

SISTEMAS INFORMÁTICOS II – PRÁCTICAS – PREGUNTAS DE TESTS

P1A:

¿Cuándo utilizarías el api JDBC?

Cuando tenga que realizar operaciones sobre una base de datos desde el lenguaje de programación Java.

En una arquitectura de tres capas, donde se establecen las reglas del negocio, se corresponde a la capa de ...
... lógica de negocio.

Los servlets:

Son objetos java que se ejecutan en el servidor de aplicaciones en el contenedor de servlets.

Señala cuál se corresponde con una directiva JSP:

page

Indicar cuál de los siguientes elementos podríamos encontrarnos en un archivo WDSL:

import, portType, message.

De los siguientes pasos, ¿cuál es incorrecto para desplegar un servlet en un servidor web?

Implementando la clase del servlet, que implementará la interfaz de HttpServlet.

En el api java.sql, el objeto PreparedStatement, se usa para:

Ejecutar una sentencia SQL de forma repetida, variando sus parámetros.

En una aplicación web que está utilizando servlets, el fichero de configuración que mapea una URL al nombre del servlet, es ...

web.xml

Para transferir la petición a otra página JSP, se utiliza:

La etiqueta forward.

De los siguientes métodos, ¿cuál es el que deberíamos usar para obtener un parámetro de inicialización desde ServletContext?

String getInitParameter(String name).

P1B:

Los Java Beans...

...no pueden utilizarse en entornos de objetos distribuidos al no soportar nativamente la invocación remota (RMI).

¿Cuál de estas afirmaciones es verdadera?

a. Los archivos EAR (Enterprise Application aRchive) empaquetan uno o más módulos para hacer el despliegue en el servidor de aplicaciones de manera simultánea y coherente.

b. Todas las afirmaciones son verdaderas.

c. Los archivos JAR (Java ARchive) empaquetan archivos Java, Metadatos y recursos asociados. No tienen una jerarquía de archivos específica.

d. Los archivos WAR (Web aplicación ARchive) empaquetan archivos de una aplicación web (archivos Java, Servlets, Java Server Pages - JSP, páginas web estáticas...).

¿Cuál de estas afirmaciones es falsa?

Los EJB de Sesión sin estado son objetos distribuidos que carecen de estado asociado por tanto no pueden tener variables de instancia.

En la aplicación distribuida formada por P1-ejb-servidor-remoto y P1-ejb-cliente-remoto, la clase que se encarga de realizar consultas y cambios en la base de datos, y que por tanto es el “cliente” de la base de datos es:

VisaDAOBean del proyecto P1-ejb-servidor-remoto.

Indica si la siguiente afirmación es verdadera o falsa: “Java Messaging Service (JMS) tiene dos tipos de destinos de mensajes a los que conectarse para enviar/obtener mensajes: colas de mensaje (Message Queue) y Tema de mensaje (Message Topic)”

Verdadero.

¿Para qué hemos usado las colas de mensajes en la práctica P1B?

Para cancelar un pago realizado a través de la aplicación web.

Indica si la siguiente afirmación es verdadera o falsa: “Los clientes locales de EJB dan más flexibilidad al poder distribuir la aplicación después”

Falso.

Indica si la siguiente afirmación es verdadera o falsa: “Para un cliente remoto de un EJB la ubicación del EJB es transparente”

Verdadero.

¿Cómo se llaman los elementos que envían y reciben mensajes de una cola de mensajes?

Productor y consumidor.

¿Cuál de estas propiedades es una característica específica de los EJB dirigidos por mensajes (Message-driven EJBs)?

Proporciona comunicación asíncrona entre el servidor y el cliente.

P2:

Si al final de la ejecución de un plan de pruebas en JMeter, la columna "Error" del reporte agregado marca 0,00%...:

...podemos asegurar que el código HTTP de retorno de cada petición realizada es 200 (OK)

Indica si la siguiente afirmación es verdadera o falsa: "nmon es un programa que permite monitorizar el estado de diferentes componentes de una instancia de Glassfish, tales como el estado del pool de conexiones HTTP."

Falso

El periodo de despegue (o "ramp-up period") de la configuración de un grupo de hilos en JMeter:

Debe ser lo suficientemente largo para evitar una gran carga de trabajo al principio de la prueba, y lo suficientemente corto para que los últimos hilos comiencen antes de que los primeros terminen.

Suponga que se realiza una prueba de P1-base con un grupo de hilos configurado con los siguientes parámetros: (Número de hilos = 10, Periodo de despegue = 20, Contador de bucle = 5). Si la prueba se ejecuta con éxito, el número de pagos que se deberán ver en la base de datos es:

50

¿Qué comando nmon utilizarías para volcar a fichero una monitorización de 1 minuto de duración, en la que se tome 1 instantánea cada 5 segundos?

nmon -f -c 12 -s 5

¿Cuál de estas afirmaciones es falsa?

El punto de saturación de un sistema distribuido es el punto en el que se alcanza el máximo throughput.

