

Entrega 2 – Análisis de Capacidad

Aplicaciones Web Escalables en un Entorno Tradicional

Grupo 20

Juan P Hernández

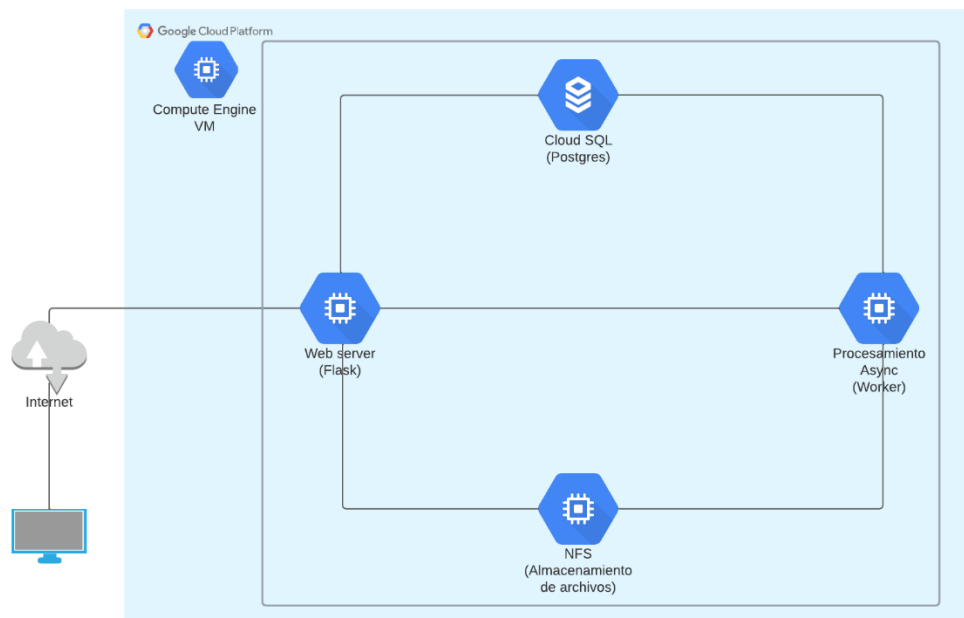
Juan D Díaz

Juan F Castaño

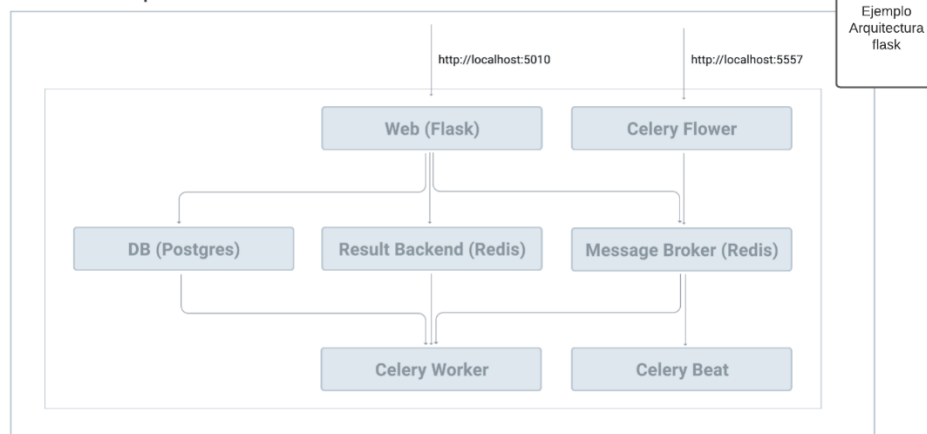
Kevin H Castrillón

1. Arquitectura de la aplicación.

1.1. Diagrama Arquitectura



Servicios de aplicaciones



1.2. Infraestructura

Para la infraestructura se utilizó las siguientes soluciones tecnológicas:

Máquina virtual: se configuraron tres instancias de tipo N1 con los requerimientos de cómputo y almacenamiento definidos, es decir, 1 GB de RAM, 1vCPU y 20GB de RAM. El sistema operativo instalado en las mismas es Ubuntu 20.04. Tienen unas etiquetas de red para que se pudiese proceder a comunicarse entre ellas, y además que aceptaran conexiones a través de internet.

