

SI2 Assignment II

Lucía Gil Maroto
Daniel Varela Sánchez
Guillermo Martín-Coello Juárez

Ejercicio I

Siguiendo todos los pasos anteriores, defina el plan completo de pruebas para realizar las tres ejecuciones secuenciales sobre los tres proyectos definidos hasta ahora (P1-base, P1-ws, P1-ejb). Adjunte el fichero generado P2.jmx al entregable de la práctica.

Importante: Para comprobar el correcto funcionamiento de la simulación y detectar posibles fallos, se recomienda añadir también al elemento P2 Test un “árbol de resultados” (View Results Tree). Para ello, sobre el plan de pruebas, botón derecho, Add Listener View Results Tree. Una vez se tenga la certeza de que la simulación funciona correctamente se desactivará el “árbol de resultados” (pulsando encima con el botón derecho del ratón) y se realizará de nuevo la simulación. El árbol de resultados permite inspeccionar los datos enviados en cada petición HTTP y la respuesta obtenida del servidor, que deberán ser correctas. Por ejemplo, no deberá aparecer ningún pago incorrecto en las respuestas.

A continuación definimos el plan completo de pruebas para P1-base:

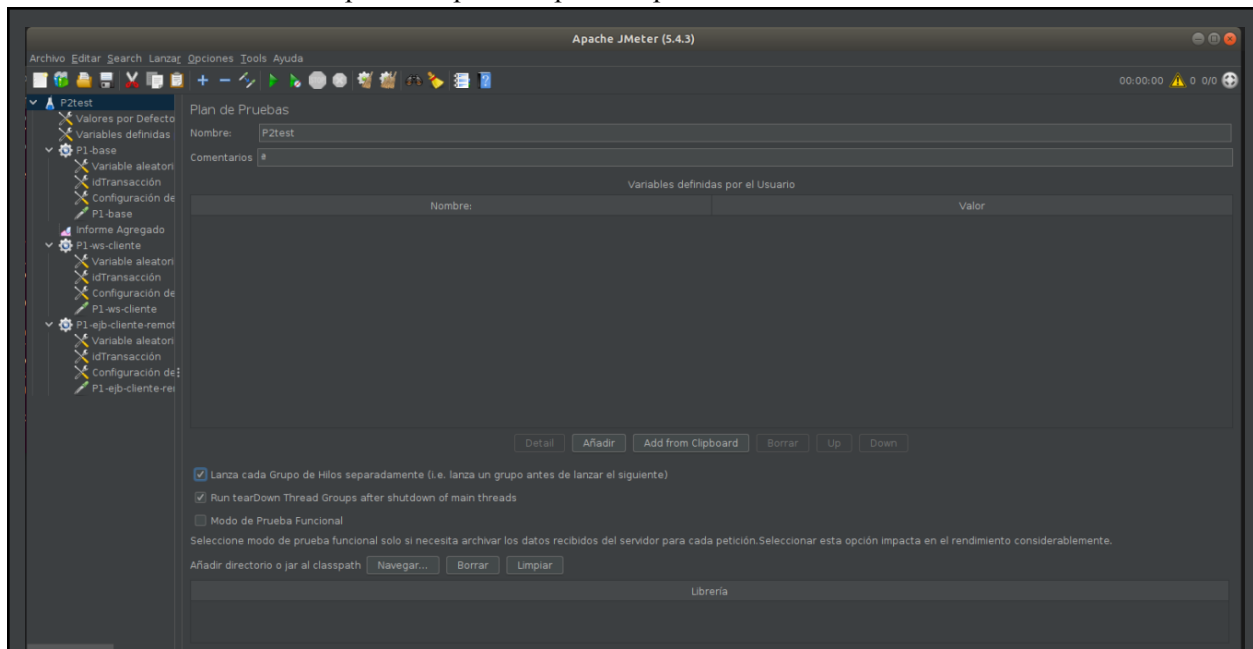


Imagen I: Plan de pruebas

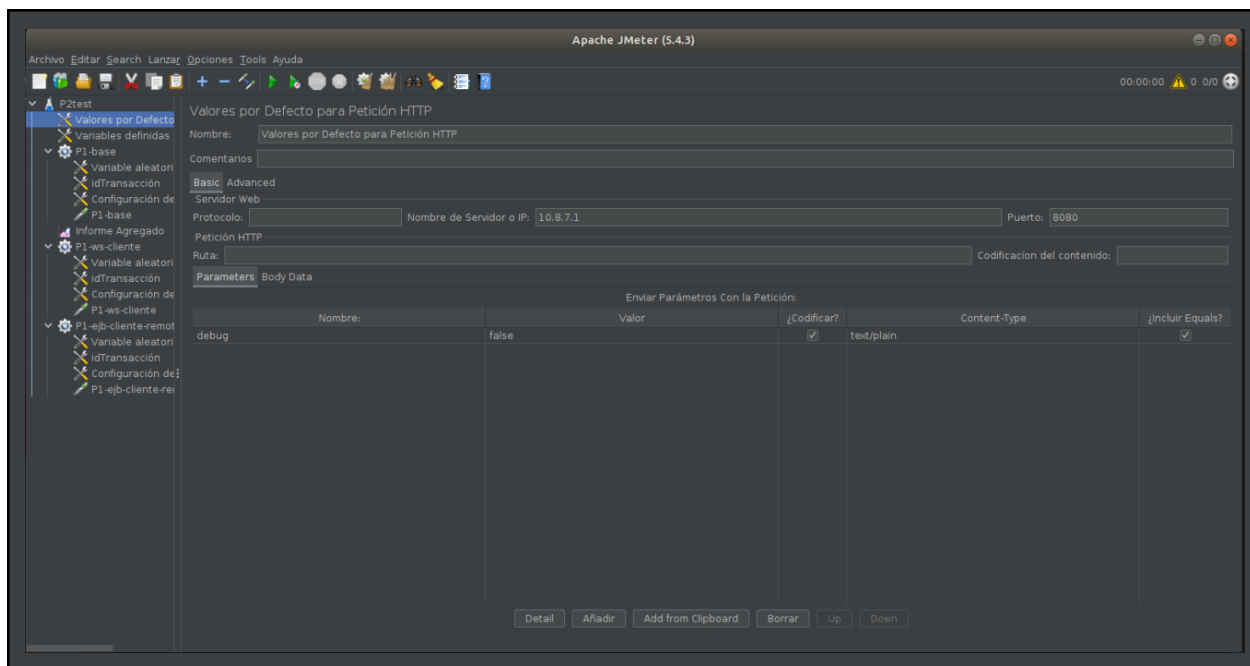


Imagen II: Valores por defecto HTTP

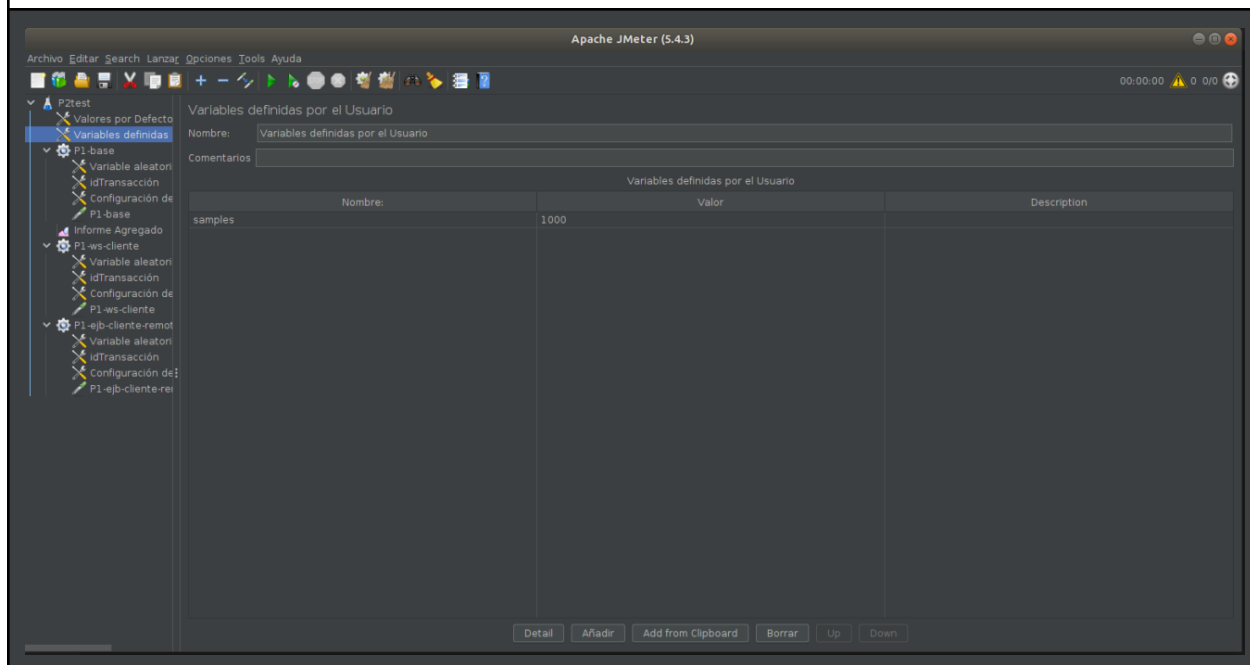


Imagen III: Variables definidas por el usuario

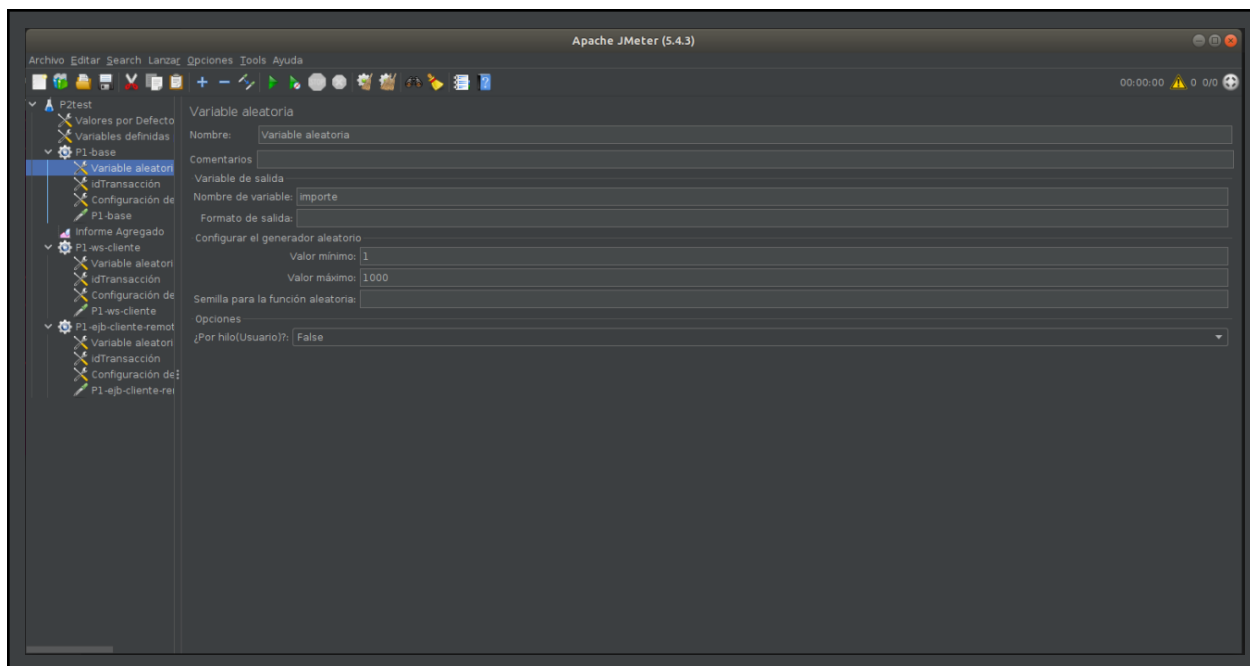


