		<b>Escuela Politécnica Superior</b> <b>Ingeniería Informática</b> <b>Prácticas de Sistemas Informáticos 2</b>			
<b>Grupo</b>	<b>2401</b>	<b>Práctica</b>	2	<b>Fecha</b>	22/03/2021
<b>Alumno/a</b>	Rivas Molina, Lucía				
<b>Alumno/a</b>	Santo-Tomás López, Daniel				

## Práctica 2: Rendimiento

Nota: el script *limpiar.sh* limpia, repliega y despliega la primera parte de la práctica mientras que *reset.sh* lo utilizamos en la segunda.

### Ejercicio número 1:

**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.**

El fichero P2.jmx se encuentra en la carpeta de la entrega. Hemos seguido los pasos descritos en el enunciado para generar los planes de pruebas usando jmeter: los parámetros para generar peticiones, la variable de usuario samples para indicar el número de pruebas, los hilos simulados, el importe del pago, el identificador de la transacción, el generador de peticiones y el aggregate report.

### Ejercicio número 2:

**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**

La dirección IP del PC es 10.10.1.41

Y las de las máquinas virtuales son: 10.2.1.1 para la VM1 del host y 10.2.1.2 para la VM2.

- **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.**

En P1-base hemos cambiado el nombre a P1-base y la IP del host a 10.2.1.2.

En P1-ejb-cliente remoto hemos cambiado la IP del host a 10.2.1.2.

En P1-ejb-servidor-remoto hemos cambiado las IPs del cliente y servidor a 10.2.1.1.

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

Cambiamos la IP a 10.2.1.1 porque el servidor se despliega en la primera máquina virtual.

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

o El destino de despliegue de la aplicación P1-base será PC2VM con IP 10.X.Y.2 (as.host)

o 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)

o 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)  
o 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.

Para el PC local:

```
danist@danist-Lenovo-E51-80:~/Documentos/UAM/SEGUNDOCUATRI/SI2/Practica2/P2/P1-ws$ free
total        used        libre    compartido búfer/caché    disponible
Memoria:    8024412    2912280    1207736    1358920    3904396    3459732
Swap:      7999484         0     7999484
```

```
nmmon-16g Hostname=danist-Lenovo Refresh= 2secs 14:44.42
Memory and Swap
PageSize:4KB RAM-Memory Swap-Space High-Memory Low-Memory
Total (MB) 7836.3 7812.0 - not in use - not in use
Free (MB) 1192.9 7812.0
Free Percent 15.2% 100.0%
Linux Kernel Internal Memory (MB)
Cached= 3364.7 Active= 3310.9
Buffers= 255.6 Swapcached= 0.0 Inactive = 2921.2
Dirty = 1.1 Writeback = 0.0 Mapped = 988.3
Slab = 247.0 Commit_AS = 11685.7 PageTables= 70.9
```

Para la MV1:

```
si2@si2srv01:~$ free
total        used        free      shared    buffers    cached
Mem: 767168 598404 168764      0      35304 188904
-/+ buffers/cache: 374196 392972
Swap: 153592      0 153592
```

```
nmmon-12f [H for help] Hostname=si2srv01 Refresh= 2secs 05:45.26
Memory Stats
RAM High Low Swap
Total MB 749.2 0.0 749.2 150.0
Free MB 163.2 0.0 163.2 150.0
Free Percent 21.8% 0.0% 21.8% 100.0%
MB MB MB MB
Cached= 184.6 Active= 458.1
Buffers= 34.5 Swapcached= 0.0 Inactive = 105.1
Dirty = 0.1 Writeback = 0.0 Mapped = 27.2
Slab = 14.0 Commit_AS = 1129.2 PageTables= 1.9
```

Para la MV2:

```
si2@si2srv02:~$ free
total        used        free      shared    buffers    cached
Mem: 767168 587160 180008      0      20924 177368
-/+ buffers/cache: 388868 378300
Swap: 153592      0 153592
```