Instancia SQL: se configuró una instancia de SQL en la cual corrimos PostgreSQL 14, esta instancia se configuro en modo de desarrollo, con 1 núcleos y 1GB de ram, y 10 gb de almacenamiento interno.

Red VPC: se aprovechó la red VPC configurada por defecto en un proyecto de Google Cloud. Se configuraron reglas de firewall respectivas para poder acceder a la máquina virtual por TCP en el puerto 5000. Se aprovechó el puerto 3389 que ya está configurado por defecto para acceder por TCP.

Para el ambiente de software se usaron las siguientes tecnologías:

A continuación, se listan las tres principales herramientas tecnológicas usadas en la máquina virtual para el desarrollo de la entrega 2:

Flask: framework web minimalista para Python. Es una herramienta popular utilizada para construir aplicaciones web pequeñas y medianas que son rápidas y eficientes.

Celery: herramienta de Python que permite la programación de tareas en segundo plano y la ejecución de trabajos de forma asíncrona y distribuida. Se utiliza ampliamente en aplicaciones web para procesar tareas que pueden ser realizadas fuera del ciclo de solicitud-respuesta.

Redis: sistema de almacenamiento en memoria de código abierto, rápido y eficiente que se utiliza como base de datos, caché y agente de mensajes. Fue diseñado para manejar grandes cantidades de datos y proporcionar un alto rendimiento y disponibilidad.

2. Análisis de capacidad:

2.1 Entorno de pruebas.

Máquina virtual: para esta entrega se configuraron tres instancias de tipo N1 con los requerimientos de cómputo y almacenamiento definidos, es decir, 1 GB de RAM, 1vCPU y 20GB de Almacenamiento interno. El sistema operativo instalado en las mismas es Ubuntu 20.04. Se le agregó una etiqueta de red para que las reglas necesarias de firewall para acceder desde internet fueran posibles.

Instancia SQL: se configuró una instancia de SQL en la cual corrimos PostgreSQL 14, esta instancia se configuro en modo de desarrollo, con 2 núcleos y 1GB de ram, y 10 gb de almacenamiento interno.

Red VPC: se aprovechó la red VPC configurada por defecto en un proyecto de Google Cloud. Se configuraron reglas de firewall respectivas para poder acceder a la máquina virtual por TCP

en el puerto 5000. Se aprovechó el puerto 3389 que ya está configurado por defecto para acceder por TCP.

Para el ambiente de software se usaron las siguientes tecnologías:

A continuación se listan las tres principales herramientas tecnológicas usadas en la máquina virtual para el desarrollo de la entrega 2:

Flask: framework web minimalista para Python. Es una herramienta popular utilizada para construir aplicaciones web pequeñas y medianas que son rápidas y eficientes.

Celery: herramienta de Python que permite la programación de tareas en segundo plano y la ejecución de trabajos de forma asíncrona y distribuida. Se utiliza ampliamente en aplicaciones web para procesar tareas que pueden ser realizadas fuera del ciclo de solicitud-respuesta.

Redis: sistema de almacenamiento en memoria de código abierto, rápido y eficiente que se utiliza como base de datos, caché y agente de mensajes. Fue diseñado para manejar grandes cantidades de datos y proporcionar un alto rendimiento y disponibilidad.

Paquetes de Python

```
flask==2.2.3
flask-SQLAlchemy==3.0.3
flask-RESTful==0.3.9
flask-marshmallow==0.15.0
marshmallow_sqlalchemy==0.29.0
marshmallow==3.19.0
flask-jwt-extended==4.4.4
flask-cors==3.0.10
werkzeug==2.2.3
celery==5.2.7
redis==4.5.4
gunicorn==20.1.0
psycpg2-binary==2.9.2
pydump==0.25.1
pylzma==0.5.0
```

2.2 Criterios de aceptación

Los criterios de aceptación para las pruebas de estrés incluyen los siguientes objetivos y limitaciones:

Tiempo de respuesta: se espera que la aplicación responda a las solicitudes de los usuarios en un tiempo razonable, sin generar tiempos de espera excesivos. El objetivo es que el tiempo de respuesta promedio sea inferior a 3 segundos, con un límite máximo de 5 segundos.