Imagen IV: Variable aleatoria

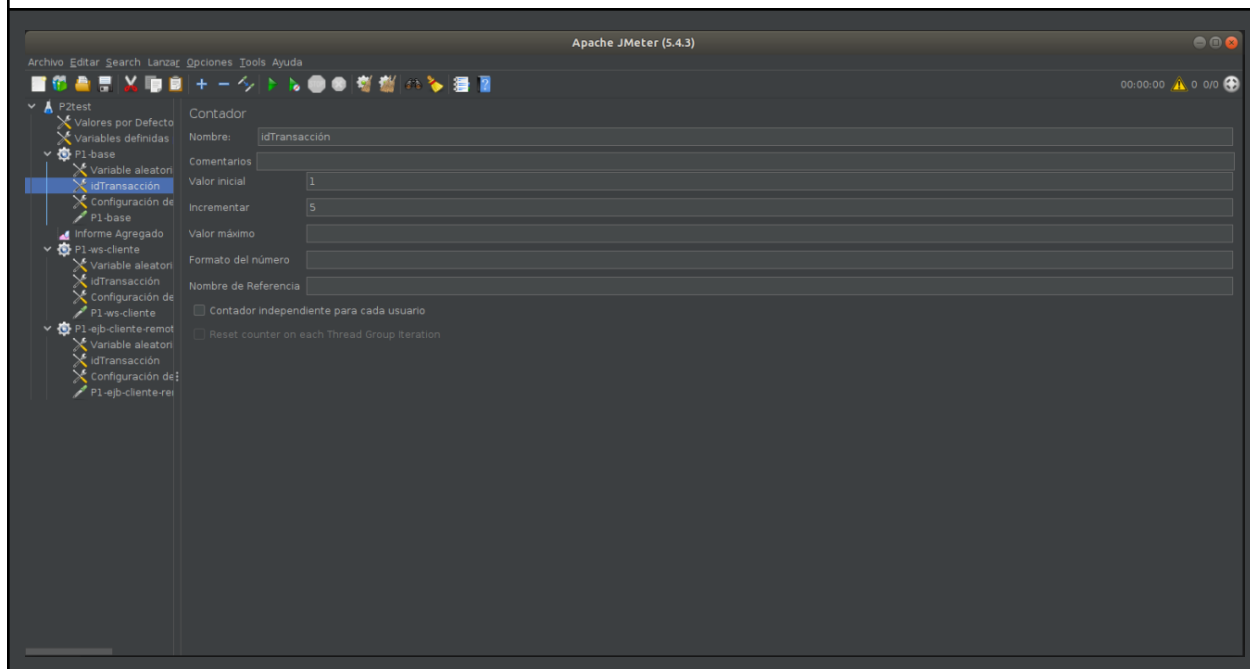


Imagen V: Plan de pruebas

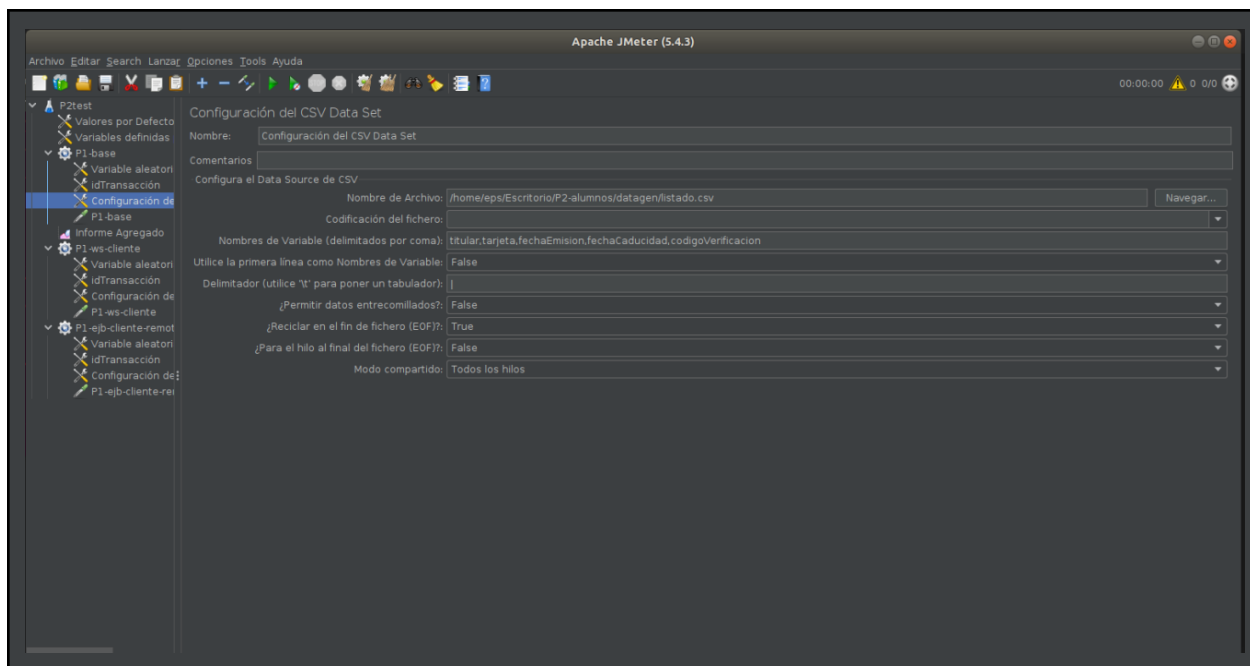


Imagen VI: Configuración del CSV Data Set

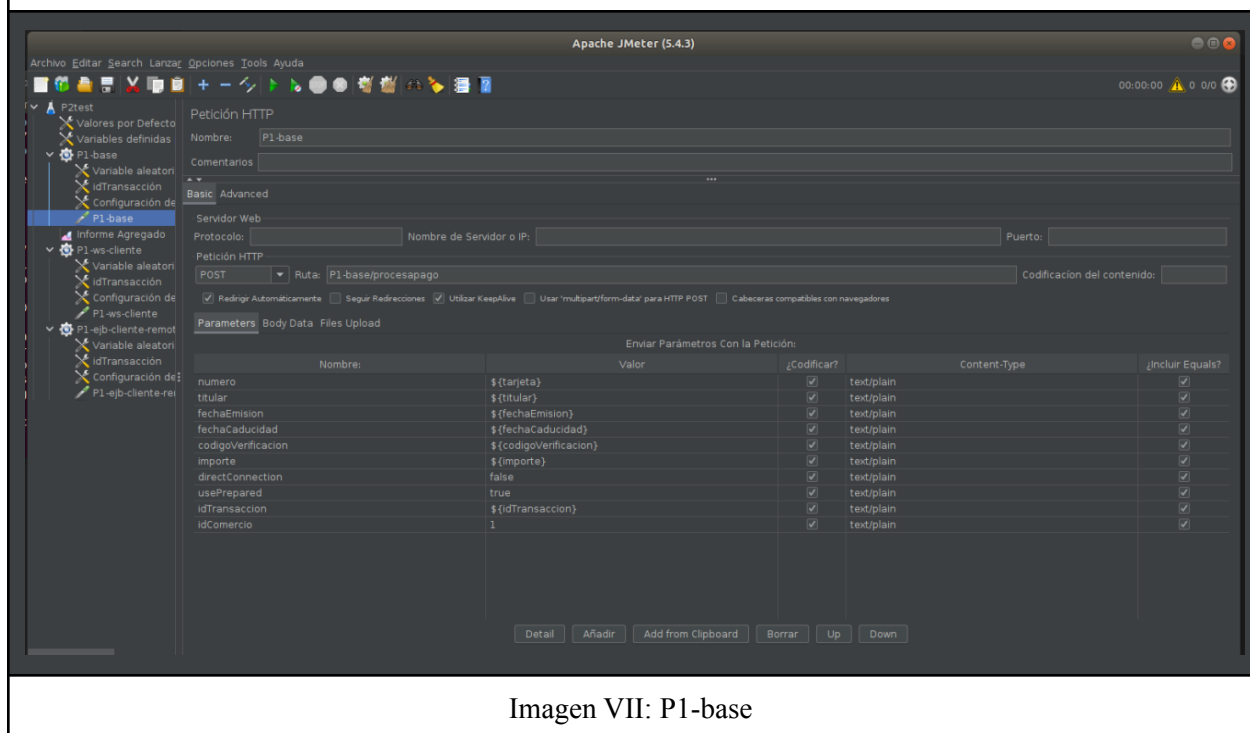


Imagen VII: P1-base

Estos pasos se repiten de manera similar para P1-ws-cliente y P1-ejb-cliente-remoto.

Finalmente, comprobamos y ejecutamos el árbol de resultados:

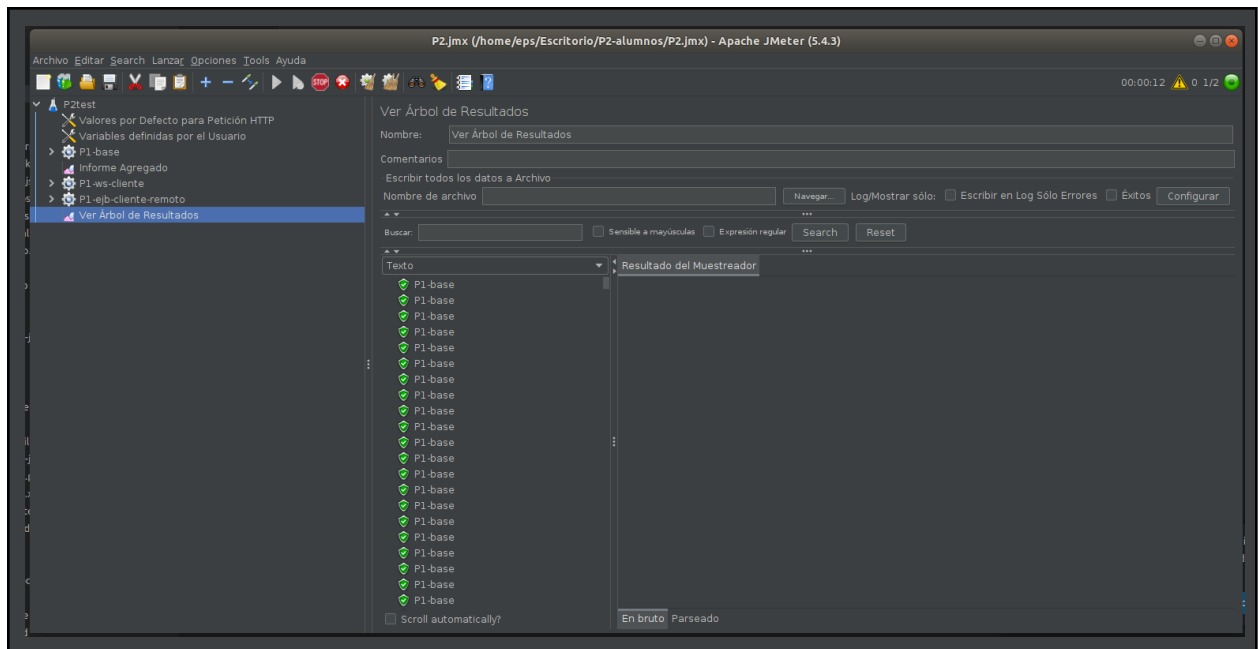


Imagen VIII: Árbol resultados

Ejercicio II

Preparar el PC con el esquema descrito en la Figura 22. Para ello:

Anote en la memoria de prácticas las direcciones IP asignadas a las máquinas virtuales y al PC

Detenga el servidor de GlassFish del PC host.