Indica si la siguiente frase es verdadera: "La variable aleatoria think time que se utiliza en el plan de pruebas para obtener la curva de productividad permite simular el tiempo que un usuario pasa pensando antes de enviar los datos de un formulario"

Verdadero

Suponga que, con la base de datos vacía, ejecutamos un plan de pruebas contra P1-base, y al terminar la prueba encontramos 3000 pagos en la base de datos. Si volvemos a ejecutar el plan de pruebas sin limpiar la base de datos, el número de pagos al final de la segunda prueba será de:

3000

Indica cuál de las siguientes afirmaciones es correcta:

El script si2-monitor.sh debe ejecutarse una vez cumplido el periodo de despegue de una prueba en JMeter, para tomar instantáneas del estado del servidor de aplicaciones cuando la prueba se encuentra en el momento de máxima carga.

El número máximo de peticiones que el servidor puede atender simultáneamente se configura en el parámetro:

Max Thread Pool Size.

P3:

Para configurar la afinidad de sesión en una aplicación desplegada en un cluster de Glassfish que se accede desde Apache:

El valor de la propiedad `jvmRoute` del cluster debe coincidir con alguno de los valores del atributo `route` de los `BalancerMember`.

Una aplicación Java desplegada en un cluster Glassfish puede obtener el nombre de la instancia en que se ejecuta:

Accediendo a una propiedad del entorno mediante la llamada:

`System.getProperty("com.sun.aas.instanceName");`

Respecto de las instancias de los clusters de Glassfish:

Para crear una instancia hay que indicar en qué nodo se encuentra y a qué cluster pertenece: los dos parámetros son necesarios.

Respecto de los clusters de Glassfish:

Un nodo puede contener varias instancias.

Un cluster puede contener varias instancias.

Un cluster de instancias de servidores de aplicaciones no proporciona:

Un hardware más potente y eficaz.

Al crear un nodo en Glassfish se requiere indicar a qué cluster pertenece:

Falso.

El protocolo ssh soporta:

Autenticación de la sesión.

Transferencia de archivos.

Port tunneling.

Encriptado de los datos transferidos.

Al configurar un balanceador de carga mediante el módulo `mod_proxy_balancer` de Apache, se ha de indicar en el fichero de configuración `proxy_balancer.conf`:

Las parejas de IP y puerto a las que se va a balancear en la sección `<Proxy balancer://[nombre-cluster-balanceo]>`.

Las diferencias en el fichero de configuración `proxy_balancer.conf` del módulo `mod_proxy_balancer` de Apache entre un balanceador sin afinidad de sesión y otro que sí la contempla son:

Los atributos `stickysession` y `scolonpathdelim` en la directiva `ProxyPass` y el atributo `route` de los `BalancerMember`.

Para iniciar una sesión ssh sin introducir la contraseña:

Se genera un par de claves, pública y privada, en el cliente y se coloca la pública en el fichero `$HOME/.ssh/authorized_keys2` del servidor.

SISTEMAS INFORMÁTICOS II – PRÁCTICAS – PREGUNTAS DE DESARROLLO

EJERCICIO 1.- Este ejercicio trata sobre el servicio web con el que se ha trabajado en la práctica P1A. Supóngase que se quiere poder invocar el servicio web independientemente de su ubicación, sin tener que recompilar o redespargar el código de la aplicación cliente. Para ello, supondremos que el servlet `ProcesaPago.java` recibe en la petición HTTP (tanto GET como POST) un parámetro adicional llamado “ipservicioweb” que contiene la dirección IP asociada a la máquina virtual donde se encuentra el servicio web.

Se pide describir las modificaciones que tendríamos que realizar, así como la instrucción JAVA necesaria para instanciar el stub del cliente y así conseguir la funcionalidad descrita en el párrafo anterior. ¿Qué conseguiríamos con esta implementación?

0,70/1,00

Para poder obtener el valor de un parámetro del formulario `pago.html` o `testbd.jsp` es primero necesario añadirlo al formulario.

A la dirección ip recibida hay que añadir "8080/P1-ws-ws/VisaDAOWSService" para encontrar el servicio web

Haría falta incluir el nuevo parámetro de la ip donde se encuentra el servicio en las constantes iniciales de la clase `ProcesaPago` de la siguiente manera:

```
public final static String PARAM_IP = "ipServidor";
```

De esta forma podríamos recibir la IP del servidor desde la petición http con el resto de los parámetros del formulario de pago. Tras añadir esto, serviría con incluir la creación de los stubs dinámicos mediante las siguientes instrucciones:

```
VisaDAOWSService service = new VisaDAOWSService();
VisaDAOWS dao = service.getVisaDAOWSPort();
BindingProvider bp = (BindingProvider) dao;
try{
    bp.getRequestContext().put(BindingProvider.ENDPOINT_ADDRESS_PROPERTY, PARAM_IP );
} catch(Exception e){
    enviaError(new Exception("Direccion del servidor no encontrada."), request, response);
    return;
}
```