Archivo Editar Ver Buscar Terminal Ayuda					
nmon-12f [H for help] Hostname=si2srv02 Refresh= 2secs 05:46.07					
Memory Stats					
	RAM	High	Low	Swap	
Total MB	749.2	0.0	749.2	150.0	
Free MB	174.2	0.0	174.2	150.0	
Free Percent	23.2%	0.0%	23.2%	100.0%	
	MB		MB	MB	
Buffers=	20.4	Cached=	173.3	Active=	459.0
Dirty =	0.0	Swapcached=	0.0	Inactive =	94.8
Slab =	13.1	Writeback =	0.0	Mapped =	22.6
		Commit_AS =	1056.9	PageTables=	1.5

### Ejercicio número 3:

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;`

SELECT COUNT(*) FROM PAGO	
Result	Execution plan Visualize Logging
#	count
1	3000

- Compruebe que ninguna de las peticiones ha producido un error. Para ello revise que la columna %Error indique 0% en todos los casos.

Una vez que los resultados han sido satisfactorios:

- Añote los resultados del informe agregado en la memoria de la práctica.

Tras ejecutar el plan de ejecución varias veces, podemos ver que el mejor tarda en total 1 minuto 14 segundos.

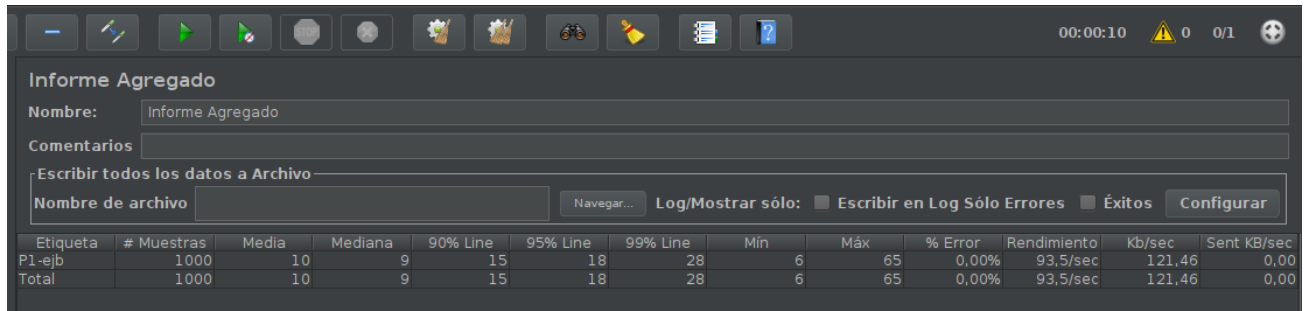
jmx (/home/danist/Documentos/UAM/SEGUNDOCUATRI/SI2/Practica2/P2/P2.jmx) - Apache JMeter (5.2.1)												
Informe Agregado												
Nombre:		Informe Agregado										
Comentarios												
Escribir todos los datos a Archivo												
Nombre de archivo		Navegar...		Log/Mostrar sólo:		Escribir en Log Sólo Errores		Éxitos		Configurar		
Etiqueta	# Muestras	Media	Mediana	90% Line	95% Line	99% Line	Min	Máx	% Error	Rendimiento	Kb/sec	Sent KB/sec
P1-base	1000	10	10	14	16	26	7	43	0,00%	92,3/sec	118,58	0,00
P1-ws	1000	47	45	58	63	77	37	468	0,00%	20,8/sec	26,99	0,00
P1-ejb	1000	15	14	20	22	27	10	38	0,00%	64,3/sec	49,46	0,00
Total	3000	24	15	49	55	69	7	468	0,00%	40,3/sec	45,02	0,00

- Salve el fichero server.log que se encuentra en la ruta glassfish/domains/domain1/logs de Glassfish y adjúntelo con la práctica.
- Añada a la memoria de prácticas la siguiente información: ¿Cuál de los resultados le parece el mejor? ¿Por qué? ¿Qué columna o columnas elegiría para decidir este resultado?

Nos parece mejor el resultado de P1-base pues es el que tiene los mejores datos en el procesamiento (media, mediana y mínimo). Además, tiene el mayor rendimiento con 92.3/segundo, siendo ésta la columna

que consideramos más importante, pues aporta mayor información y es la base de las demás.