Inicie los servidores GlassFish en las máquinas virtuales.

Repliegue todas las aplicaciones o pruebas anteriores (P1-base, P1-ws, etc), para limpiar posibles versiones incorrectas.

Revise y modifique si es necesario los ficheros build.properties (propiedad “nombre”) de cada versión, de modo que todas las versiones tengan como URL de despliegue las anteriormente indicadas.

Revise y modifique si es necesario el fichero glassfish-web.xml, para indicar la IP del EJB remoto que usa P1-ejb-cliente.

Despliegue las siguientes prácticas: P1-base, P1-ws, P1-ejb-servidor-remoto y P1-jeb-cliente-remoto, con el siguiente esquema:

- El destino de despliegue de la aplicación P1-base será PC2VM con IP 10.X.Y.2 (as.host)
- El destino del despliegue de la parte cliente de P1-ws y de P1-ejb-cliente-remoto será PC2VM con IP 10.X.Y.2 (as.host.client de P1-ws y as.host de P1-ejb-cliente-remoto)
- El destino del despliegue de la parte servidor de P1-ws y de P1-ejb-servidor-remoto será PC1VM con IP 10.X.Y.1 (as.host.server de P1-ws y as.host.server y as.host.client de P1-ejb-servidor-remoto)
- La base de datos en todos ellos será la de PC1VM con IP 10.X.Y.1 (db.host)

Tras detener/iniciar todos los elementos indicados, anotar la salida del comando “free” así como un pantallazo del comando “nmon” (pulsaremos la tecla “m” para obtener el estado de la RAM) tanto en las máquinas virtuales como en el PC host. Anote sus comentarios en la memoria.

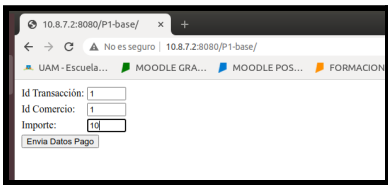
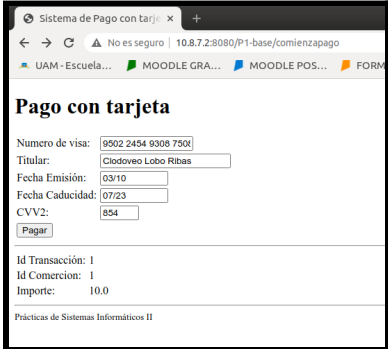

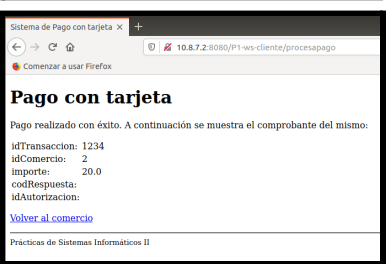
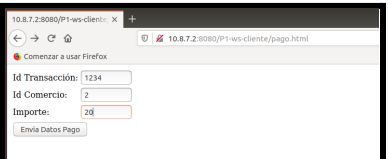
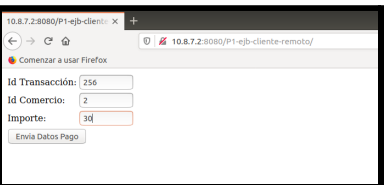
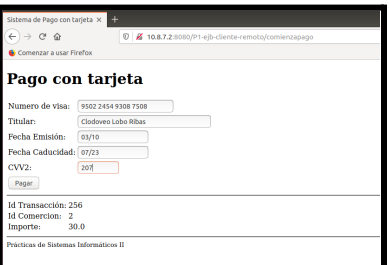
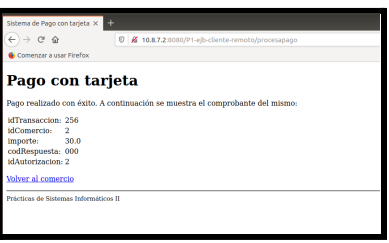
Pruebe a ejecutar un pago “de calentamiento” por cada uno de los métodos anteriores y verifique que funciona a través de la página testbd.jsp.

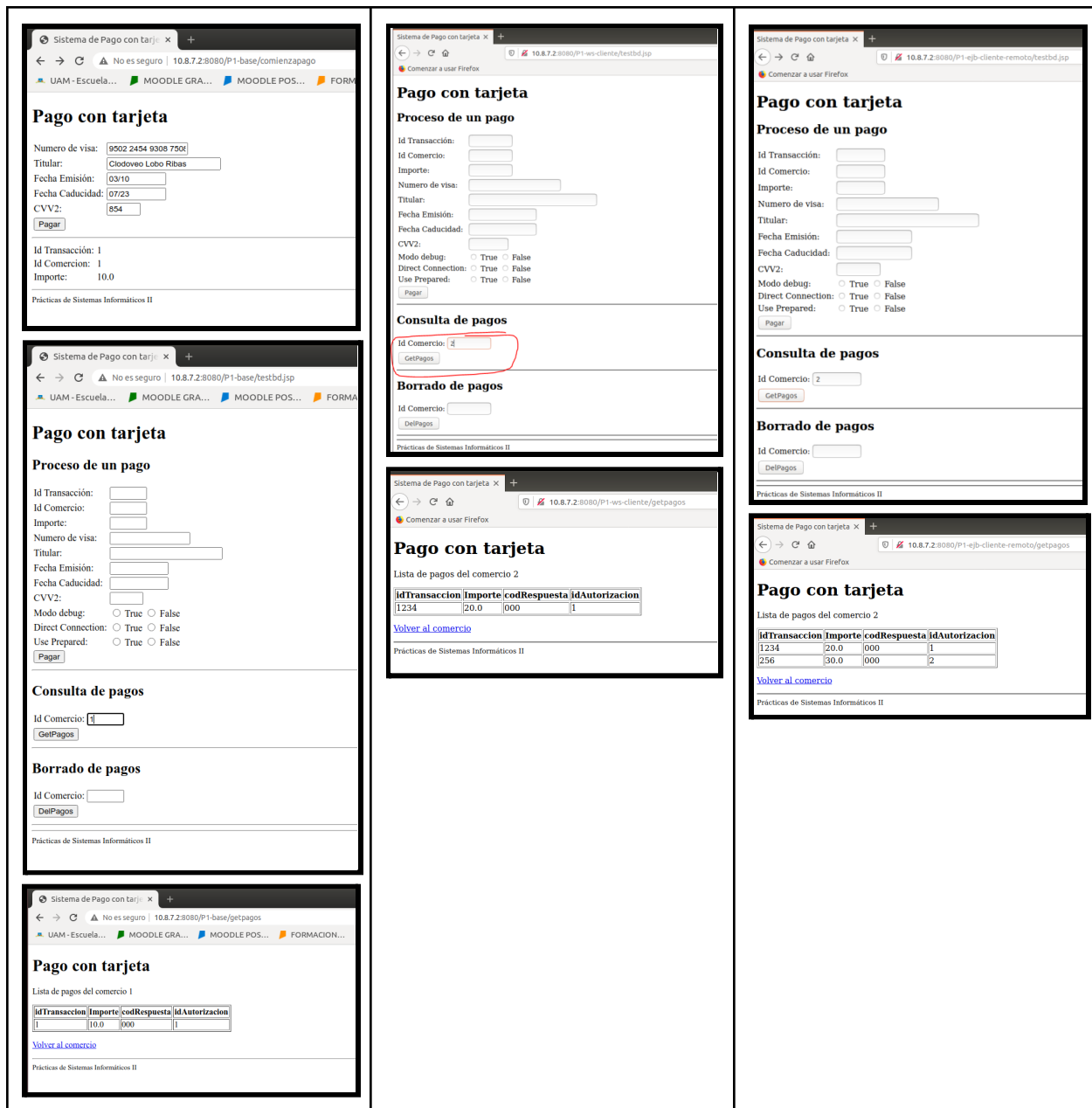
Máquina	IP
PC host	10.10.108.134
PC1VM (Servidor)	10.8.7.1
PC2VM (Cliente)	10.8.7.2

Ejecutamos el comando free para ver el estado de la memoria y el swap y el comando nmon m para comprobar el estado de la RAM.