Con el binding provider podremos generar el stub a partir de la dirección introducida en la que se encuentra el servidor. Se tendrían que incluir los imports necesarios para `VisaDaoWS` y `BindingProvider`:

```
import ssii2.visa.VisaDAOWSService; // Stub generado automáticamente
import ssii2.visa.VisaDAOWS; // Stub generado automáticamente
import javax.xml.ws.WebServiceRef;
import javax.xml.ws.BindingProvider;
```

0,80/1,00 ¿Y dónde cogeríamos el valor de ipservicioweb? Habría que añadir un nuevo campo "IPSERVICIOWEB" en los formularios formdatosvisa.jsp y tstadb.jsp

Tendríamos que modificar el servlet ProcesaPago cambiando:

```
bp.getRequestContext().put(BindingProvider.ENDPOINT_ADDRESS_PROPERTY,  
getServletContext().getInitParameter("webmaster") );
```

```
String primero="http://"
```

```
String segundo = ":8080/P1-ws-ws/VisaDAOWSService"
```

```
bp.getRequestContext().put(BindingProvider.ENDPOINT_ADDRESS_PROPERTY, primero + ipservicioweb +  
segundo );
```

Comando:

```
wsimport -d build/client/WEB-INF/classes -p ssii2.visa http://ipservicioweb:8080/P1-ws-  
ws/VisaDAOWSService?WSDL
```

Conseguimos transparencia de localización

EJERCICIO 2.- Se han realizado diversas pruebas sobre algunos de los desarrollos que se han visto a lo largo de las prácticas. Más concretamente se han analizado el modelo de aplicación web (P1-base), el modelo que implementa un servicio web con cliente remoto (P1-ws), y el modelo con EJBs con cliente local (P1-ejb). P1-base y los servicios de P1-ws y de P1-ejb se encuentran en un servidor, los clientes de P1-ws y P1-ejb en otro y la base de datos en otro diferente. A la hora de analizar los resultados sobre los tiempos de respuesta, se obtiene el siguiente ranking:

1. Servicio web.
2. Aplicación web.
3. EJBs.

Esto quiere decir que el servicio web con cliente remoto es el modelo más rápido de los tres analizados, seguido por la aplicación web, y el modelo más lento es aquel que utiliza EJBs con cliente remoto.

PREGUNTA 1. ¿Cree que son coherentes estos resultados? Razone su respuesta.

PREGUNTA 2. En caso de que estos resultados no le parezcan coherentes, ¿cuál cree usted que podría ser el ranking de velocidades? ¿Por qué?

PREGUNTA 3. ¿Cómo cree que se modificarían los resultados si se desplegaran todos los componentes en un único servidor?

1,00/1,00

Esos resultados no son coherentes, ya que se sabe que la aplicación web suele tener un mayor rendimiento en comparación con los web services. Un servicio web permite crear proyectos más sólidos y extensos, a coste de una mayor necesidad computacional y de rendimiento (ofrecen transparencia que afecta a éste).

El ranking de velocidades más adecuado sería Aplicación web, EJB, y servicio web. EJB también reduce mucho su coste si se utiliza de manera local, y no remota, por lo que los resultados se modificarían en cuanto a que la aplicación web y el EJB estarían mucho más igualados.

EJERCICIO 3.- En la práctica 3, con el balanceador correctamente configurado sobre las instancias de un cluster, al revisar el contenido de la tabla pagos de la BD se observa lo siguiente:

- 1. Indicar qué puede haber sucedido, justificando la respuesta.
- 2. Indicar qué herramientas y mandatos usarías durante el proceso anterior para monitorizar el sistema y poder confirmar lo sucedido.

1,00/1,00

El comportamiento del balance de carga refleja lo siguiente: Al comienzo, las peticiones se reparten equitativamente entre ambas instancias, es decir, más o menos la mitad de ellas para cada una. Pasado un rato, todas las peticiones pasan a ser tratadas por la instancia 2, y tras otro periodo, vuelve a aparecer la instancia 1 por lo que se vuelve a balancear la carga equitativamente.

Esto se puede deber a un proceso fail-over, en el cual la instancia 1 ha sido eliminada del sistema, puede que con el comando 'kill'. En ese momento, al balanceador no le queda otra que reenviar todas las peticiones a la instancia 2. Tras ese periodo, ocurre un proceso de fail-back en el que se reinicia la instancia 1 y vuelve a repartirse la carga.

Las herramientas o mandatos que utilizaría para comprobar todo esto son:

- Desde la página del Load Balancer Manager, comprobar que la instancia 1 tiene status ERR, por lo que no está activa.
- Desde la base de datos, comprobar que todas las peticiones efectivamente están siendo tratadas únicamente con la instancia 2.
- Tras ese rato, en el Load Balancer Manager se vería que la instancia 1 vuelve a estar activa, y que reparte equitativamente las peticiones (Elected).

