



UNIVERSIDAD  
**NACIONAL**  
DE COLOMBIA

### **Severe Trip**

#### **Estudiantes:**

Daniel Alfonso Pinzon Chavarria  
Alex Jose Alberto Barreto Cajicá  
Katherine Viviana Sierra Briñez  
Alberto Nicolai Romero Martinez  
Julian David Rodriguez Ruiz  
Verbo Sebastian Camacho Silva

## Índice

<b>Introducción</b>	<b>3</b>
<b>identificación de historias de usuario.</b>	<b>4</b>
<b>3. Identificación y distribución y asignación de tareas según su prioridad</b>	<b>6</b>
<b>4. Acuerdo de cuales son las tareas se van a entregar en el Sprint</b>	<b>6</b>
<b>Registro de los daily Sprint #5</b>	<b>10</b>
<b>Sprint retrospective</b>	<b>13</b>
<b>Sprint Review #5</b>	<b>16</b>
<b>b. Gestión de calidad</b>	<b>18</b>

## a. Gestión del proyecto

Desde una metodología ágil (SCRUM) el Scrum Master debe especificar para este sprint:

### 1. Introducción:

El sprint planning se usa de manera recurrente en cada uno de los sprint para pulir el backlog entre el product owner y el scrum master, eso es importante para asignar prioridades o nivel de dificultad de las tareas que se van a asignar y como se va a planificar el transcurso del sprint.

De esta manera vamos a aplicar de manera estricta cada una de las normas estipuladas dentro de la metodología Scrum, teniendo en cuenta un sistema de prioridades de tareas derivadas de las historias de usuario que se acuerden y las capacidades del grupo de trabajo para cumplir a cabalidad las tareas que se estipulan, según la dificultad y el nivel del grupo de trabajo.

Para esta iteración se ha decidido terminar requisitos pendientes de la primera iteración y evaluar los alcances de la segunda. Lo que tenemos en nuestra lista es lo siguiente:

ítems sugeridos por el Product owner sprint #1 (pendientes):

- 1) despliegue de todos los microservicios

ítems sugeridos por el Product owner sprint #2:

- 1) El sistema de software debe poseer un Front-End, constituido por dos tipos de clientes diferentes.
  - a) El primer cliente debe estar construido como un componente, tipo aplicación web, en cualquier lenguaje de programación y framework (si aplica).
  - b) El segundo cliente, tipo aplicación móvil, en cualquier lenguaje de programación y framework (si aplica). Puede obtenerse a través de:
    - i) Un diseño responsive o
    - ii) Una aplicación móvil nativa o
    - iii) Una aplicación híbrida.
- 2) El cliente web debe estar dockerizado.
- 3) Los dos clientes deben comunicarse con el Back-End únicamente mediante el API Gateway.
- 4) Una de las bases de datos de los microservicios debe ser noSQL
- 5) Las vistas de arquitectura deben ser construidas con Visual Paradigm
- 6) La organización y seguimiento del Sprint se debe hacer a través de una herramienta de gestión de proyectos

## 2. identificación de historias de usuario.

Con la información que nos suministro el product owner podemos empezar a crear nuestras historias de usuario de acuerdo a la funcionalidad estipulada.

Para esto se tendrán en cuenta 3 cosas desde el punto de vista del usuario de la aplicación:

1. **Tipo de usuario:** Clase de usuario que va a acceder.
2. **Funcionalidad:** Funcionalidad
3. **Solución:** Lo que resuelve.

En nuestro proyecto hemos identificado las siguientes historias de usuario (backlog) según el equipo de desarrollo:

1. Daniel. (Scrum master)
2. Sebastian.
3. Alex.
4. Nicolai.
5. Julian.
6. Katherine. (Tester)

De acuerdo a los ítems propuestos vamos a identificar en primer lugar las vistas importantes del **Front-End** para el desarrollo de la parte visual del proyecto. Cabe tener en cuenta que todas las vistas irán conectadas a API Gateway para que puedan acceder a los distintos servicios del **Back-End**, por lo que cada vista necesariamente debe ir con su conexión y será verificada de acuerdo a la importancia y complejidad evaluada por el equipo de desarrollo.

A continuación vamos a ver el recopilado del backlog y su respectiva evaluación:

id	Responsable	Tipo de usuario	Solución	Descripción	1	2	3	4	5	6	total
t1	Katherine, Julián	-	Prueba de rendimiento antes de escalar	Pruebas de carga y estrés con única instancia por componente.	1	1	2	3	1	1	1,5