Máquina	Free	Nmon
PC host	<pre> eps@labvirtips:~/Escritorio/P2-alumnos\$ free total used libre compartido búfer/caché disponible Memoria: 8124376 2580648 410144 1695080 5133584 3548804 Swap: 7999484 7948 7991536 </pre>	<pre> nmon-10f [H for help]--Hostname=labvirtips--Refresh= 2secs --01:13:01 Memory and Swap Total (MB) 7934.8 7812.0 7812.0 7812.0 Free (MB) 7804.4 7804.4 7804.4 7804.4 Free Percent 9.8% 99.9% Linux Kernel Internal Memory (MB) Caché= 4327.4 Active= 3887.0 Buffers= 491.8 Swapped= 0.0 Inactive= 2892.1 Dirty = 0.4 Writeback = 0.0 Hopped = 2896.2 Slab = 293.1 Commit_AS = 10099.3 PageTables= 63.4 </pre>
PC1VM	<pre> sl2@sl2srv01:~\$ free total used free shared buffers cached Mem: 767168 544560 222608 0 37248 190672 -/+ buffers/cache: 316640 450528 Swap: 153592 0 153592 </pre>	<pre> nmon-12f [H for help]--Hostname=sl2srv02--Refresh= 2secs --01:15:03 Memory Stats RAM High Low Swap Total MB 749.2 0.0 749.2 150.0 Free MB 312.7 0.0 312.7 150.0 Free Percent 41.7% 0.0% 41.7% 100.0% MB MB MB MB Caché= 250.7 Active= 201.4 Buffers= 16.6 Swapped= 0.0 Inactive= 211.6 Dirty = 0.0 Writeback = 0.0 Hopped = 22.5 Slab = 14.7 Commit_AS = 785.3 PageTables= 1.3 </pre>
PC2VM	<pre> sl2@sl2srv02:~\$ free total used free shared buffers cached Mem: 767168 445124 322044 0 17004 256572 -/+ buffers/cache: 171548 595620 Swap: 153592 0 153592 </pre>	<pre> nmon-12f [H for help]--Hostname=sl2srv02--Refresh= 2secs --01:15:03 Memory Stats RAM High Low Swap Total MB 749.2 0.0 749.2 150.0 Free MB 312.7 0.0 312.7 150.0 Free Percent 41.7% 0.0% 41.7% 100.0% MB MB MB MB Caché= 250.7 Active= 201.4 Buffers= 16.6 Swapped= 0.0 Inactive= 211.6 Dirty = 0.0 Writeback = 0.0 Hopped = 22.5 Slab = 14.7 Commit_AS = 785.3 PageTables= 1.3 </pre>

Tras esto ejecutamos un pago de calentamiento de cada uno de los métodos para verificar su funcionamiento.

P1-base	P1-ws	P1-ejb
 	  	  



Ejercicio III

Ejecute el plan completo de pruebas sobre las 3 versiones de la práctica, empleando el esquema de despliegue descrito anteriormente. Realice la prueba tantas veces como necesite para eliminar ruido relacionado con procesos periódicos del sistema operativo, lentitud de la red u otros elementos.

Compruebe que efectivamente se han realizado todos los pagos. Es decir, la siguiente consulta deberá devolver “3000”:

```
SELECT COUNT(*) FROM PAGO;
```

Compruebe que ninguna de las peticiones ha producido un error. Para ello revise que la columna %Error indique 0% en todos los casos. Anote los resultados del informe agregado en la memoria de la práctica. Salve el fichero server.log que se encuentra en la ruta glassfish/domains/domain1/logs de Glassfish y adjúntelo con la práctica.

Ejecutamos y comprobamos que no hayan errores:

Informe Agregado

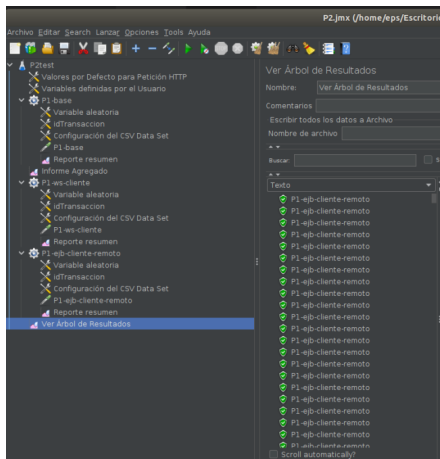
Nombre: Informe Agregado

Comentarios

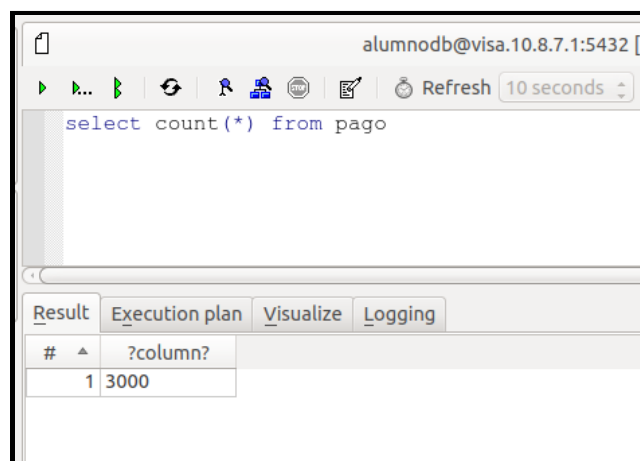
Escribir todos los datos a Archivo

Nombre de archivo Navegar... Log/Mostrar sólo: ☐ Escribir en Log Sólo Errores ☐ Éxitos

Etiqueta	# Muestras	Media	Mediana	90% Line	95% Line	99% Line	Mín	Máx	% Error	Rendimie...	kb/sec	Sent KB/sec
P1-base	1000	10	10	14	16	22	6	53	0,00%	94,4/sec	121,26	0,00
P1-ws-clie...	1000	72	69	89	96	119	52	607	0,00%	13,8/sec	16,71	0,00
P1-eb-cli...	1000	19	18	26	30	42	11	74	0,00%	50,7/sec	66,53	0,00
Total	3000	34	18	75	84	104	6	607	0,00%	29,1/sec	37,00	0,00



Comprobamos que se han producido todos los 3000 pagos:



¿Cuál de los resultados le parece el mejor?

Segun la fotografía adjuntada al principio de este apartado podemos decir que el mejor en este caso sería P1-base

¿Por qué?

Porque obtiene un mayor rendimiento tardando así menos tiempo de media en cada uno de los pagos.

¿Qué columna o columnas elegiría para decidir este resultado?

Las columnas más importantes son las de rendimiento y media como se indica previamente.

Repita la prueba de P1-ejb (inhabilite los „Thread Group“ P1-base y P1-ws) con el EJB local incluido en P1-

ejb-servidor-remoto. Para ello, cambie su „HTTP Request“, estableciendo su „Server Name or IP“ a 10.X.Y.1

(VM1) y su „Path“ a „P1-ejb-cliente/procesapago“. Compare los resultados obtenidos con los anteriores.

El fichero P2.jmx entregado no debe contener estos cambios, es decir, debe estar configurado para probar el EJB remoto.