Rendimiento: se espera que la aplicación pueda manejar una carga de trabajo adecuada, sin experimentar una disminución significativa en el rendimiento. El objetivo es que la aplicación pueda manejar al menos 100 usuarios concurrentes sin afectar negativamente el rendimiento.

Utilización de recursos: se espera que la aplicación utilice los recursos del sistema de manera eficiente, sin generar cuellos de botella en el procesamiento o en la utilización de memoria. El objetivo es que la utilización de recursos se mantenga por debajo del 80% en todo momento.

Además de estos objetivos y limitaciones, los criterios de éxito del proyecto también incluyen la identificación de los cuellos de botella en la aplicación y su infraestructura, y la determinación de las configuraciones óptimas para maximizar el rendimiento.

2.3 Escenarios de prueba

Escenarios de prueba:

1. Conversión de (1, 10, 100) archivos pequeños a ZIP.
2. Conversión de (1, 10, 100) archivos grandes a ZIP.
3. Conversión de (1, 10, 100) archivos pequeños a 7Z.
4. Conversión de (1, 10, 100) archivos grandes a 7Z.
5. Conversión de (1, 10, 100) archivos pequeños a TAR.GZ.
6. Conversión de (1, 10, 100) archivos grandes a TAR.GZ.
7. Conversión de (1, 10, 100) archivos pequeños a TAR.BZ2.
8. Conversión de (1, 10, 100) archivos grandes a TAR.BZ2.
9. Obtención de (1, 10, 100) archivos grandes.
10. Obtención de (1, 10, 100) archivos pequeños.
11. Autenticación de (1, 10, 100) usuarios concurrentes.
12. Crear (1, 10, 100) usuarios.
13. Obtener (1, 10, 100) tareas de conversión de forma simultánea.
14. Eliminar una tarea.
15. Cambiar el formato de una tarea (1, 10, 100) de forma simultánea

Escenarios clave:

- Comprobar que la aplicación puede manejar diferentes tipos de archivos y tamaños.
- Verificar que la conversión de los archivos se realiza correctamente a los diferentes formatos.
- Validar que la aplicación no pierda información o datos durante el proceso de compresión.
- Asegurar que la aplicación mantiene la estructura de carpetas durante la compresión.
- Asegurar que un usuario pueda crear una cuenta.
- Asegurar que un usuario pueda hacer log-in de forma correcta.
- Permitir que se pueda comprobar que las tareas se crean y se suben en el sistema.

Variabilidad entre servicios representativos:

Se puede simular la variabilidad en los servicios representativos de la aplicación mediante la generación de diferentes tipos de archivos de diferentes tamaños y formatos.

Datos de prueba:

- Archivos pequeños y grandes de diferentes formatos.
- Varios archivos para comprimir.
- Si el archivo comprimido se puede abrir correctamente.
- Si la estructura de carpetas se mantiene durante la compresión.
- Credenciales de acceso.
- Datos en texto plano correspondiente a formularios.
- Información de una tarea de conversión.

2.4 Parámetros de configuración

Los parámetros para las peticiones GET, PUT y DELETE se configuran inicialmente en 100 threads y van creciendo o decreciendo hasta el punto de éxito o fallo del escenario, además se autoriza a través de jwt.

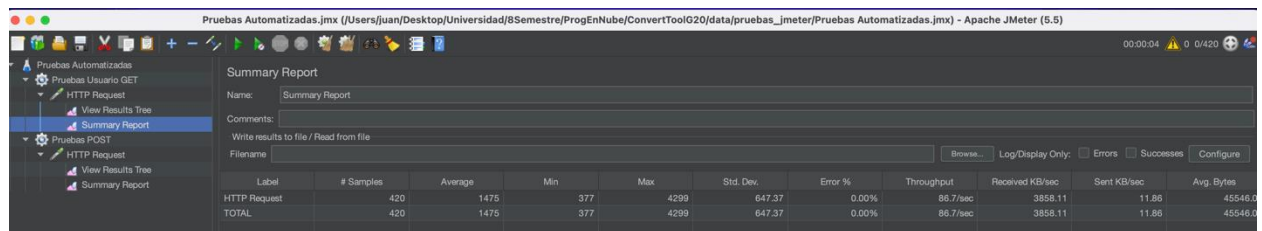
Los parámetros para el POST se configuran inicialmente en 100 threads y van decreciendo hasta el punto de éxito o fallo del escenario, además se autoriza a través de jwt.

