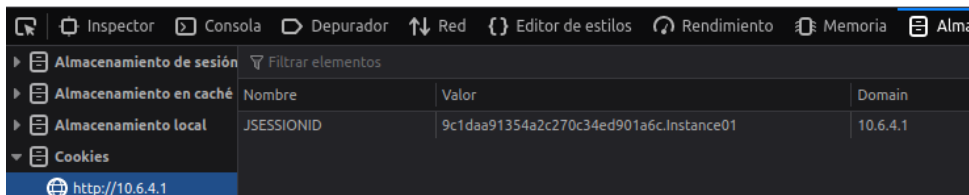
como válida. no se reconoce como válida la sesión



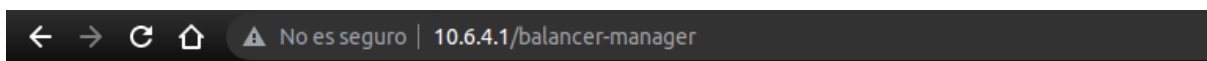
Pago con tarjeta

Tarjeta no autorizada:

Prácticas de Sistemas Informáticos II



Cabe destacar que el load manager vuelve a mostrar la instancia con error.



Load Balancer Manager for 10.6.4.1

Server Version: Apache/2.2.14 (Ubuntu)
Server Built: Nov 3 2011 03:31:27

LoadBalancer Status for balancer://si2cluster

StickySession	Timeout	FailoverAttempts	Method
JSESSIONID jsessionid	0	1	byrequests

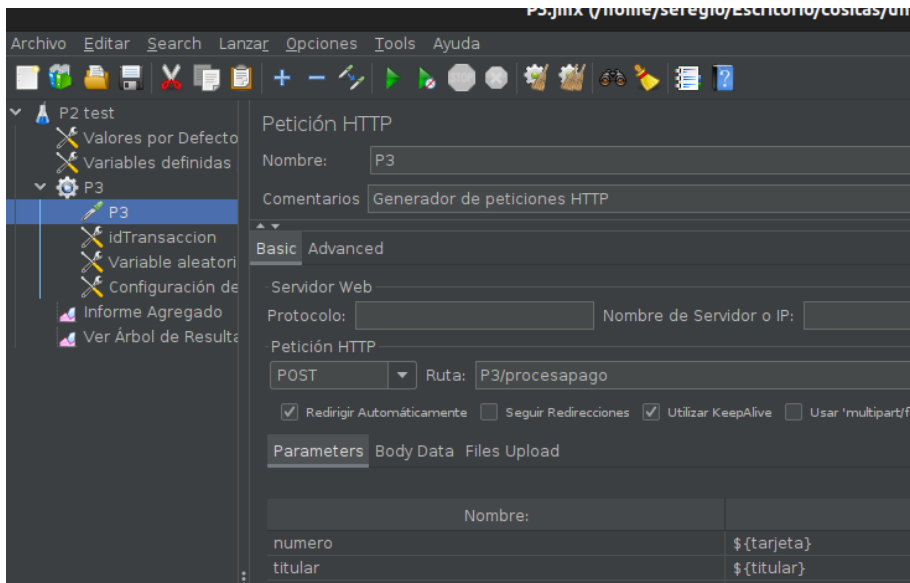
Worker URL	Route	RouteRedir	Factor	Set	Status	Elected	To	From
http://10.6.4.2:28080	Instance01		1	0	Ok	31	20K	25K
http://10.6.4.3:28080	Instance02		1	0	Err	23	13K	17K