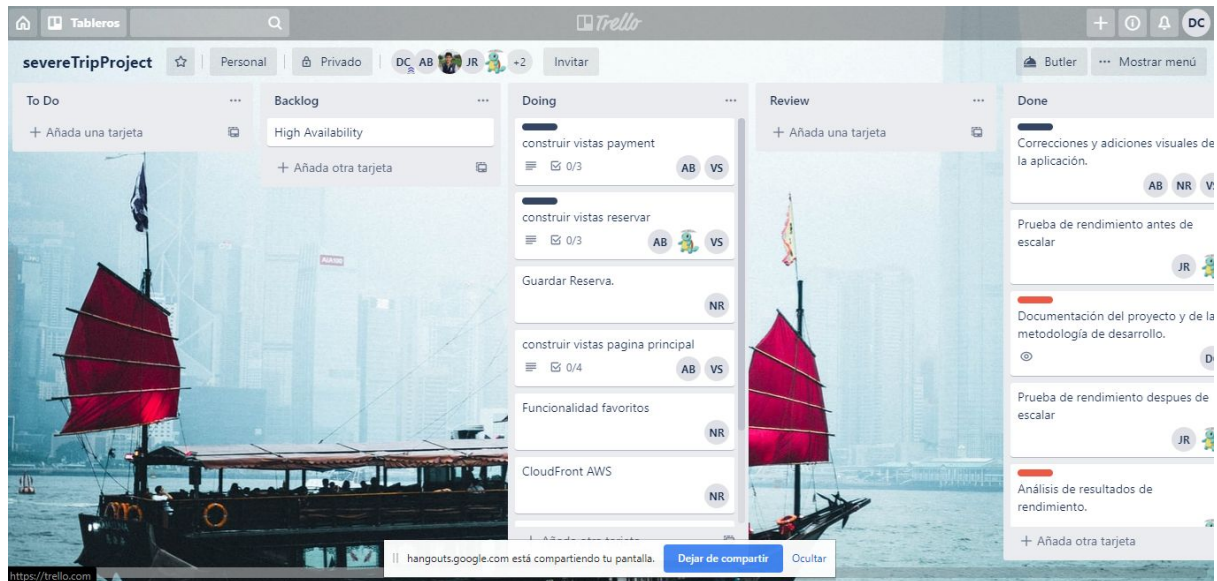
t2	Katherine , Julián	-	Prueba de rendimiento <b>despues</b> de escalar	Pruebas de rendimiento con instancias multiples por componente, con escalamiento horizontal.	2	2	1	2	2	2	1,83
t3	Katherine	-	Análisis de resultados de rendimiento.	Gráficas de rendimiento. Identificación de pico de carga.	3	3	3	1	3	3	2,67
f1	Nicolai, Sebastián, Alex	Cliente Proveedor	Correcciones y adiciones visuales de la aplicación.	Adición de características, corrección de bugs, cambios de UI.	5	5	5	5	5	5	5
a1	Daniel	-	Documentación del proyecto y de la metodología de desarrollo.	Documento de Descripción de Arquitectura, Actividades de SCRUM.	4	4	4	4	4	4	4

Como la recolección del backlog en su máxima expresión, se requiere hacer un análisis de cada vista con el product owner, esté a su vez dejó estipulada una funcionalidad desde el principio, el evaluador del orden de prioridad de las tareas va a ir ligado a la perspectiva del grupo de trabajo teniendo en cuenta aspectos como dificultad y prioridad, donde ellos van a evaluar esto.

### 3. Identificación y distribución y asignación de tareas según su prioridad

Habiendo culminado esto, sabemos cuales son las tareas y su prioridad, después consultamos cómo podríamos distribuir estas tareas dentro del equipo de desarrollo y para esto decidimos no tener una asignación puntual de tareas, aunque sí les dimos prioridad dentro de la funcionalidad de cada uno de los microservicios.

Luego de esto montamos las las tareas como una composición de los microservicios dentro del tablero de trabajo (Trello), en donde cada una de las tareas son parte de los check points que se pueden programar dentro de la plataforma, esto se pensó con el fin de darle libertad al equipo de desarrollo de autogestionar sus propios deberes siempre teniendo en cuenta la prioridad que se le dio a cada una de las vistas de dónde vienen estas tareas.



Esto también ayuda a tener una mejor disposición dentro de los repositorios, haciendo que cada uno de estos tengan únicamente 2, un branch developer en donde todos tienen la participación y la actualización en cada push previamente anunciado, y el master, en el que solo se harán commits el día del sprint.

De ahora en adelante se usará otro repositorio en donde se guarde el **Front-End**, en donde se guardaran todas las vistas.

#### 4. Acuerdo de cuales son las tareas se van a entregar en el Sprint

En este numeral vamos a evaluar una nueva variable, el tiempo de desarrollo destinado para las tareas que se estipulan, entonces se van a revisar las tareas y se informará hasta donde llegara el desarrollo de este sprint.