2.5 Escenario 1

Obtener un archivo convertido.

Se prueba obtener un archivo procesado, la restricción es la siguiente (En las pruebas de estrés el tiempo de respuesta promedio de la aplicación debe ser de máximo 1.500 ms, si este tiempo no se cumple, se concluye que el sistema NO soporta la cantidad de requests de la prueba. En caso de que durante una prueba se generen más de un 1% de errores en los requests de la prueba, se concluye que la aplicación NO soporta la cantidad de requests de la prueba.), se desea saber cuántas conexiones máximas se pueden tener en simultaneo, tal que se cumplan las restricciones establecidas.

Ensayo con 420 hilos: Se valida que con 420 hilos sigue funcionando cumpliendo el estándar.



Label	# Samples	Average	Min	Max	Std. Dev.	Error %	Throughput	Received KB/sec	Sent KB/sec	Avg. Bytes
HTTP Request	420	1475	377	4299	647.37	0.00%	86.7/sec	3858.11	11.86	45546.0
TOTAL	420	1475	377	4299	647.37	0.00%	86.7/sec	3858.11	11.86	45546.0

Ensayo con 421 hilos: Se valida que a partir de 421 hilos deja de cumplir el estándar.

Summary Report

Name: Summary Report

Comentarios:

Write results to file / Read from file

Filename: Browse... Log/Display Only: ☐ Errors ☐ Successes ☐ Configure

Label	# Samples	Average	Min	Max	Std. Dev.	Error %	Throughput	Received KB/sec	Sent KB/sec	Avg. Bytes
HTTP Request	421	1533	360	4334	676.32	0.00%	84.8/sec	3773.77	11.60	45546.0
TOTAL	421	1533	360	4334	676.32	0.00%	84.8/sec	3773.77	11.60	45546.0

Reporte resumen

Nombre: Summary Report

Comentarios:

Escribir todos los datos a Archivo

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

Etiqueta	# Muestras	Media	Min	Máx	Desv. Estándar	% Error	Rendimiento	Kb/sec
HTTP Request	79	22251	11087	29711	3878,80	0,00%	2,6/sec	44995,54
Total	79	22251	11087	29711	3878,80	0,00%	2,6/sec	44995,54

2.6 Escenario 2

Crear una tarea.

Se prueba enviar y procesar un archivo, se busca saber cuántos archivos máximo se pueden procesar en un tiempo menos a 600 segundos y con 0% de error que nos indica que se procese correctamente

Ensayo con 70 hilos: Se valida que con 70 hilos sigue funcionando cumpliendo el estándar.

Summary Report

Name: Summary Report

Comentarios:

Write results to file / Read from file

Filename: Browse... Log/Display Only: ☐ Errors ☐ Successes ☐ Configure

Label	# Samples	Average	Min	Max	Std. Dev.	Error %	Throughput	Received KB/sec	Sent KB/sec	Avg. Bytes
HTTP Request	70	16891	4931	28903	7618.70	0.00%	2.4/sec	0.68	41751.54	291.0
TOTAL	70	16891	4931	28903	7618.79	0.00%	2.4/sec	0.68	41751.54	291.0

Ensayo con 85 hilos: Se valida que entre el rango de 75-85 se empiezan a generar errores y no cumple el estándar.

Label	# Samples	Average	Min	Max	Std. Dev.	Error %	Throughput	Received KB/sec	Sent KB/sec	Avg. Bytes
HTTP Request	85	25029	8223	41634	8536.09	32.94%	2.0/sec	2.31	23690.26	1167.9
TOTAL	85	25029	8223	41634	8536.09	32.94%	2.0/sec	2.31	23690.26	1167.9

2.7 Escenario 4

Hacer Log-in.

Ensayo con 1550 hilos: Se valida que con 1550 hilos sigue funcionando cumpliendo el estándar.