**Incluir el directorio P2 en la entrega. 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	Min	Máx	% Error	Rendimiento	Kb/sec	Sent KB/sec
P1-ejb	1000	10	9	15	18	28	6	65	0,00%	93,5/sec	121,46	0,00
Total	1000	10	9	15	18	28	6	65	0,00%	93,5/sec	121,46	0,00

Podemos observar ahora que todos los tiempos se han disminuido bastante, siendo el tiempo total de 10 segundos, luego el rendimiento ha aumentado considerablemente.

## Ejercicio número 4:

**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 \$J2EE\_HOME/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 del laboratorio con scp.**

Estos cambios son respecto del PC2:

1. Adaptamos la máquina virtual a modo servidor
2. Establecemos los valores de memoria máximos y mínimos.
3. Eliminamos el despliegue y la recarga automática.
4. Activamos la precompilación y los niveles de monitorización pedidos.

Una vez configurado, detenemos el servidor y copiamos el archivo de configuración domain.xml con el comando scp en el pc local.

**Revisar el script si2-monitor.sh e indicar los mandatos asadmin que debemos ejecutar en el Host PC1 para averiguar los valores siguientes, mencionados en el Apéndice 1, del servidor P1VM1:**

Tras revisar el script indicado, los comandos pedidos son:

### 1. Max Queue Size del Servicio HTTP

```
asadmin --host 10.1.2.1 --user admin --passwordfile ./passwordfile get configs.config.thread-pools.thread-pool.http-thread-pool.max-queue-size configs.config.thread-pools.thread-pool.http-thread-pool.max-queue-size=4096
```

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

```
asadmin --host 10.1.2.1 --user admin --passwordfile get resources.jdbc-connection-pool.VisaPool.max-pool-size resources.jdbc-connection-pool.VisaPool.max-pool-size=32
```

### Ejercicio número 5:

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

Se pueden encontrar en el excel adjunto:

Parámetros de configuración		
Elemento	Parámetro	Valor
JVM Settings	Heap Máx. (MB)	512
JVM Settings	Heap Mín. (MB)	512
HTTP Service	Max.Thread Count	5
HTTP Service	Queue size	4096
Web Container	Max.Sessions	-1
Visa Pool	Max.Pool Size	32

### Ejercicio número 6:

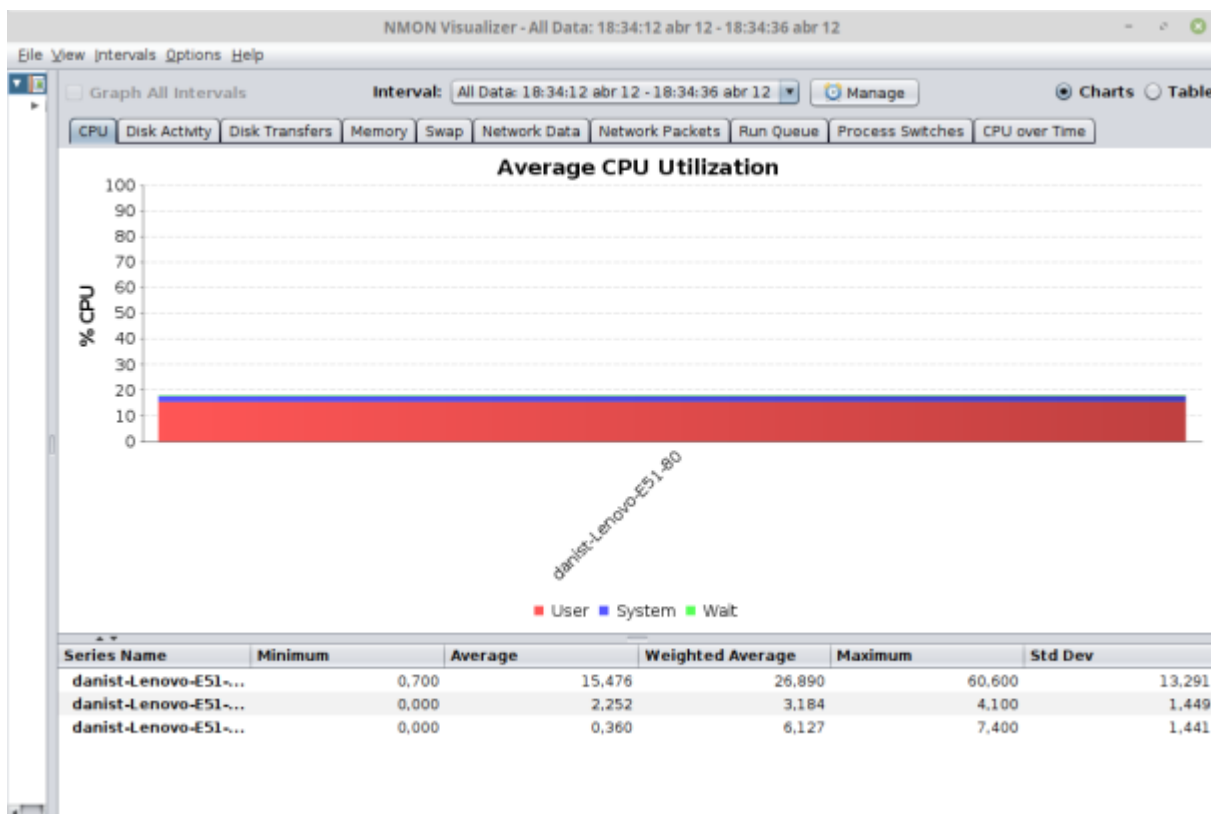
Tras habilitar la monitorización en el servidor, repita la ejecución del plan de pruebas anterior.