1. Se van a entregar las vistas con diseño responsive enfocado a web.
2. Se creará un orquestador de peticiones al **Back-End** (API Gateway).
3. Se agregara la siguiente funcionalidad pendiente:
  - Mensajería
  - Creacion de imagenes y contenedores en docker
  - Documentación completa (descripción y pruebas)
4. Microservicio LDAP
5. JWT para autenticación
6. Pruebas de carga y estrés antes y despues en cada microservicio
7. Escalamiento de las bases de datos MySQL como master-slave

#### PENDIENTE

Para la revisión, entrar a los siguientes repositorios:

#### Back-End

<https://github.com/daapinzonch/tripAdvisorPayment>  
<https://github.com/daapinzonch/tripAdvisorProvider>  
<https://github.com/daapinzonch/tripAdvisorClient>  
<https://github.com/daapinzonch/tripAdvisorReservation>  
<https://github.com/daapinzonch/tripAdvisorPostProvider>  
<https://github.com/daapinzonch/tripAdvisorFavorites>  
<https://github.com/daapinzonch/tripAdvisorMessage>

## **Front-End y Gateway**

<https://github.com/trip-Advisor-Project-Arquisoft/FrontEndTripAdvisor>  
<https://github.com/trip-Advisor-Project-Arquisoft/FrontEndSevereTrip>  
<https://github.com/trip-Advisor-Project-Arquisoft/API-Gateway>

## **Sprint # 3**

Implementación LDAP  
Implementación Token, JWT  
Implementación DNS

## **Sprint #4**

Balanceador de carga  
Escalamiento horizontal  
Análisis pruebas de carga y estrés

## **Sprint #5**

Escalamiento de las bases de datos (master-master, master-slave)  
Revisión de errores en el Front-End

### **1. Daily scrum**

A continuación se presentarán de manera ordenada una síntesis de los daily diseñados para el primer sprint. Estos se harán entre semana los días Lunes, Miércoles y viernes, los fines de semana los días Sábados y domingos y una leve reunión al salir de clases los días Martes y Jueves (de estos se tendrán registros solo cuando sea pertinente ya que son reuniones de emergencia).

Las presentación se dará de la siguiente manera:

1. Fecha, día, intervalo de tiempo.
2. participantes (las personas subrayadas con resaltador amarillo son las que no estuvieron presentes en el daily)

3. Charla: Aquí es donde se sacan las cosas más importantes como tareas para hacer hasta el próximo daily, además de anotar información importante, las tareas más importantes se registraron en el Trello como tareas no funcionales.

### **Registro de los daily Sprint #5**

#### **Sprint #5**

##### **- 16/1/2020 Miércoles 9:00 pm - 10:00 pm**

participantes: Daniel (Scrum master), Sebastian, Alex, Nicolai, Julian, katherine (tester)

charla:

1. Alex terminar de conectar reservation al front
2. Sebastián corregirá término de cesión de la página
3. Nicolai va a revisar errores en la reservación
4. Julián va a escalar las bases de datos del docker
5. Daniel actualizará documentos de entrega ADD e inf
6. Kathe colaborará con extras del front

##### **- 16/1/2020 Miércoles 9:00 pm - 10:00 pm**

participantes: Daniel (Scrum master), Sebastian, Alex, Nicolai, Julian, katherine (tester)

charla:

1. Alex avance en deslogueo de la página
2. Sebastián colaboración en búsqueda de soluciones a problema del logueo
3. Nicolai revisión de problema de comunicación de reservas entre back-end y front-end
4. Julián redesplicue de back-end con el escalamiento de las bases de datos
5. Daniel corrección de vista de despliegue con lo hecho en las bases de datos
6. Kathe revisión de funcionalidades criticas despues del ultimo despliegue

#### **Sprint retrospective**

A continuación vamos a describir brevemente nuestros principales fallos y aciertos durante el Sprint #2 y ofrecemos posibles soluciones y mejoras para mitigar estos predicamentos para la siguiente entrega.

##### **1. Qué ha funcionado bien en el último Sprint.**

1. Comunicación frecuente (fines de semana) para planeación.
2. La elección del framework (Spring boot).
3. Documento de herramientas y ayuda.



4. Disposición del grupo para aprender cosas nuevas.
  5. Asignación de tareas (actualización constante del backlog).
  6. Documentación al día y participativa.
- 2. Cuáles cosas hay que mejorar de cara al siguiente Sprint.**
1. Hacer la asignación de tareas con anticipación.
  2. Mejorar tiempos del sprint planning.
  3. Diagramación a detalle de cada microservicio.
  4. Diseño de mockups para saber cómo planear el backend.
  5. Tener mejor contacto con el product owner.
- 3. Problemas que haya tenido para poder progresar correctamente en el último Sprint.**
1. El tiempo tan corto para el desarrollo de algo tan robusto como el backend.
  2. Problemas iniciales de diagramación (planeación).
  3. Aprendizaje de una nueva herramienta (Spring boot).
- 4. Recomendaciones a aplicar en el siguiente Sprint.**
1. Planeación del tiempo en los daily para mejorar el rendimiento.
  2. Diseñar antes que implementar
  3. Medir mejor la disposición dentro del sprint.
- 5. Riesgos.**
1. Tiempo disponible para el Sprint
  2. Nuevas herramientas y aprendizaje
  3. Carga estudiantil
  4. Integrantes del equipo que trabajan
  5. Funcionamiento en Cloud
- 5. Retroalimentación del Product Owner**
1. Hubo un error en la planeación al no incluir los diseños de las vistas desde el principio o al menos desde que se hizo el análisis de las historias de usuario, lo cual se intentó mitigar inmediatamente.
  2. Hubo una planeación desigual en cuanto al tiempo de dedicación a la parte de programación y a la parte de documentación, haciendo que el enfoque haya afectado posiblemente la entrega del primer sprint a cabalidad.
  3. Los Daily si bien están bien distribuidos, el Product owner sugiere que se hagan todos los días, a lo que el grupo acepta disponer de más tiempo para estas reuniones pero no de la forma en que el Product owner sugirió puesto que todos tienen más obligaciones.
  4. Se solucionan dudas acerca de cómo resolver algunos numerales del proyecto teniendo en cuenta que cosas como el Sprint review y el Sprint retrospective deberían ir finalizando cada sprint.
  5. Se solucionan dudas acerca de cómo disponer de los modelos en la documentación.
  6. Se solucionan dudas acerca de si se podía entregar más documentación o con lo exigido estaba bien (los dos documentos).