Summary Report

Name: Summary Report

Comentarios:

Escribir todos los datos a Archivo

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

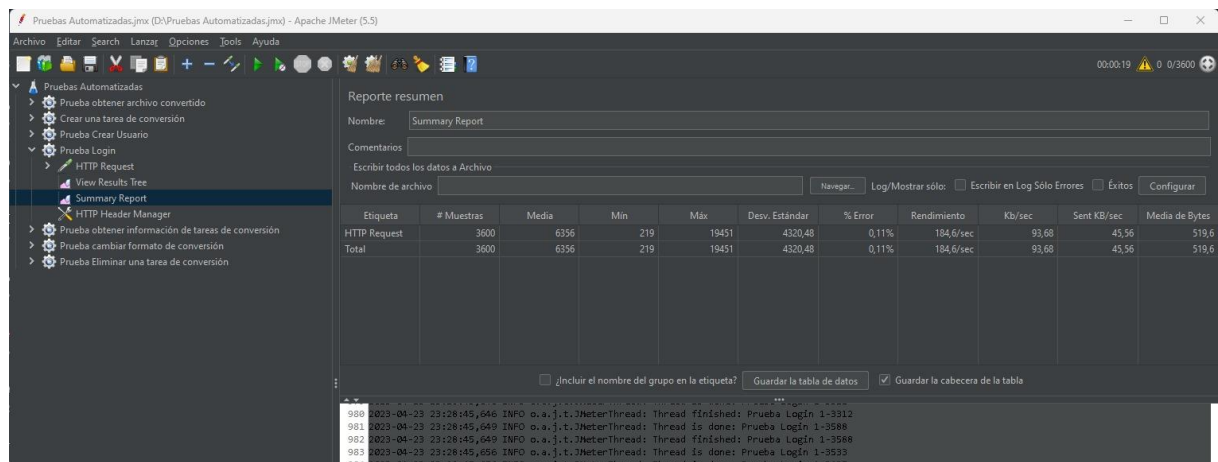
Etiqueta	# Muestras	Media	Min	Máx	Desv. Estándar	% Error	Rendimiento	Kb/sec	Sent KB/sec	Media de Bytes
HTTP Request	3550	6528	216	15454	4505.82	0.00%	210.9/sec	106,90	52,12	517,0
Total	3550	6528	216	15454	4505,82	0,00%	210,9/sec	106,90	52,12	517,0

☐ ¿Incluir el nombre del grupo en la etiqueta? ☒ Guardar la cabecera de la tabla

```

080 2023-04-23 23:27:59,055 INFO o.a.j.t.JMeterThread: Thread finished: Prueba Login 1-3513
081 2023-04-23 23:27:59,055 INFO o.a.j.t.JMeterThread: Thread is done: Prueba Login 1-3481
082 2023-04-23 23:27:59,055 INFO o.a.j.t.JMeterThread: Thread finished: Prueba Login 1-3481
  
```

Ensayo con 3600 hilos: Se valida que con 3600 hilos se empiezan a generar errores y no cumple el estándar.



Reporte resumen

Nombre: Summary Report

Comentarios:

Escribir todos los datos a Archivo

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

Etiqueta	# Muestras	Media	Min	Máx	Desv. Estándar	% Error	Rendimiento	Kb/sec	Sent KB/sec	Media de Bytes
HTTP Request	3600	6356	219	19451	4320,48	0,11%	184,0/sec	93,68	45,56	519,6
Total	3600	6356	219	19451	4320,48	0,11%	184,0/sec	93,68	45,56	519,6

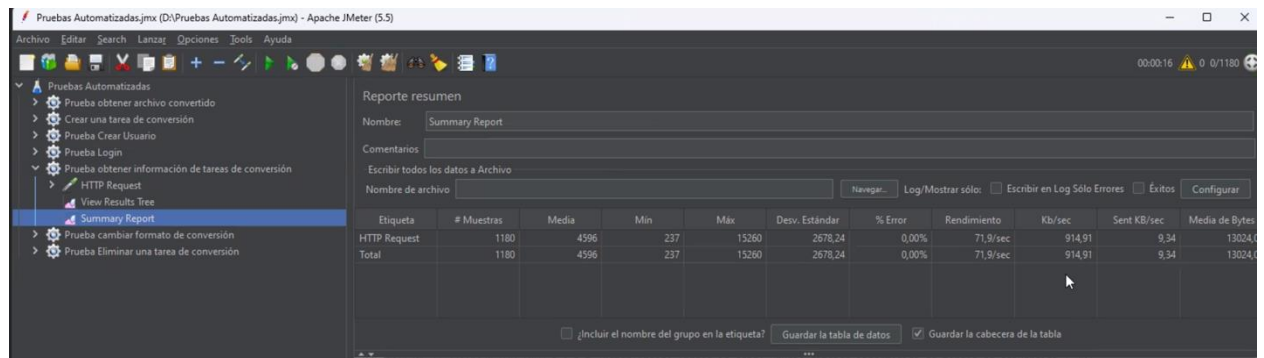
☐ ¿Incluir el nombre del grupo en la etiqueta? ☒ Guardar la cabecera de la tabla