idautorizacion	idcomercio	instancia
37412	2090314	Instance02
37413	8744810	Instance01
37414	7071629	Instance02
37415	311186	Instance01
37416	3546112	Instance02
37417	7458599	Instance01
37418	5051409	Instance02
37419	1410682	Instance01
<líneas con el mismo comportamiento>		
38713	2726048	Instance02
38714	491627	Instance01
38715	6490129	Instance02
38716	6064112	Instance02
38717	8064148	Instance02
38718	1609425	Instance02
38719	9444495	Instance02
38720	465703	Instance02
<líneas con el mismo comportamiento>		
40023	1925440	Instance02
40024	7996146	Instance02
40025	1143788	Instance02
40026	475548	Instance01
40027	9618116	Instance02
40028	8653517	Instance01
40029	5253920	Instance02
40030	4094660	Instance01
40031	7198306	Instance02
40032	2101745	Instance01
<líneas con el mismo comportamiento>		

EJERCICIO 4.- La URL del servicio web usado en la P1A es `http://10.1.1.1/P1-ws-ws/VisaDAOWSService`, y el cliente de dicho servicio web se encuentra ubicado en la máquina virtual con la dirección IP 10.1.1.2. Sin embargo, por cuestiones técnicas, se tiene que cambiar la dirección IP de la máquina virtual donde se encuentra ubicado el servicio web y ésta pasa a ser la dirección 10.1.1.3. Indicar el código java que permitiría que la aplicación cliente siga funcionando correctamente sin necesidad de regenerar (mediante el uso de `wsimport`) los stubs que permiten acceder al servicio web.

1,00/1,00

La generación dinámica de stubs con `BindingProvider` en los módulos java nos permiten acceder de forma dinámica a la dirección del servidor establecida en el `web.xml`.

```
BindingProvider bp = (BindingProvider) dao;
try{
    bp.getRequestContext().put(BindingProvider.ENDPOINT_ADDRESS_PROPERTY,
        getServletContext().getInitParameter("server_address"));
} catch(Exception e){
    enviaError(new Exception("Direccion del servidor no encontrada."), request, response);
    return;
}
```

Con el método `getServletContext().getInitParameter("server_address")`, teniendo en `web.xml` la dirección adecuada declarada de la siguiente forma:

```
<context-param>
    <param-name>server_address</param-name>
    <param-value>http://10.1.1.3:8080/P1-ws-ws/VisaDAOWSService</param-value>
</context-param>
```

EJERCICIO 5.- Cuando se implementan aplicaciones Java, todos los archivos que constituyen dicha aplicación se comprimen y empaquetan en un solo archivo. Cuando se habla de aplicaciones web, dicho archivo contiene tanto servlets, JSPs y EJBs. En una aplicación web de las desarrolladas en la asignatura, se encuentran los siguientes archivos: MANIFEST.MF, PagoBean.class, P1-ejb.war, cabecera.jsp, sun-ejb-jar.xml y application.xml.

Identifica en qué fichero de los diferentes empaquetados vistos en la asignatura se deberían encontrar cada uno, justificando porqué.

0,60/1,00

Los ficheros MANIFEST.MF y sun-ejb-jar.xml irán juntos dentro del archivo P1-ejb.jar que contendría las características de despliegue del ejb. En el fichero de aplicación de empresa con extensión .ear se encontrará el fichero application.xml que es un descriptor de despliegue de la aplicación.

cabecera.jsp se encontrará dentro del p1-ejb.war, que contiene archivos web y servlets.

EJERCICIO 6.- El punto de partida de este ejercicio será el script P2.jmx desarrollado en la práctica P2 para la realización de pruebas de rendimiento utilizando la herramienta JMeter. A partir de la experiencia adquirida en la práctica 2 para configurar un script de pruebas de dicha aplicación, se pide editar con JMeter dicho script y modificarlo para satisfacer las siguientes condiciones:

1. Realización de 225 pagos por parte de cada uno de 20 usuarios sobre la aplicación P1-base.
2. El importe de los pagos realizados será aleatorio y se encontrará entre 150 y 500 euros.
3. El identificador de transacción de la operación para cada pago empezará en 125 y se irá incrementando de 1 en 1.
4. El ID del comercio utilizado al realizar el pago será igual a 173.

Deberá describir cada uno de los pasos que ha de realizar para modificar dicho script.

1,00/1,00

Dentro de JMeter cambiaremos, en la carpeta de P1-Base, cambiaremos los siguientes parámetros en este orden para realizar los cambios numerados del 1 al 4 anteriormente:

- Number of threads (en thread Group) = 20
- Samples (en variables definidas por el usuario) = 225
- Minimum value (Variable aleatoria) = 150
- Maximum value (") = 500
- Starting Value (idTransaccion) = 125
- Increment (idTransaccion) = 1
- idComercio (HTTP Request) = 173

EJERCICIO 7.- En la práctica número 3 se despliega la aplicación P3 en el clúster SI2Cluster que cuenta con dos instancias.