7. Se solicitó una guía para el despliegue de los distintos microservicios, además se mostró un avance importante tanto a nivel de desarrollo como a nivel de documentación.

- **Calificación del Sprint #5**

Integrante	Calificación
Nicolai	Se involucro y hubo compromiso con las tareas asignadas, cumpliendo.
Julian	Realizo todas las tareas asignadas en el tiempo correspondiente del Sprint, colaboró resolviendo dudas a los demás integrantes del equipo. Se sobre exige en el despliegue, para cumplir.
Sebastian	Aprendió en tiempo record para cumplir a cabalidad con las tareas asignadas, hubo compromiso a pesar de otras responsabilidades que limitaron su tiempo disponible.
Alex	Apoyo las tareas del Sprint, cumpliendo sus responsabilidad y ayudando a los demás, hubo mucho compromiso de parte suya.
Katherine	A pesar de la limitante de tiempo cumplio a cabalidad con la revision y el control de calidad de cada uno de los servicios desplegados.
Daniel	Estuvo pendiente del equipo, ayudó a la buena comunicación entre todos incluyendo al product owner, realizó la asignación de tareas, motivando y liderando al equipo.

## 2. Sprint review

### Sprint Review #2

En el Sprint review se dejará anotado todo el avance de las tareas acordadas en el sprint planning y su estado.

En verde estarán las tareas que se realizaron y revisaron al 100%, en amarillo las tareas sin completar y en rojo las que no se realizaron.

id	Tipo de usuario	Responsable	Funcionalidad	Tareas - Microservicios involucrados
v1	Cliente Proveedor	Alex, Sebastian, Nicolai	Página principal de la página.	- Ver todas las publicaciones - Post - Ver favoritos (login) Favoritos - Información de perfil de usuario - Cliente
v2	Cliente	Alex, Sebastian, Nicolai	Ver publicaciones de los proveedores.	- Ver proveedor - Proveedor - información de usuario - Cliente - post - Post - Comentarios y calificación. - Post - precio - Post
v3	Cliente Proveedor	Alex, Sebastian, Nicolai	Reserva-Payment	- Calendario de fechas - Reserva - Cantidad de personas - Reserva - Información proveedor - Proveedor - Valor a pagar - Reserva - Fecha de pago - Payment - Estado del pago - Payment
v4	Cliente Proveedor	Alex, Sebastian, Nicolai	Perfil de usuario de acuerdo a su rol.	- Perfil del cliente - Cliente - Perfil de proveedor - Proveedor - Mensajería - Cliente - Crear publicación. - Proveedor.
gw	General	Julian	Orquestador de peticiones y servicios.	- Redireccionar peticiones.
t	Tester	Katherine	Gestión de calidad.	- Peticiones de postman del gateway. - Validación de vistas.
sm	Coordinación y documentación	Daniel	Metodología.	-Correcciones y adiciones al documento de descripción de arquitectura. -Correcciones de vistas.

				-Tareas del Front.
--	--	--	--	--------------------

Adicional a estas tareas, surgieron tareas de contingencia para mitigar y/o corregir errores. Estas son:

1. Redespliegue - Julian, Nicolai
2. CORS - Julian
3. Cambio de fetch-axios - Nicolai

## Sprint Review #3

### A noviembre 7:

Qué se ha hecho:

- Agrupación de microservicios de Cliente, Proveedor y Favoritos en nuevo microservicio: Usuario
- Modificación de API Gateway para soportar la nueva estructura de los microservicios
- Adición de dependencias de Spring Security y Spring JWT a microservicio de Usuario
- Selección de las herramientas y tecnologías a utilizar para la implementación del LDAP y el microservicio de autenticación.