Log: 980 2023-04-23 23:28:45,646 INFO o.a.j.t.JMeterThread: Thread finished: Prueba Login 1-3312
981 2023-04-23 23:28:45,649 INFO o.a.j.t.JMeterThread: Thread is done: Prueba Login 1-3588
982 2023-04-23 23:28:45,649 INFO o.a.j.t.JMeterThread: Thread finished: Prueba Login 1-3588
983 2023-04-23 23:28:45,656 INFO o.a.j.t.JMeterThread: Thread is done: Prueba Login 1-3533

2.8 Escenario 5

Obtener información de una tarea de conversión.

Ensayo con 1180 hilos: Se valida que con 1180 hilos sigue funcionando cumpliendo el estándar.



Reporte resumen

Nombre: Summary Report

Comentarios:

Escribir todos los datos a Archivo

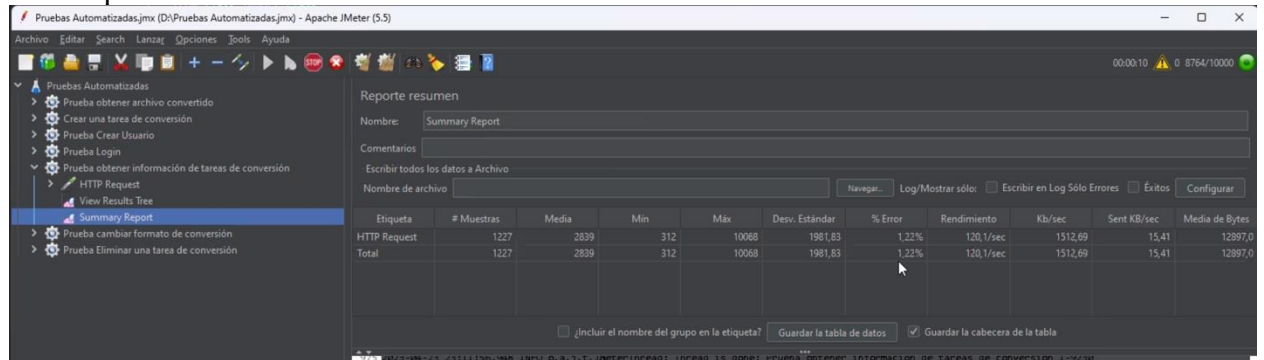
Nombre de archivo: Navegar... Log/Mostrar sólo: ☐ Escribir en Log Sólo Errores ☐ Éxitos ☐ Configurar

Etiqueta	# Muestras	Media	Min	Máx	Desv. Estándar	% Error	Rendimiento	Kb/sec	Sent KB/sec	Media de Bytes
HTTP Request	1180	4596	237	15260	2678,24	0,00%	71,9/sec	914,91	9,34	13024,0
Total	1180	4596	237	15260	2678,24	0,00%	71,9/sec	914,91	9,34	13024,0

☐ ¿Incluir el nombre del grupo en la etiqueta? ☒ Guardar la cabecera de la tabla

Log: 977 2023-04-23 23:28:45,646 INFO o.a.j.t.JMeterThread: Thread is done: Prueba Obtener información de tareas de conversión 1-3000

Ensayo con 1227 hilos: Se valida que entre el rango de 1227 se empiezan a generar errores y no cumple el estándar.