Etiqueta	# Muestras	Media	Mediana	90% Line	95% Line	99% Line	Mín	Máx	% Error	Rendimie...	Kb/sec	Sent KB/sec
P1-ejb-cli...	1000	18	18	25	28	42	10	88	0,00%	52,4/sec	40,30	0,00
Total	1000	18	18	25	28	42	10	88	0,00%	52,4/sec	40,30	0,00

Podemos observar que en este caso los resultados son peores que los anteriores ya que en este caso tiene que acceder al cliente remoto lo cual consume un tiempo que no necesita consumir cuando el cliente es local.

Ejercicio IV

Adaptar la configuración del servidor de aplicaciones a los valores indicados. Guardar, como referencia, la configuración resultante, contenida en el archivo de configuración localizado en la máquina virtual en `$opt/glassfish4/glassfish/domains/domain1/config/domain.xml`.

Para obtener la versión correcta de este archivo es necesario detener el servidor de aplicaciones. Incluir este fichero en el entregable de la práctica. Se puede copiar al PC con scp.

Configuramos las JVM options:

General

Path Settings

JVM Options

Profiler

JVM Options

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

Configuration Name: server-config

Options (32)

☒ ☐ | Add JVM Option Delete

Select	Value
<input type="checkbox"/>	-Xms512m
<input type="checkbox"/>	-XX:MaxPermSize=192m
<input type="checkbox"/>	-server
<input type="checkbox"/>	-Djava.awt.headless=true
<input type="checkbox"/>	-Djdk.corba.allowOutputStreamSubclass=true
<input type="checkbox"/>	-Djavax.xml.accessExternalSchema=all
<input type="checkbox"/>	-Djavax.management.builder.initial=com.sun.enterprise.v3.admin.AppServerMBeanServerBuilder
<input type="checkbox"/>	-XX:+UnlockDiagnosticVMOptions
<input type="checkbox"/>	-Djava.endorsed.dirs=\${com.sun.aas.installRoot}/modules/endorsed\${path.separator}\${com.sun.aas.installRoot}/lib/
<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"/>	-Xmx512m
<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}
<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.AppserverC
<input type="checkbox"/>	-Dorg.glassfish.additionalOSGiBundlesToStart=org.apache.felix.shell,org.apache.felix.gogo.runtime,org.apache.felix

Configuramos la configuración de aplicación quitando el “reload” y “auto deploy”

Applications Configuration

Enable reloading so that changes to deployed applications are detected and the modified classes reloaded. Also enable and configure automatic deployment of applications. Click Add Property to specify additional settings.

[Load Defaults](#)

Reload: ☐ Enabled
Enables dynamic reloading of applications.

Reload Poll Interval: Seconds
Frequency for checking reload requests.

Admin Session Timeout: Minutes
A value of 0 means the session never times out.

Auto Deploy Settings

Auto Deploy: ☐ Enabled
Automatically deploys applications in the autodeploy directory.

Auto Deploy Poll Interval: Seconds
Frequency at which the autodeploy directory is checked for applications; interval does not affect amount of time to load the application or module.

Auto Deploy Retry Timeout: Seconds
Time to report failure after a file remains stable in size but cannot be opened.

Auto Deploy Directory:
Directory to monitor for autodeploy applications.

XML Validation:
Type of deployment descriptor validation.

Verifier: ☐ Enabled
Performs detailed verification before deployment.

Precompile: ☒ Enabled
Precompiles JSPs, deploys only resulting class files.

Additional Properties (0)

[Add Property](#) [Delete Properties](#)

Select	Name	Value	Description
--------	------	-------	-------------

Revisar el script si2-monitor.sh e indicar los mandatos asadmin que debemos ejecutar en el PC host para averiguar los valores siguientes, mencionados en el Apéndice 1, del servidor PC1VM1:

1. Max Queue Size del Servicio HTTP

```
eps@labvirteps:~/Escritorio/P2$ asadmin --host 10.8.7.2 --user admin --passwordfile ./passwordfile get configs.config.server-config.thread-pools.thread-pool.http-thread-pool.max-queue-size
configs.config.server-config.thread-pools.thread-pool.http-thread-pool.max-queue-size=4096
Command get executed successfully.
eps@labvirteps:~/Escritorio/P2$
```

Podemos ver que la max queue size del servicio HTTP es de 4096.

2. Maximum Pool Size del Pool de conexiones a nuestra DB

```
eps@labvirteps:~/Escritorio/P2$ asadmin --host 10.8.7.2 --user admin --passwordfile ./passwordfile get resources.jdbc-connection-pool.VisaPool.max-pool-size
resources.jdbc-connection-pool.VisaPool.max-pool-size=32
Command get executed successfully.
eps@labvirteps:~/Escritorio/P2$
```

La Max pool size del pool de conexiones a nuestra BD es de 32.

Así como el mandato para monitorizar el número de errores en las peticiones al servidor web.

```
eps@labvirteps:~/Escritorio/P2$ asadmin --host 10.8.7.2 --user admin --passwordfile ./passwordfile monitor --type httpListener
ec mt pt rc
0 8545 29.00 1011
0 8545 29.00 1011
0 8545 29.00 1011
```

El número de errores en las peticiones al servidor web es de 0.

Ejercicio V

Registrar en la hoja de cálculo de resultados los valores de configuración que tienen estos parámetros.

Parámetro	Valor
Heap Máximo	Xmx512m
Heap Mínimo	Xms512m

Thread Count Máximo	5
Tamaño Cola	4096
Máximo de Sesiones	-1
Tamaño de pool máximo	32

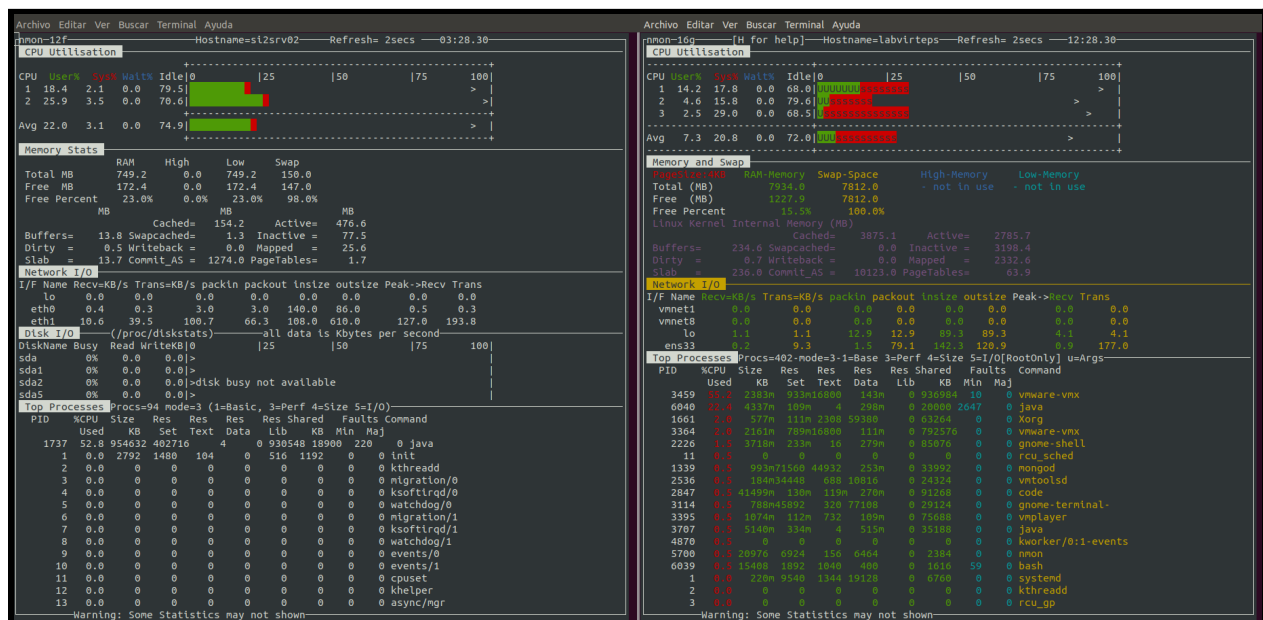
Ejercicio VI

Tras habilitar la monitorización en el servidor, repita la ejecución del plan de pruebas anterior. Durante la prueba, vigile cada uno de los elementos de monitorización descritos hasta ahora. Responda a las siguientes cuestiones:

A la vista de los resultados, ¿qué elemento de proceso le parece más costoso? ¿Red? ¿CPU? ¿Acceso a datos? En otras palabras, ¿cuál fue el elemento más utilizado durante la monitorización con nmon en un entorno virtual? (CPU, Memoria, disco ...)