Qué impedimentos hay:

- Tiempo por parte del equipo
- Anormalidad académica

Qué falta por hacer:

- Implementar el LDAP
- Implementar soporte para JWT en el Frontend
- Implementar manejo de JWT y microservicio de autenticación en el Backend

id	Tipo de usuario	Solución	Funcionalidad
v1	Cliente Proveedor	JWT implementado desde Back-End	generador y verificador de tokens de inicio de sesion
v2	Cliente Proveedor	JWT implementado desde Front-End	guardar y mantener actualizado el estado del token generado
v3	Cliente Proveedor	LDAP implementado desde Back-End	microservicio verificador de las credenciales suministradas por el usuario desde la interfaz

v4	Cliente Proveedor	SSL	Certificado de seguridad
v5	Cliente Proveedor	DNS	asocia un nombre de dominio a una ip (prerrequisito SSL)
v6	Cliente Proveedor	Configuración HTTPS	Protocolo de seguridad de hipertexto.
gw	General	Tener sus datos actualizados dentro de la aplicación.	Orquestador de peticiones y servicios

Las tareas v4, v5 y v6 son tareas propuestas desde clase y van por fuera del entregable.

## Sprint Review #4

### A Diciembre 3:

Qué se ha hecho:

- Escalamiento horizontal de todos los microservicios
- Balanceador de cargas
- Análisis gráfico de las pruebas de rendimiento (antes y despues de la optimizacion)
- Actualización de documentos
- Agregar funcionalidad al Front-End

Qué impedimentos hay:

- Tiempo por parte del equipo
- Anormalidad académica

Qué falta por hacer:

- Probar nueva funcionalidad de la pagina
- Gestionar nodos de AWS

id	Responsabl e	Tipo de usuario	Solución	Descripción
t1	Katherine, Julián	-	Prueba de rendimiento <b>antes</b> de escalar	Pruebas de carga y estrés con única instancia por componente.
t2	Katherine, Julián	-	Prueba de rendimiento <b>despues</b> de escalar	Pruebas de rendimiento con instancias multiples

				por componente, con escalamiento horizontal.
t3	Katherine	-	Análisis de resultados de rendimiento.	Gráficas de rendimiento. Identificación de pico de carga.
f1	Nicolai, Sebastián, Alex	Cliente Proveedor	Correcciones y adiciones visuales de la aplicación.	Adición de características, corrección de bugs, cambios de UI.
a1	Daniel	-	Documentación del proyecto y de la metodología de desarrollo.	Documento de Descripción de Arquitectura, Actividades de SCRUM.

#### b. Gestión de calidad

Por cada microservicio

1. Casos de prueba
2. Datos de prueba
3. Procedimiento de prueba
4. Resultados de prueba

## Sprint Review #5

### A Enero 20:

Qué se ha hecho:

- Escalamiento horizontal de todos los microservicios
- Escalamiento de las bases de datos MySQL y mongodb como master-slave y master-master
- Balanceador de cargas
- Análisis gráfico de las pruebas de rendimiento (antes y después de la optimización)
- Actualización de documentos
- Agregar funcionalidad al Front-End

Qué impedimentos hay:

- Tiempo por parte del equipo
- Anormalidad académica

Qué falta por hacer:

- Probar nueva funcionalidad de la pagina
- Gestionar nodos de AWS

id	Responsable	Tipo de usuario	Solución	Descripción
t1	Julián	-	Escalamiento de bases de datos	escalamiento de las bases de datos como maestras o esclavas
t2	Katherine, Julián	-	Prueba de rendimiento <b>despues</b> de escalar	Pruebas de rendimiento con instancias multiples por componente, con escalamiento horizontal.
t3	Alex	-	Busqueda de solucion al bad request en reservation	Solucion a problema de conexion con el microservicio reservation
f1	Nicolai, Sebastián, Alex	Cliente Proveedor	Correcciones y adiciones visuales de la aplicación.	Adición de características, corrección de bugs, cambios de UI.
a1	Daniel	-	Documentación del proyecto y de la metodología de desarrollo.	Documento de Descripción de Arquitectura, Actividades de SCRUM.

## b. Gestión de calidad

- **JMETER**

DISEÑO DE LOS CASOS DE PRUEBA	
1. Código del caso de prueba: <1 >	2. Nombre del caso de prueba: <b>Login escalabilidad</b>
3. Identificación del escenario del Caso de Uso: JMETER concurrencia 1 usuario	
● 4. Descripción de la prueba:	

- Se realiza una prueba del login de la aplicación con la concurrencia un usuario.