Se prueba a realizar un pago frente al balanceador de carga accediendo a la página pago.html. Tras acceder a esta página el estado del balanceador de carga es:

Load Balancer Manager for 10.1.99.1									
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://10.1.99.1:28080	Instance01		1	0	Ok	1		613	25
http://10.1.99.1:28081	Instance02		1	0	Ok	0		0	0
Apache/2.2.14 (Ubuntu) Server at 10.1.99.1 Port 80									

Se introducen los datos del pago como número de comercio, importe e ID de transacción. Tras esto se obtiene el formulario solicitando los datos del titular de la tarjeta, fecha de caducidad, etc. Y se pulsa en pagar. Sin embargo, se obtiene como resultado el mensaje pago incorrecto. El estado del balanceador de carga tras realizar el pago es el siguiente:

Load Balancer Manager for 10.1.99.1									
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://10.1.99.1:28080	Instance01		1	0	Ok	2		1.5K	532
http://10.1.99.1:28081	Instance02		1	0	Ok	1		748	1.7K
Apache/2.2.14 (Ubuntu) Server at 10.1.99.1 Port 80									

Responda razonadamente a las siguientes cuestiones:

1. Suponiendo que el balanceador de carga ha sido correctamente configurado, la aplicación P3 ha sido correctamente desplegada, que ésta se encuentra libre de errores, que todos los datos introducidos en los formularios son correctos, y que no hay pagos almacenados en la base de datos, justificar los posibles motivos del fallo producido a la hora de realizar el pago en el anterior escenario.
2. Si durante la realización del pago anterior inspeccionáramos las cookies en el navegador, ¿cuál de estos valores sería posible para la cookie JSESSIONID? Razone su respuesta.

a. be1afff1f7615a6a33b70e584d42

b. be1afff1f7615a6a33b70e584d42.Instance01

1,00/1,00

Podemos plantear que el error ha ocurrido ya que no se ha añadido la propiedad de jvmRoute a través de la consola de administración de glassfish, que permite mantener la afinidad de las sesiones en el balanceador de carga. Es por esto, que seguramente se podría ver la cookie JSESSIONID como la opción a.be1afff1f7615a6a33b70e584d42, ya que no tendría la instancia asignada correctamente con el fin de mantener la afinidad. Se solucionaría añadiendo esta propiedad (jvmRoute) y volviendo a ejecutar los pagos.

EJERCICIO 8.- A lo largo de las prácticas, hemos visto un fichero que se le conoce como descriptor de despliegue de un archivo EAR. ¿A qué archivo nos estamos refiriendo? Copia el código aquí de dicho fichero de alguna de tus prácticas, explicando cada línea.

1,00/1,00

El descriptor de despliegue de los archivos ear se llama application.xml. A continuación, se incluye el código de dicho fichero para el archivo P1-ejb.ear de la práctica 1 B:

```
<?xml version="1.0" encoding="UTF-8"?>
<application version="5" xmlns="http://java.sun.com/xml/ns/javaee"
xmlns:xsi="http://www.w3.org/2001/XMLSchemainstance"
xsi:schemaLocation="http://java.sun.com/xml/ns/javaee
http://java.sun.com/xml/ns/javaee/application_5.xsd">
    <display-name>P1-ejb</display-name>
    <module>
        <ejb>P1-ejb.jar</ejb>
    </module>
    <module>
        <web>
            <web-uri>P1-ejb-cliente.war</web-uri>
            <context-root>/P1-ejb-cliente</context-root>
        </web>
    </module>
</application>
```

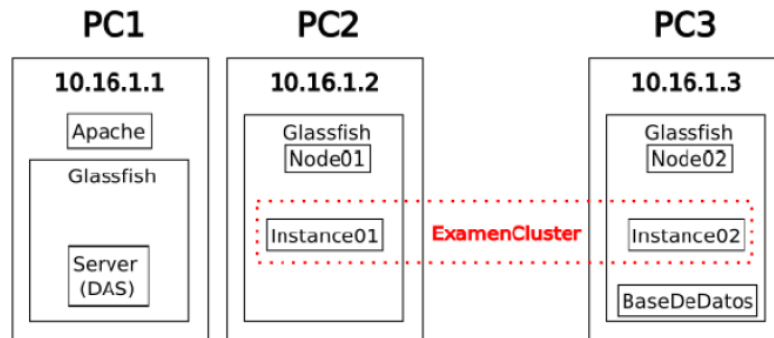
El tag 'display name' incluye el nombre con el que se muestra el archivo de aplicación (p1-ejb), Y luego se pueden ver los distintos módulos incluidos:

El módulo ejb (p1-ejb.jar) que es el módulo EJB servidor que tiene el descriptor EJB y las clases utilizadas.

El módulo web (P1-ejb-cliente.war) que contiene los servlets y los archivos jsp o ficheros de error y html.

También se incluye la ruta de contexto para dicho módulo web, a parte de la url.

EJERCICIO 9.- En la práctica 3, se desea configurar un clúster con nombre “ExamenCluster” de acuerdo al siguiente esquema (se asume usuario si2):



1. Indique los pasos estrictamente necesarios que tendría que seguir para realizar el intercambio de claves entre las máquinas virtuales. Explique por qué es necesario realizar este intercambio de claves.

2. Indique los valores que deben tomar las siguientes variables de configuración de los ficheros build.properties y postgresql.properties. Justifique la respuesta

build.properties

```
...
as.host=
...
as.target=
...
```

postgresql.properties

```
...
db.host=
...
db.client.host=
...
```

0,95/1,00 - sin admon remota

1.