Nota: Las respuestas a estas cuestiones deben estar acompañadas por datos que respalden la argumentación, en forma de pantallazos de nmon (o gráficas de Nmon Visualizer) y pantallazos de si2-monitor-sh.

Repetimos la ejecución obteniendo el siguiente resultado:



Durante la monitorización, podemos observar como la CPU es el elemento más utilizado.

¿Le parece una situación realista la simulada en este ejercicio? ¿Por qué?

Este ejercicio no es una representación realista, ya que muestra un uso con muy pocos clientes (en concreto, 1), lo que produce que nunca lleguen a entrar a la cola. En un caso realista el número de clientes sería mayor y el sistema reaccionaría de manera diferente.

Teniendo en cuenta cuál ha sido el elemento más saturado, proponga otro esquema de despliegue que resuelva esa situación.

Para mejorar la sobrecarga de la CPU se puede aumentar la cantidad de cpu de la máquina de ejecución, o, como solución más viable se podría distribuir cliente y servidor en diferentes máquinas.

Ejercicio VIII

Obtener la curva de productividad, siguiendo los pasos que se detallan a continuación:

Previamente a la ejecución de la prueba se lanzará una ejecución del script de pruebas (unas 10 ejecuciones de un único usuario) de la que no se tomarán resultados, para iniciar el sistema y preparar medidas consistentes a lo largo de todo al proceso.

Borrar los resultados de la ejecución anterior. En la barra de acción de JMeter, seleccionar Run -> Clear All.

Borrar los datos de pagos en la base de datos VISA.

Ejecutar la herramienta de monitorización nmon en ambas máquinas, preferiblemente en modo "Data-collect" (Ver 8.2.2).

Seleccionar el número de usuarios para la prueba en JMeter (parámetro C de la prueba)

Conmutar en JMeter a la pantalla de presentación de resultados, Aggregate Report.

Ejecutar la prueba. En la barra de acción de JMeter, seleccionar Run -> Start.

Ejecutar el programa de monitorización si2-monitor.sh

- Arrancarlo cuando haya pasado el tiempo definido como rampa de subida de usuarios en JMeter (el tiempo de ejecución en JMeter se puede ver en la esquina superior derecha de la pantalla).
- Detenerlo cuando esté a punto de terminar la ejecución de la prueba. Este momento se puede detectar observando cuando el número de hilos concurrentes en JMeter (visible en la esquina superior derecha) comienza a disminuir (su máximo valor es C).

Registrar los resultados que proporciona la monitorización en la hoja de cálculo.

Durante el periodo de monitorización anterior, vigilar que los recursos del servidor si2srv02 y del ordenador que se emplea para realizar la prueba no se saturen. En caso de usar nmon de forma interactiva, se deben tomar varios pantallazos del estado de la CPU durante la prueba, para volcar en la hoja de cálculo del dato de uso medio de la CPU (CPU average %). En caso de usar nmon en modo "Data-collect", esta información se puede ver posteriormente en NMonVisualizer. Una tercera opción (recomendada) es ejecutar el comando vmstat en una terminal remota a la máquina si2srv02, para extraer directamente el valor de uso medio de su CPU 5.

- Finalizada la prueba, salvar el resultado de la ejecución del Aggregate Report en un archivo, y registrar en la hoja de cálculo de resultados los valores Average, 90% line y Throughput para las siguientes peticiones:
- ProcesaPago.
- Total.

Una vez realizadas las iteraciones necesarias para alcanzar la saturación, representar la curva de Throughput versus usuarios. Incluir el fichero P2-curvaProductividad.jmx en la entrega.

Nota: Todos los datos de monitorización que se entreguen como parte de la curva de rendimiento (derivados de JMeter, nmon y si2-monitor.sh) deben estar respaldados por pruebas que demuestren su veracidad, ya sea en la propia memoria o en ficheros adicionales (recomendado). En el caso de los recursos del sistema, se pueden mostrar tanto pantallazos del modo interactivo de nmon como gráficas de NMONvisualizer.

250

Etiqueta	# Muestras	Media	Mediana	90% Line	95% Line	99% Line	Min	Máx	% Error	Rendimiento	Kb/sec	Sent KB/sec
/P1-base/comienzapago	250	786	538	2058	2223	2491	2	2549	0,00%	25,2/sec	48,84	0,00
/P1-base/procesapago	250	781	469	2353	2473	2614	6	2627	0,00%	32,3/sec	39,45	0,00
Total	500	783	521	2255	2409	2548	2	2627	0,00%	50,3/sec	79,48	0,00

500

Etiqueta	# Muestras	Media	Mediana	90% Line	95% Line	99% Line	Min	Máx	% Error	Rendimiento	Kb/sec	Sent KB/sec
/P1-base/comienzapago	750	271	4	1218	1851	2351	2	2549	0,00%	14,6/sec	28,41	0,00
/P1-base/procesapago	750	276	10	786	2255	2541	5	2627	0,00%	15,3/sec	15,92	0,00
Total	1500	274	8	906	2129	2474	2	2627	0,00%	29,3/sec	43,64	0,00

1000

Etiqueta	# Muestras	Media	Mediana	90% Line	95% Line	99% Line	Min	Máx	% Error	Rendimiento	Kb/sec	Sent KB/sec
/P1-base/comienzapago	1750	220	77	425	971	2162	1	2549	0,00%	21,0/sec	40,84	0,00
/P1-base/procesapago	1750	236	106	446	726	2420	5	2627	0,00%	21,6/sec	21,40	0,00
Total	3500	228	91	436	786	2348	1	2627	0,00%	42,1/sec	61,67	0,00

1500

```
eps@labvirteps:~/Escritorio/P2$ bash si2-monitor.sh 10.8.7.2
#Muestra numJDBCCount numHTTPCount numHTTPQ
0 4 5 534
1 5 5 508
2 4 5 468
3 5 5 427
4 3 5 366
5 3 5 0
6 0 0 0
^C
TOT.MUESTRAS MEDIA:
7 3.42857 4.28571 329
eps@labvirteps:~/Escritorio/P2$
```

Etiqueta	# Muestras	Media	Mediana	90% Line	95% Line	99% Line	Min	Máx	% Error	Rendimiento	Kb/sec	Sent KB/sec
/P1-base/comienzapago	3250	1076	376	3534	4040	8480	1	8720	0,00%	26,1/sec	50,60	0,00
/P1-base/procesapago	3250	1215	397	3977	4144	4421	5	8699	0,00%	26,4/sec	24,60	0,00
Total	6500	1145	385	3808	4129	4489	1	8720	0,00%	51,9/sec	74,59	0,00

2000

```
eps@labvirteps:~/Escritorio/P2$ bash si2-monitor.sh 10.8.7.2
#Muestra numJDBCCount numHTTPCount numHTTPQ
0 5 5 583
1 5 5 586
^C
TOT.MUESTRAS MEDIA:
2 5 5 584.5
eps@labvirteps:~/Escritorio/P2$
```