#### 5. Datos de entrada para ejecutar el caso de prueba:

Datos	Resultados Esperados	Salida	Pasa / Falla
<a href="http://https://www.severetrip.cf/sign-in">http://https://www.severetrip.cf/sign-in</a>	Obtener un porcentaje no mayor a 0, la aplicación de debe responder a todas las peticiones del usuario.	Obtener un porcentaje no mayor a 0, la aplicación de debe responder a todas las peticiones del usuario.	PASA

#### 6. Comentarios:

JMETER

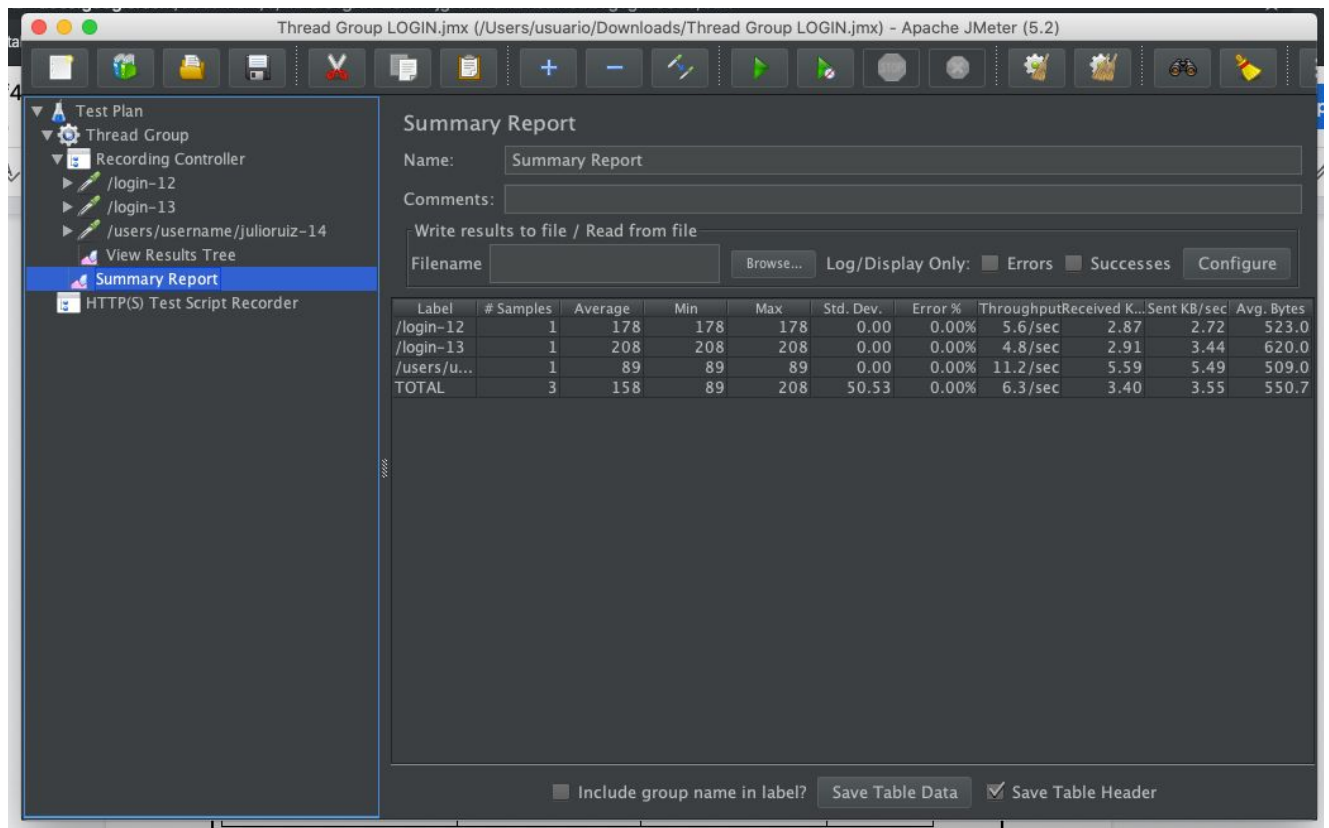
#### 7. Responsable(s) de realizar la prueba:

Katherine Sierra

	Nombre: Katherine Sierra
	Fecha: 1/10/2019
	Rol: QA

**Evidencia:**

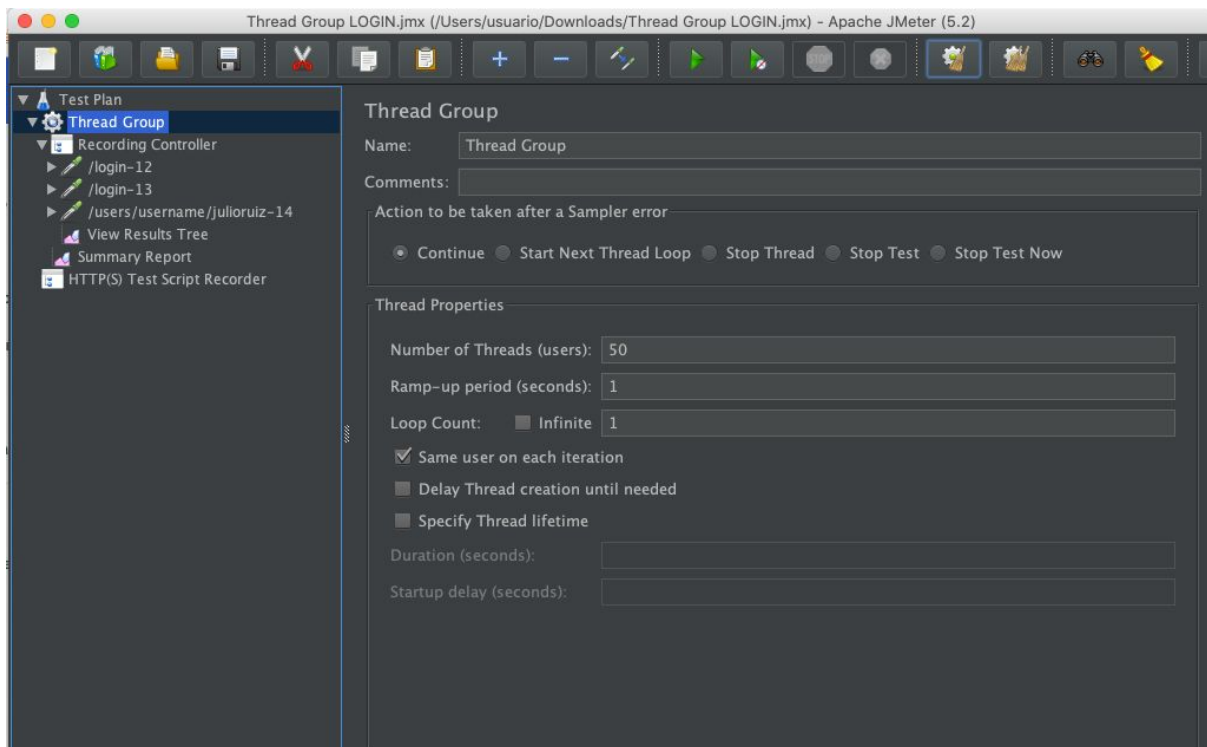


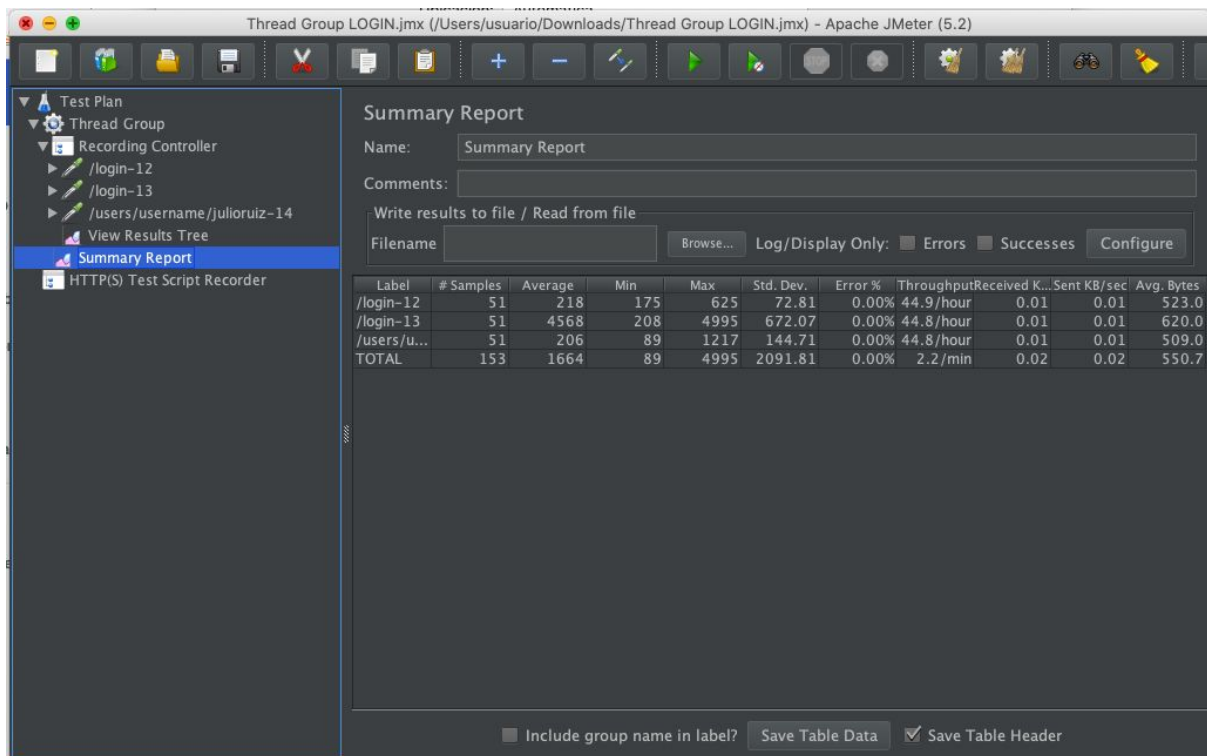


DISEÑO DE LOS CASOS DE PRUEBA			
1. Código del caso de prueba: <2 >		2. Nombre del caso de prueba: Login escalabilidad	
3. Identificación del escenario del Caso de Uso: JMETER concurrencia 50 usuarios			
4. Descripción de la prueba: <ul style="list-style-type: none"><li>Se realiza una prueba del login de la aplicación con la concurrencia un usuario.</li></ul>			
5. Datos de entrada para ejecutar el caso de prueba:			
Datos	Resultados Esperados	Salida	Pasa / Falla
http:// <a href="https://www.severetrip.cf/sign-in">https://www.severetrip.cf/sign-in</a>	Obtener un porcentaje no mayor a 0, la aplicación de debe responder a todas las peticiones del usuario.	Obtener un porcentaje no mayor a 0, la aplicación de debe responder a todas las peticiones del usuario.	PASA