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 costosa? ¿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, ...)

El resultado de nmon en el pc local con pruebas es:



El resultado del script de monitorización es:

^C			
TOT.MUESTRAS	MEDIA:		
14	0.0714286	0	0

Podemos ver que el sistema más saturado es la CPU, sin embargo, tampoco es que esté demasiado saturada, no destaca.

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

No parece una situación muy realista dado que la red y el entorno donde ejecutamos las pruebas están controlados y con características fijas, y no existen situaciones impredecibles como en la vida real, donde un dispositivo y sistema distribuido conectado a internet puede tenerlas.

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

La CPU es el sistema más saturado, de modo que si aumentásemos el procesador y su capacidad lo mejoraríamos. Sin embargo, estamos en un sistema real y no es posible cambiarlo (como lo sería en una máquina virtual). Sin embargo, tampoco está muy saturado.

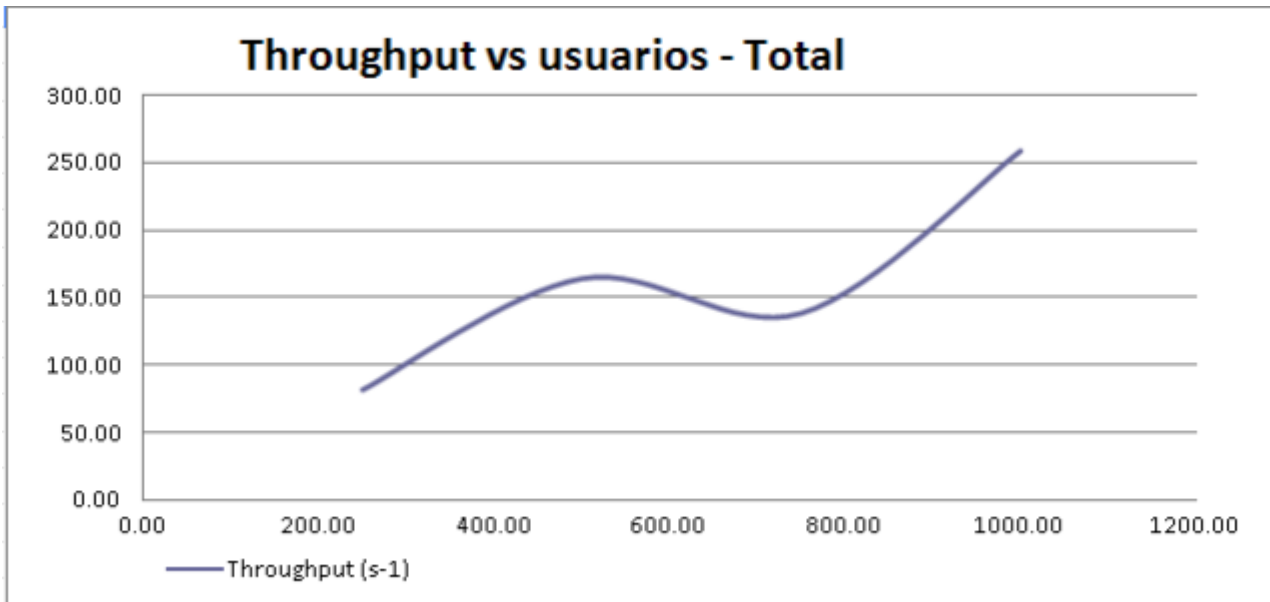
## ***Ejercicio número 8:***

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

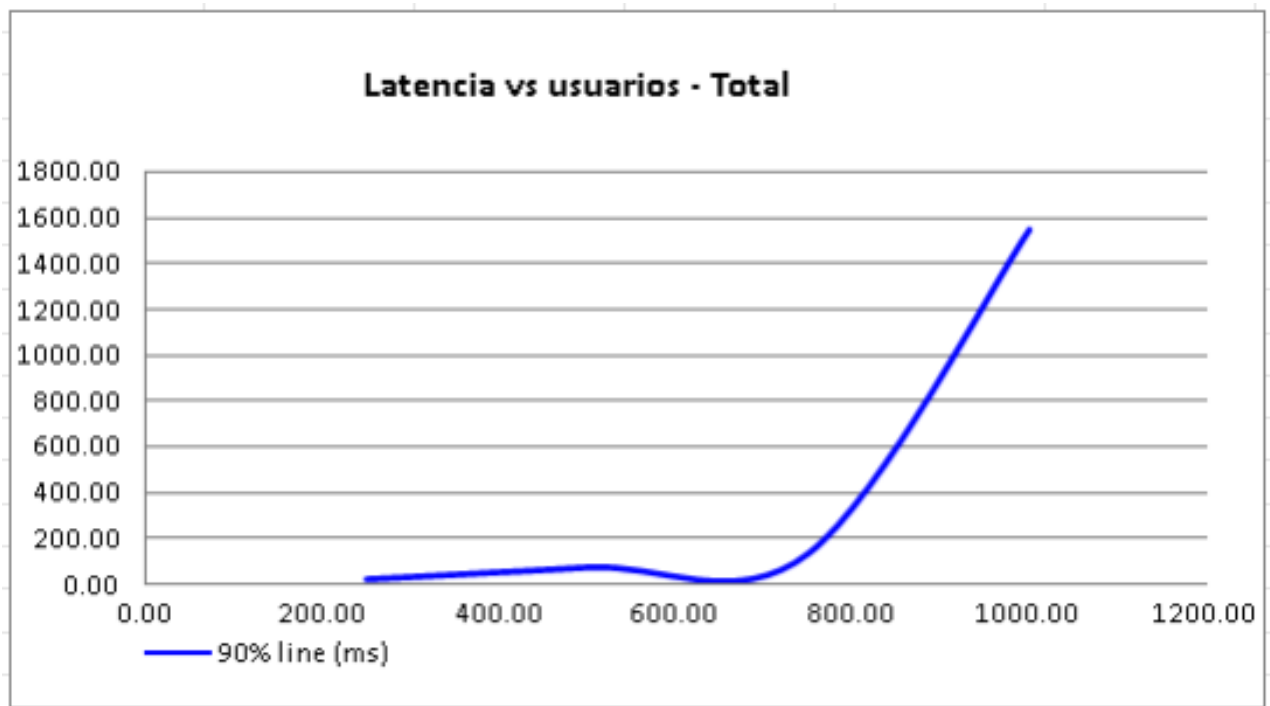
- Previamente a la ejecución de la prueba se lanzará una ejecución del script de pruebas (unas 10 ejecuciones de un único usuario) de la que no se tomarán resultados, para iniciar el sistema y preparar medidas consistentes a lo largo de todo al proceso. Borrar los resultados de la ejecución anterior. En la barra de acción de JMeter, seleccionar Run -> Clear All.
- Borrar los datos de pagos en la base de datos VISA.
- 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.
  - Detenerlo cuando esté a punto de terminar la ejecución de la prueba.
  - 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, mediante inspección del programa de monitorización nmon que se ejecuta en ambas máquinas.
- 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.