Desde PC1 primero tendremos que generar el par de claves con dsa, con el comando siguiente:

```
ssh-keygen -t dsa
```

Una vez hecho esto, importamos la clave publica en PC2 y PC3 mediante el comando scp, concretamente de esta manera:

```
scp .ssh/id_dsa.pub si2@10.16.1.2:
```

```
scp .ssh/id_dsa.pub si2@10.16.1.3:
```

En PC2 y PC3 se tienen que realizar los siguientes comandos:

```
mkdir -m 700 ~/.ssh
```

```
cat ~/id_dsa.pub >> ~/.ssh/authorized_keys2
```

Después habría que comprobar desde PC1 que se puede acceder mediante ssh a PC2 y PC3 sin que se pida la contraseña.

2.

as.host = 10.16.1.1 --> es el host desde donde se van a hacer las peticiones

as.target = ExamenCluster --> es como llamamos al cluster

db.host = 10.16.1.3 --> es donde se encuentra la base de datos

db.client_host = 10.16.1.1 --> es el host cliente

EJERCICIO 10.- Referente también a la práctica P2, en la obtención de la curva de productividad, se configuran los siguientes parámetros:

Elemento	Parámetro	Valor
JVM Settings	Heap Máx. (MB)	512
JVM Settings	Heap Mín. (MB)	512
HTTP Service	Max.Thread Count	16
HTTP Service	Queue size	60
Web Container	Max.Sessions	-1
Visa Pool	Max.Pool Size	64

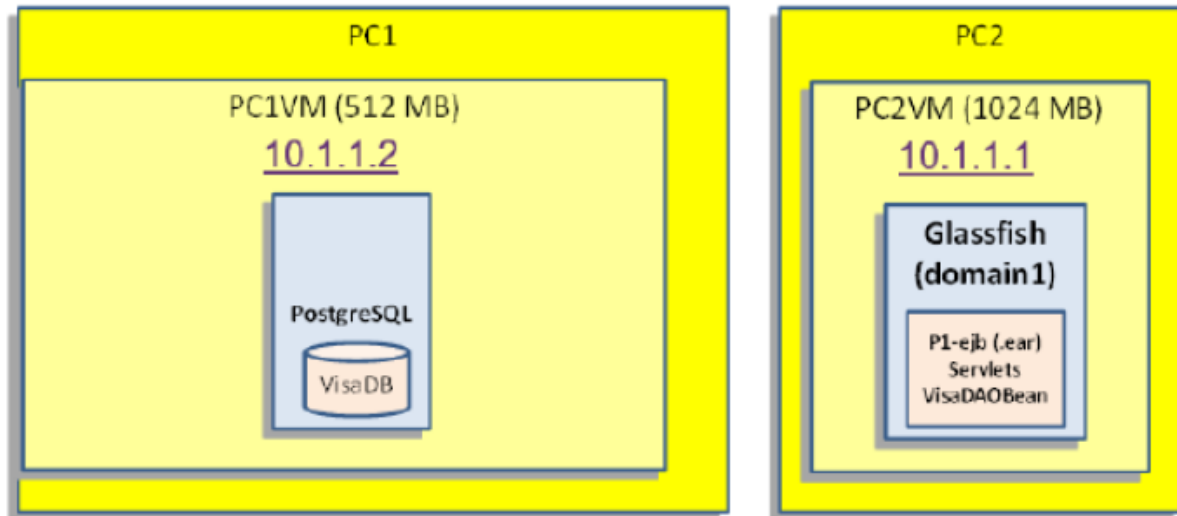
De los siguientes conjuntos de resultados de la monitorización, indique cuáles serían posibles y cuáles no. Razone su respuesta:

- 1. NumJDBCCount=24 numHTTPCount=12 numHTTPQ=24
- 2. NumJDBCCount=12 numHTTPCount=24 numHTTPQ=24
- 3. NumJDBCCount=12 numHTTPCount=12 numHTTPQ=6
- 4. NumJDBCCount=12 numHTTPCount=12 numHTTPQ=70
- 5. NumJDBCCount=12 numHTTPCount=12 numHTTPQ=40

0,70/1,00 – Explicación del 1 mal, falta explicar 2

- 1. No puede ser porque NumJDBCCount excede los 16 max thread count..
- 2. No puede ser.
- 3. Sí puede ser.
- 4. No se puede porque numHTTPQ excede el queue size que es 60.
- 5. Sí puede ser.

EJERCICIO 11.- Imaginad que tenemos que preparar esta configuración para el despliegue de la aplicación con EJB local:



Indica qué ficheros de propiedades tienes que modificar y justifica el porqué de esos valores.

0,80/1,00 – as.server/cli ? mal expl dbcli

Hay que modificar los ficheros postgresql.properties y build.properties.