Apache/2.2.14 (Ubuntu) Server at 10.6.4.1 Port 80

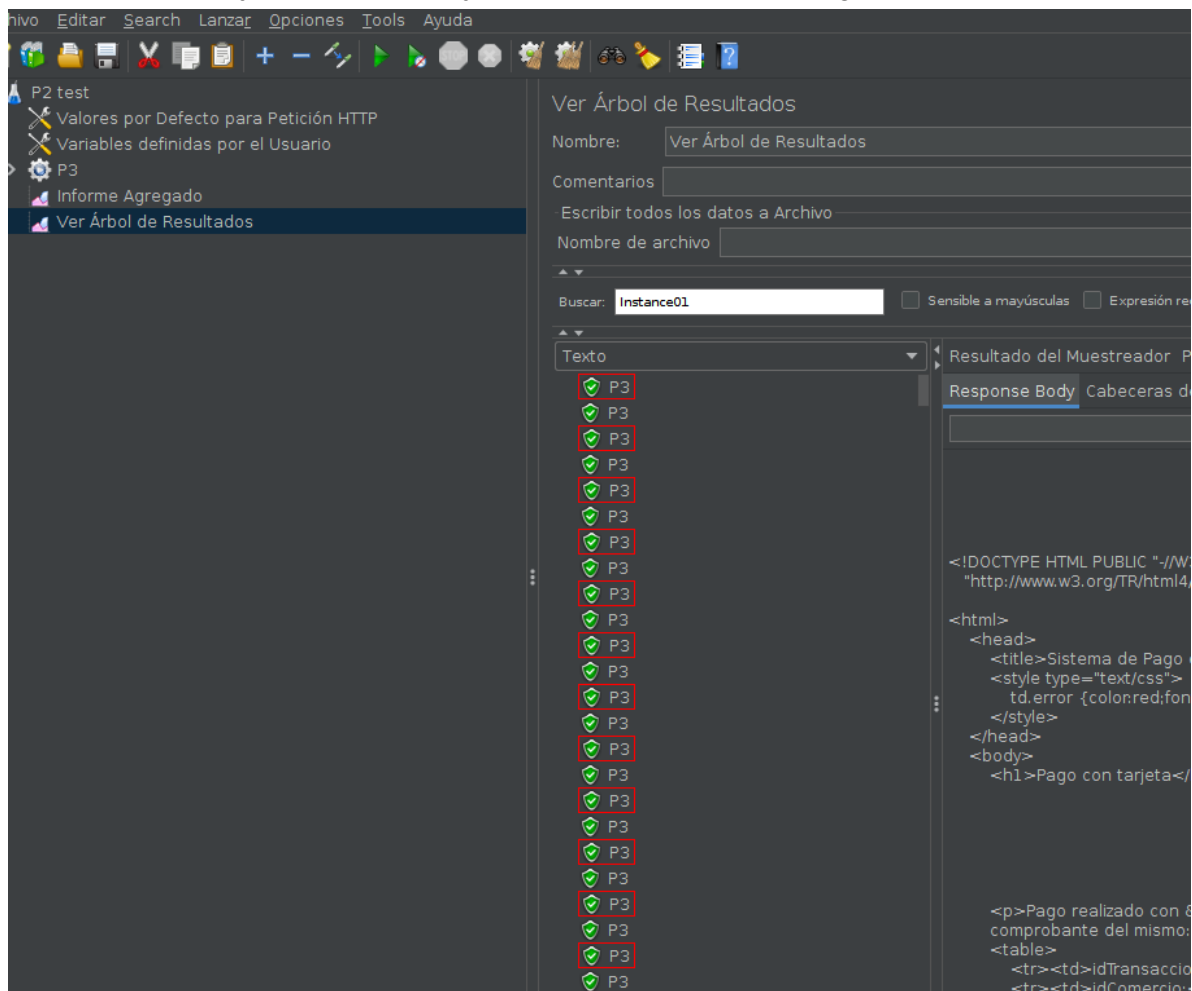
Cabe destacar que hasta que la cookie no fue borrada no se pudo realizar el pago.

Ejercicio 9: Modificar el script de pruebas JMeter desarrollado durante la P2. (P2.jmx) Habilitar un ciclo de 1000 pruebas en un solo hilo contra la IP del cluster y nueva URL de la aplicación: <http://10.X.Y.1/P3>
Eliminar posibles pagos previos al ciclo de pruebas. Verificar el porcentaje de pagos realizados por cada instancia, así como (posibles) pagos correctos e incorrectos.
¿Qué algoritmo de reparto parece haber seguido el balanceador? Comente todas sus conclusiones en la memoria de prácticas.

Se han cambiado tanto la ip de los valores por defecto como la ruta y el nombre del hilo.



No obstante se dejará el archivo adjuntado en el directorio entregado.



Parece seguir round-robin pues se van alternando en todo momento las peticiones