Tras ejecutarlo varias veces nos salen las siguientes gráficas:



Podemos observar que el rendimiento va creciendo, excepto cuando tenemos 750 hilos, que decae disimuladamente. No hemos podido ejecutar más de 1000 hilos pues nuestros ordenadores se saturaban, precisamente, en la siguiente gráfica de la latencia podemos observar que la curva se dispara hacia arriba cuando llega al punto de saturación, a partir de 1000.



Adjuntamos, además, los ficheros resultantes del ejercicio.

## Ejercicio número 9:

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.

Como ya hemos comentado antes, la saturación se obtiene a partir de 1000 usuarios, pues ni siquiera podíamos ejecutarlo con 1250. En este punto el throughput máximo es de 250 peticiones por segundo, y el máximo de throughput alcanzado es un poco más de 250 peticiones por segundo.

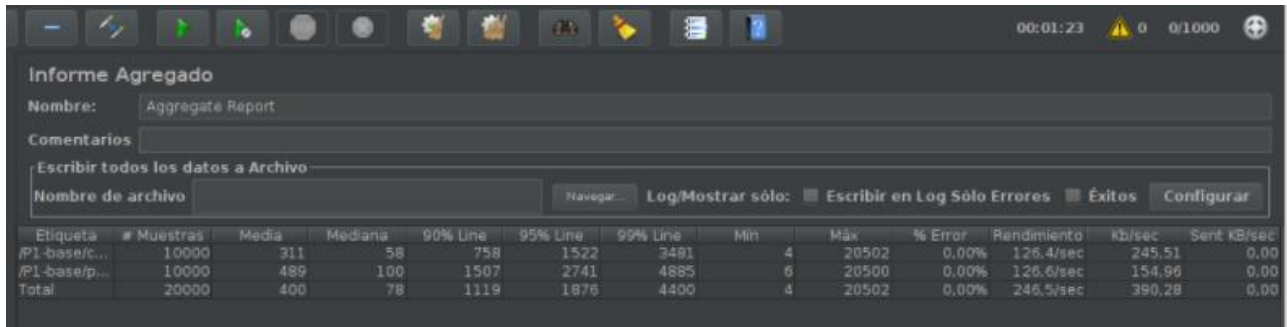
- Analizando los valores de monitorización que se han ido obteniendo durante la elaboración de la

curva, sugerir el parámetro del servidor de aplicaciones que se cambiaría para obtener el punto de saturación en un número mayor de usuarios.

En ejercicios anteriores ya vimos que el parámetro que limita al servidor es la capacidad de procesamiento, luego para poder tener más usuarios tendríamos que aumentar la capacidad, lo cual no podemos hacer en nuestros ordenadores locales. Pero en la máquina virtual sí que podemos poner más procesadores. Además, también podemos poner más hilos en el servidor.

- 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.

Pues tras poner más hilos en las peticiones http del servidor, concretamente los hemos subido de 5 a 500, podemos observar que el cambio más significativo es que la latencia y el average general han disminuido. Por otro lado, en procesa\_pago ha mejorado disimuladamente.



The screenshot shows the JMeter Aggregate Report window. At the top, there's a toolbar with various icons and a status bar showing '00:01:23', '0' warnings, and '0/1000' errors. The main area is titled 'Informe Agregado' (Aggregate Report). Below the title, there's a 'Nombre:' field containing 'Aggregate Report' and a 'Comentarios:' field. A section titled 'Escribir todos los datos a Archivo' (Write all data to file) contains a 'Nombre de archivo' field, a 'Navegar' button, and checkboxes for 'Log/Mostrar sólo: Escribir en Log Sólo Errores' and 'Éxitos', along with a 'Configurar' button. The bottom section is a table with performance metrics.

Etiqueta	# Muestras	Media	Mediana	90% Line	95% Line	99% Line	Min	Máx	% Error	Rendimiento	kb/sec	Sent kb/sec
/P1-base/c...	10000	311	58	758	1522	3481	4	20502	0.00%	126.4/sec	245.51	0.00
/P1-base/p...	10000	489	100	1507	2741	4885	6	20500	0.00%	126.6/sec	154.96	0.00
Total	20000	400	78	1119	1876	4400	4	20502	0.00%	246.5/sec	390.28	0.00