Etiqueta	# Muestras	Media	Mediana	90% Line	95% Line	99% Line	Min	Máx	% Error	Rendimiento	Kb/sec	Sent KB/sec
/P1-base/comienzapago	5250	2815	1409	9428	12897	15542	1	31533	0,00%	26,0/sec	50,55	0,00
/P1-base/procesapago	5250	1904	1030	4010	4312	15435	5	31386	0,00%	26,3/sec	23,34	0,00
Total	10500	2360	1218	4229	10712	15510	1	31533	0,00%	52,1/sec	73,63	0,00

2500

```
eps@labvirteps:~/Escritorio/P2$ bash si2-monitor.sh 10.8.7.2
```

#Muestra	numJDBCCount	numHTTPCount	numHTTPQ
0	5	5	582
1	5	5	515
2	5	5	439
3	2	5	434
4	1	5	429
5	4	5	451
6	5	5	472
7	5	5	494
8	5	5	499
9	2	5	398
10	2	5	374
11	3	5	369
12	5	5	431
13	1	5	301
14	4	5	322
15	5	5	95
16	0	0	0
17	0	0	0
18	0	0	0
19	0	0	0
20	0	0	0
21	0	0	0

Informe Agregado

Nombre:

Comentarios

Escribir todos los datos a Archivo

Nombre de archivo Log/Mostrar sólo: ☐ Escribir en Log Sólo Errores ☐ Éxitos

Etiqueta	# Muestras	Media	Mediana	90% Line	95% Line	99% Line	Min	Máx	% Error	Rendimiento	Kb/sec	Sent KB/sec
/P1-base/comienzapago	7750	4535	2146	15446	16577	31510	1	64121	0,00%	25,4/sec	49,40	0,00
/P1-base/procesapago	7750	2433	1498	4081	9279	16834	5	64187	0,00%	25,6/sec	21,91	0,00
Total	15500	3484	1827	11250	15950	31423	1	64187	0,00%	50,9/sec	71,14	0,00

3000

```

eps@labvirteps:~/Escritorio/P2$ bash si2-monitor.sh 10.8.7.2
#Muestra numJDBCCount numHTTPCount numHTTPQ
0 5 5 557
1 3 5 560
2 5 5 570
3 2 5 560
4 5 5 562
5 5 5 545
6 0 5 535
7 5 5 551
8 1 5 482
9 3 5 429
10 3 5 345
11 4 5 234
12 5 5 112
13 1 0 0
14 0 0 0
15 0 0 0
16 0 0 0
17 0 0 0
18 0 0 0
19 0 0 0
20 0 0 0
21 0 0 0
22 0 5 13
23 4 5 76
24 4 5 65
25 0 5 34
26 0 0 0
27 0 0 0
28 0 0 0
29 0 0 0
30 0 0 0
31 0 0 0
32 0 0 0
33 0 0 0
34 0 0 0
35 0 0 0

```

Informe Agregado

Nombre:

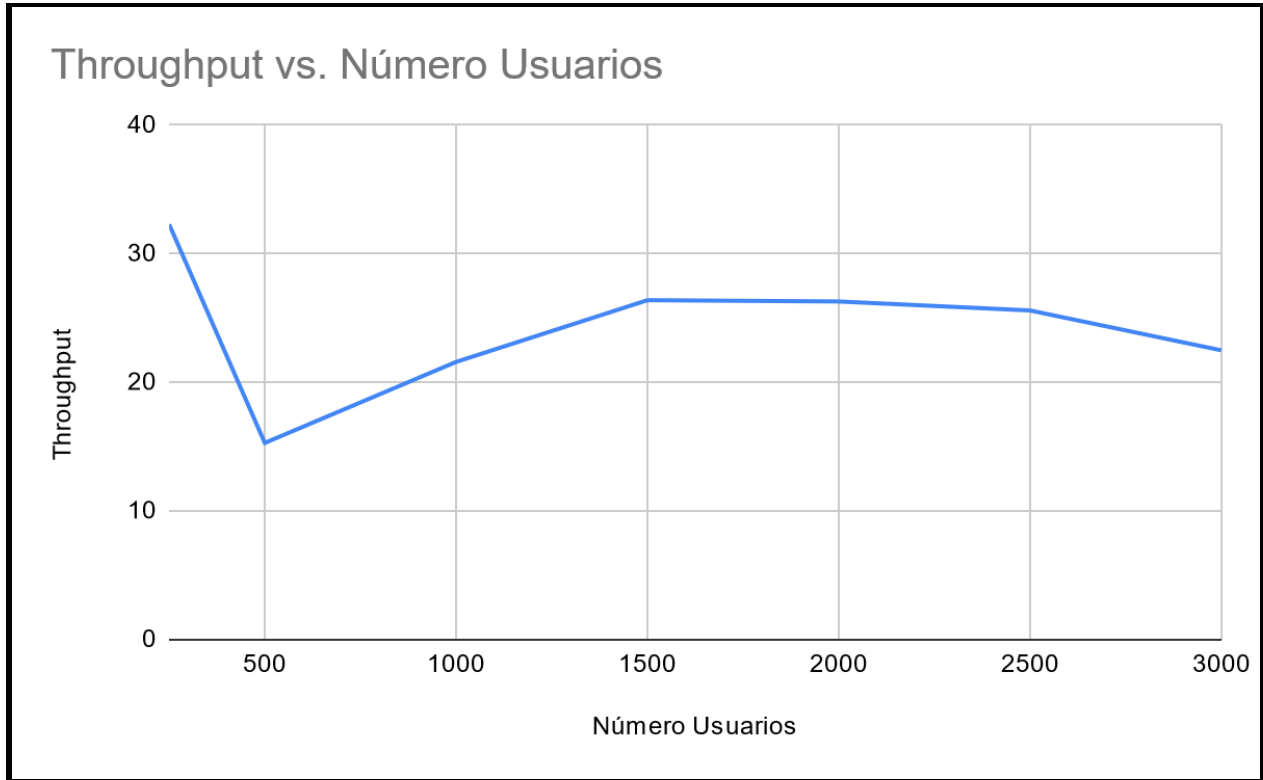
Comentarios

Escribir todos los datos a Archivo

Nombre de archivo Log/Mostrar sólo: ☐ Escribir en Log Sólo Errores ☐ Éxitos

Etiqueta	# Muestras	Media	Mediana	90% Line	95% Line	99% Line	Mín	Máx	% Error	Rendimiento	Kb/sec	Sent KB/sec
/P1-base/comienzapago	10750	6820	2894	17691	31458	31916	1	131217	0.25%	22.4/sec	43.42	0.00
/P1-base/procesapago	10750	3096	1846	4153	12755	31548	5	131139	0.14%	22.5/sec	18.72	0.00
Total	21500	4958	2243	15477	18604	31754	1	131217	0.20%	44.7/sec	62.05	0.00

Por lo tanto, la curva de productividad consiste en:



Donde se puede observar la saturación a partir de los 3000 usuarios activos.

Ejercicio IX

Responda a las siguientes cuestiones:

A partir de la curva obtenida, determinar para cuántos usuarios conectados se produce el punto de saturación, cuál es el throughput que se alcanza en ese punto, y cuál el throughput máximo que se obtiene en zona de saturación.

Analizando los valores de monitorización que se han ido obteniendo durante la elaboración de la curva, sugerir el parámetro del servidor de aplicaciones que se cambiaría para obtener el punto de saturación en un número mayor de usuarios.

Realizar el ajuste correspondiente en el servidor de aplicaciones, reiniciarlo y tomar una nueva muestra cercana al punto de saturación. ¿Ha mejorado el rendimiento del sistema? Documente en la memoria de prácticas el cambio realizado y la mejora obtenida.

El punto de saturación se produce para 1500 usuarios aproximadamente.

Observando la monitorización obtenida en el apartado anterior, podemos concluir que si aumentamos el tamaño máximo de la cola, el punto de saturación aumentaría, ya que hay más clientes a la espera de ser procesados antes de que el servidor se sature.

Comprobamos los resultados de aumentar la longitud de la cola, y, como era de esperar, el punto de saturación aumenta.