Reporte resumen

Nombre: Summary Report

Comentarios:

Escribir todos los datos a Archivo

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

Etiqueta	# Muestras	Media	Min	Máx	Desv. Estándar	% Error	Rendimiento	Kb/sec	Sent KB/sec	Media de Bytes
HTTP Request	1227	2839	312	10068	1981,83	1,22%	120,1/sec	1512,69	15,41	12897,0
Total	1227	2839	312	10068	1981,83	1,22%	120,1/sec	1512,69	15,41	12897,0

☐ ¿Incluir el nombre del grupo en la etiqueta? ☒ Guardar la cabecera de la tabla

Log: 977 2023-04-23 23:28:45,646 INFO o.a.j.t.JMeterThread: Thread is done: Prueba Obtener información de tareas de conversión 1-3000

2.9 Escenario 6

Cambiar formato de conversión.

Ensayo con 600 hilos: Se valida que con 600 hilos sigue funcionando cumpliendo el estándar.

Reporte resumen

Nombre: Summary Report

Comentarios:

Escribir todos los datos a Archivo

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

Etiqueta	# Muestras	Media	Min	Máx	Desv. Estándar	% Error	Rendimiento	Kb/sec	Sent KB/sec	Media de Bytes
HTTP Request	600	4960	247	9922	2778,57	0,00%	56,0/sec	21,72	11,32	397,0
Total	600	4960	247	9922	2778,57	0,00%	56,0/sec	21,72	11,32	397,0

☐ ¡Incluir el nombre del grupo en la etiqueta? ☒ Guardar la cabecera de la tabla

Ensayo con 652 hilos: Se valida que entre el rango de 652 se empiezan a generar errores y no cumple el estándar.

Reporte resumen

Nombre: Summary Report

Comentarios:

Escribir todos los datos a Archivo

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

Etiqueta	# Muestras	Media	Min	Máx	Desv. Estándar	% Error	Rendimiento	Kb/sec	Sent KB/sec	Media de Bytes
HTTP Request	652	5309	548	10633	2979,46	6,13%	57,6/sec	30,05	10,93	534,1
Total	652	5309	548	10633	2979,46	6,13%	57,6/sec	30,05	10,93	534,1

☐ ¡Incluir el nombre del grupo en la etiqueta? ☒ Guardar la cabecera de la tabla

2.10 Escenario 7

Eliminar una tarea de conversión.

Ensayo con 3400 hilos: Se valida que con 3400 hilos sigue funcionando cumpliendo el estándar.

Reporte resumen

Nombre: Summary Report

Comentarios:

Escribir todos los datos a Archivo

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

Etiqueta	# Muestras	Media	Min	Máx	Desv. Estándar	% Error	Rendimiento	Kb/sec	Sent KB/sec	Media de Bytes
HTTP Request	3400	5313	232	15441	3910,47	0,00%	202,5/sec	47,86	42,32	242,0
Total	3400	5313	232	15441	3910,47	0,00%	202,5/sec	47,86	42,32	242,0

Ensayo con 3545 hilos: Se valida que entre el rango de 3545 se empiezan a generar errores y no cumple el estándar.

Reporte resumen

Nombre: Summary Report

Comentarios:

Escribir todos los datos a Archivo

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

Etiqueta	# Muestras	Media	Min	Máx	Desv. Estándar	% Error	Rendimiento	Kb/sec	Sent KB/sec	Media de Bytes
HTTP Request	3545	6686	252	13965	3659.82	0.48%	246.7/sec	61.07	\$1.32	253.5
Total	3545	6686	252	13965	3659.82	0.48%	246.7/sec	61.07	\$1.32	253.5

☐ ¡Incluir el nombre del grupo en la etiqueta? ☒ Guardar la cabecera de la tabla

```

975 2023-04-23 23:04:02,358 INFO o.a.j.t.JMeterThread: Thread is done: Prueba Eliminar una tarea de conversión 1-2691
976 2023-04-23 23:04:02,358 INFO o.a.j.t.JMeterThread: Thread finished: Prueba Eliminar una tarea de conversión 1-2691
977 2023-04-23 23:04:02,358 INFO o.a.j.t.JMeterThread: Thread is done: Prueba Eliminar una tarea de conversión 1-5394
978 2023-04-23 23:04:02,358 INFO o.a.j.t.JMeterThread: Thread finished: Prueba Eliminar una tarea de conversión 1-5394
979

```

3. Documentación Postman

En el siguiente link se encuentra la documentación de las pruebas realizadas en Postman.

<https://documenter.getpostman.com/view/19568698/2s93Y5PKYM>