En postgresql.properties hay que cambiar:

db.host = 10.1.1.2 -> la base de datos está en PC1 que tiene la IP 10.1.1.2

db.client.host = 10.1.1.1 -> el cliente se encuentra en el PC2 con IP 10.1.1.1

En build.properties hay que cambiar:

as.host = 10.1.1.1 -> el host es el PC2 que tiene la IP 10.1.1.1

EJERCICIO 12.- Basándonos en el código de la aplicación P1-base, se nos piden las siguientes modificaciones/requisitos:

- La aplicación a desplegar se llamará P1-base-examen.
- Incluir un nuevo campo a rellenar por el usuario, donde deberá elegir si el pago se hará a crédito (valor 'C') o a débito (valor 'D').

Indica qué ficheros deberíamos modificar para recoger estas dos modificaciones que se nos piden, justificando la respuesta.

0,25/1,00 –

- El nuevo campo que nos piden debemos incluirlo en el JSP formdatosvisa.jsp.
 - Modificar el objeto PagoBean para que recoja dicho campo, además de incluir los getters/setters correspondientes.
 - Modificar el servlet ProcesaPago para que genere el objeto pago con este nuevo campo.
 - Modificar la BBDD para incluir el nuevo campo en la tabla PAGO.
 - Modificar el método realizaPago de VisaDAO.java para que el pago se realice como haya elegido el usuario (crédito o débito).
-
- Para cambiar el nombre de la aplicación se accede al build.properties de P1-base. Se cambia el atributo nombre.
 - Y para añadir el nuevo campo, modificaríamos en VisaDao.java

0,40/1,00 – sin sql, sin dao, sin jsp

- En el build.properties, la variable nombre pasará a llamarse P1-base-examen
- Para incluir el nuevo campo debe cambiarse el archivo ComienzaPago.java y el archivo PagoBean para añadir un nuevo atributo a la clase que sea el tipo de pago

1,00/1,00

1. testdb.jsp

Añadimos la opción:

```
<tr>
    <td>Pago a crédito (valor 'C') o a débito (valor 'D')</td>
    <td><input type= "radio" name= "C" value = "true"/>True<input type= "radio" name= "D" value =
    "false"/>False</td>
</tr>
```

2. PagoBean

```
private Boolean credito
getCredito()
setCredito(Boolean credito)
```

3. ComienzaPago

En el método en el que se crea el pago:
Pago.setCredito(request.getParameter(credito))

4. ProcesaPago.java

En el método en el que se crea el pago:
Pago.setCredito(request.getParameter(credito))

5. VisaDao.java

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

String getQryInsertPago(PagoBean pago){
    String qry = "insert into pago("
    + "idTransaccion, importe, idComercio, numeroTarjeta, credito)" +
    + "values("
    + "" + pago.getIdTransaccion() + ","
    + pago.getImporte() + ","
    + "" + pago.getIdComercio() + ","
    + "" + pago.getTarjeta().getNumero() + ","
    + "" + pago.getCredito() + ")";

    return qry
}
```

EJERCICIO 13.- El punto de partida de este ejercicio será el script P2.jmx desarrollado en la práctica P2 para la realización de pruebas de rendimiento utilizando la herramienta JMeter. A partir de la experiencia adquirida en la práctica 2 para configurar un script de pruebas de dicha aplicación, se pide editar con JMeter dicho script y modificarlo para satisfacer las siguientes condiciones:

1. Realización de 225 pagos por parte de cada uno de 20 usuarios sobre la aplicación P1-base.
2. El importe de los pagos realizados será aleatorio y se encontrará entre 150 y 500 euros.
3. El identificador de transacción de la operación para cada pago empezará en 125 y se irá incrementando de 1 en 1.
4. El ID del comercio utilizado al realizar el pago será igual a 173.

Deberá describir cada uno de los pasos que ha de realizar para modificar dicho script.

1,00/1,00

Cambiaríamos el valor de Samples a 225. Number of threads = 20 para crear 20 usuarios.

Para el importe, en la pestaña random Variable, ponemos en el apartado configure the Random generator, minimum value y maximum value 150 y 500 respectivamente.

En idTransaccion En el campo Starting value 125 y en increment 1.

En http request, al introducir los parámetros, indicaremos 173 en el campo idComercio.

EJERCICIO 14.-

Pregunta 1. Se van a realizar una serie de pruebas del rendimiento de la aplicación de pagos desarrollada, utilizando la herramienta JMeter. En concreto, se han configurado 3 pruebas, en 3 grupos de hilos distintos, con las siguientes configuraciones

1. Número de hilos = 20 Periodo de subida = 10 Contador del bucle = 5
2. Número de hilos = 50 Periodo de subida = 15 Contador del bucle = 10
3. Número de hilos = 5 Periodo de subida = 3 Contador del bucle = 1000

¿Cuántos pagos se esperan ver en la base de datos después de la ejecución de cada prueba?

Pregunta 2. Para obtener los resultados de rendimiento de la prueba, agregamos un elemento de tipo "Informe agregado" al plan de pruebas en JMeter. Ejecutamos la prueba, y dentro de este informe, la columna "Error" marca un valor de 0,00% para todas las pruebas realizadas. Sin embargo, al revisar la base de datos, el número de pagos es menor del que se esperaba. Indique qué podríamos hacer, dentro de la herramienta JMeter, para investigar las causas de este problema

¿?/1,00 – Aunque no se especifica nada de que haya IDs repetidos, podría ser válida tu respuesta. Por tanto, te he puesto correcta la pregunta

- 1- 5000. Los 100 primeros de la prueba 1, los 400 siguientes de la prueba 2 (los 100 primeros darán error por tener id's repetidas) y los 4500 siguientes de la prueba 3 (los 500 primeros darán error por tener id's repetidas)
- 2- Agregar otro elemento de tipo "Árbol de Resultados". Ahí podemos ver qué se ha enviado de petición y que se ha recibido de vuelta. Si ha ocurrido ese fallo muy probablemente haya habido un problema en la creación del pago, pero la petición se ha enviado y se ha recibido una respuesta con código 200. Por eso no se detecta el fallo en el "Informe Agregado"

1,00/1,00

- 1- 560
- 2- Podemos agregar el visualizador de árbol, y poniendo el modo debug a true

EJERCICIO 15.- Suponga que la aplicación con la que está trabajando, parecida a la de la práctica 3, tiene por URL `http://<server>:<port>/P3-exa` y está desplegada en un clúster cuyo nombre es “ClusterExamen”, que está formado por tres instancias: `inst1(10.1.1.3, puerto 28080)`, `inst2(10.1.1.4, puerto 28080)` e `inst3(10.1.1.3, puerto 28081)`

1. Indique los pasos que tendría que seguir para configurar el módulo de balanceo de Apache para que funcionase en modo proxy inverso con afinidad de sesión. Escriba también el contenido correspondiente al fichero `proxy_balancer.conf`
2. Explique en detalle por qué es necesario establecer la propiedad `jvmRoute` y relaciónela con el fichero anterior

0,50/1

Tienes mal puesto el nombre de las instancias en el fichero de configuración

Por otro lado, la respuesta que das a la pregunta 2 no la relacionas con el archivo de configuración

1. Crear el fichero `proxy_balancer.conf` en modo root
2. Escribir en él:

```
ProxyRequest Off
<Proxy balancer: //ClusterExamen>
    BalancerMember http://10.1.1.3:28080 route=Intsance01
    BalancerMember http://10.1.1.4:28080 route=Intsance02
    BalancerMember http://10.1.1.3:28081 route=Intsance03
</Proxy>
<Location /P3-exa>
    Order allow,deny
    Allow from all
    ProxyPass balancer://ClusterExamen/P3-exa stickysession=JSESSION|jsessionid scolonpathdelim=on
    ProxyPassReverse balancer://ClusterExamen/P3-exa
<Location /balancer-manager>
    SetHandler balancer-manager
</Location>
```
3. Activamos la nueva configuración en el servidor de Apache

```
sudo ln -sf ../mods-available/proxy.load
sudo ln -sf ../mods-available/proxy_htable.load
sudo ln -sf ../mods-available/proxy_balancer.load
```
4. Volvemos a arrancar el servidor Apache

```
Sudo service apache2 restart
```

Es necesario porque esta propiedad permite que cada instancia creada del cluster modifique la generación del identificador de sesión, incluyendo en él el nombre de la instancia que ha procesado la petición. Esto permite mantener la afinidad de sesión en el balanceador de carga.

EJERCICIO 15.- CORRECCIÓN

2. Escribir en él

```
ProxyRequests Off
<Proxy balancer://S12Cluster>
    BalancerMember http://10.1.1.3:28080 route=inst1
    BalancerMember http:// 10.1.1.4:28080 route=inst2
    BalancerMember http:// 10.1.1.3:28081 route=inst3

</Proxy>
<Location /P3>
    Order allow,deny
    Allow from all
    ProxyPass balancer://ClusterExamen/P3-exa stickysession=JSESSIONID|jsessionid
    scolonpathdelim=On
    ProxyPassReverse balancer:// ClusterExamen /P3-exa
</Location>
<Location /balancer-manager>
    SetHandler balancer-manager
</Location>
```

3. Activamos la nueva configuración en el servidor de Apache

```
sudo ln -sf ../mods-available/proxy_balancer.conf
sudo ln -sf ../mods-available/proxy.load
sudo ln -sf ../mods-available/proxy_http.load
sudo ln -sf ../mods-available/proxy_balancer.load
```

EJERCICIO 16.- Se ha codificado un Message Driven Bean de la plataforma J2EE llamado VisaDiferidoJMSBean. Se desea que dicho Bean utilice para la instanciación de conexiones a destinos JMS una “connection factory” identificada por el nombre “jms/DiferidoConnectionFactory” que ha sido creada previamente en el servidor Glassfish. Indicar qué fichero será necesario editar y la información que será necesaria colocar en él para conseguir que ocurra lo indicado al desplegar una aplicación web que contiene dicho Bean.

1,00/1,00

Editar sun-ejb-jar:

```
<ejb>
  <ejb-name> VisaDiferidoJMSBean </ejb-name>
  <mdb-connection-factory>
    <jndi-name>
      jms/DiferidoConnectionFactory
    </jndi-name>
  </mdb-connection-factory>
</ejb>
```