<b>6. Comentarios:</b>	
<b>JMETER</b>	
<b>7. Responsable(s) de realizar la prueba:</b>	
Katherine Sierra	
	Nombre: Katherine Sierra
	Fecha: 1/10/2019
	Rol: QA

### Evidencia:





DISEÑO DE LOS CASOS DE PRUEBA			
1. Código del caso de prueba: <3 >		2. Nombre del caso de prueba: Login escalabilidad	
3. Identificación del escenario del Caso de Uso: JMETER concurrencia 50 usuarios			
● 4. Descripción de la prueba:  ● Se realiza una prueba del login de la aplicación con la concurrencia un usuario.			
5. Datos de entrada para ejecutar el caso de prueba:			
Datos	Resultados Esperados	Salida	Pasa / Falla
http://https://www.severetrip.cf/sign-in	Obtener un porcentaje no mayor a 0, la aplicación de debe responder a todas las peticiones del usuario.	Obtener un porcentaje no mayor a 0, la aplicación de debe responder a todas las peticiones del usuario.	PASA

<b>6. Comentarios:</b>	
<b>JMETER</b>	
<b>7. Responsable(s) de realizar la prueba:</b>	
Katherine Sierra	
	Nombre: Katherine Sierra
	Fecha: 1/10/2019
	Rol: QA

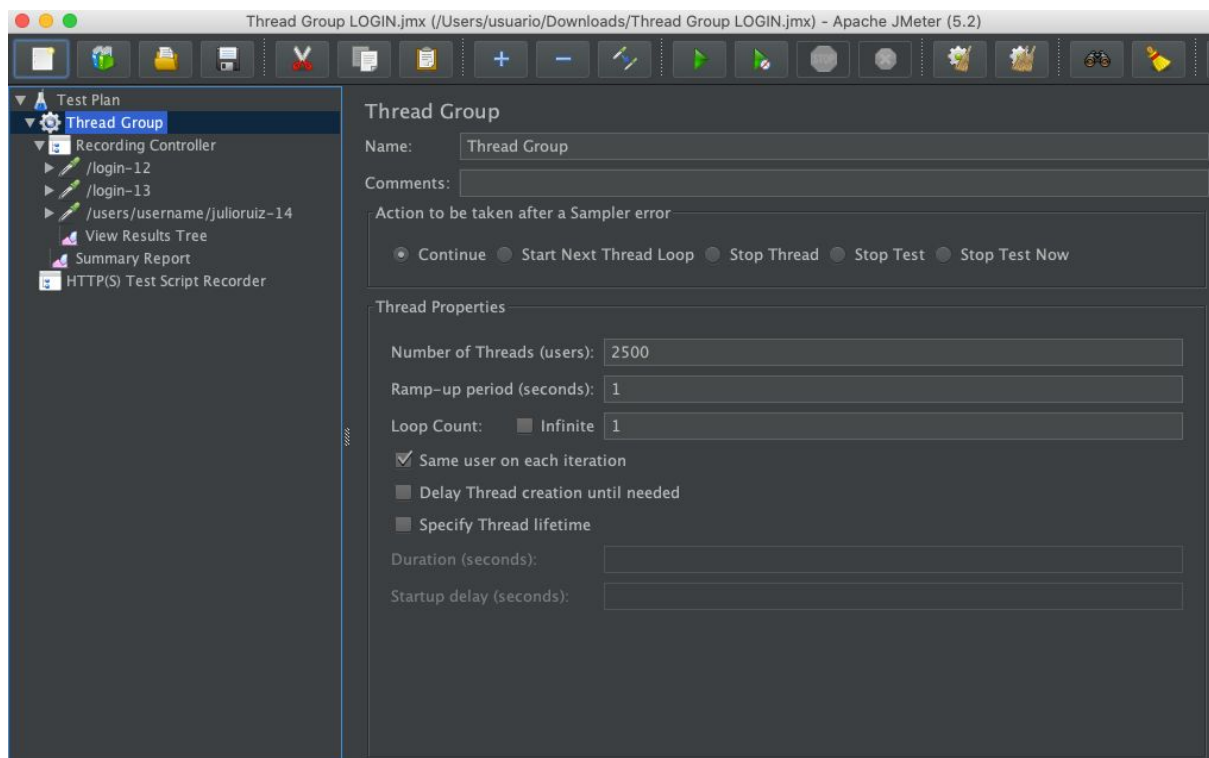
**Evidencia:**

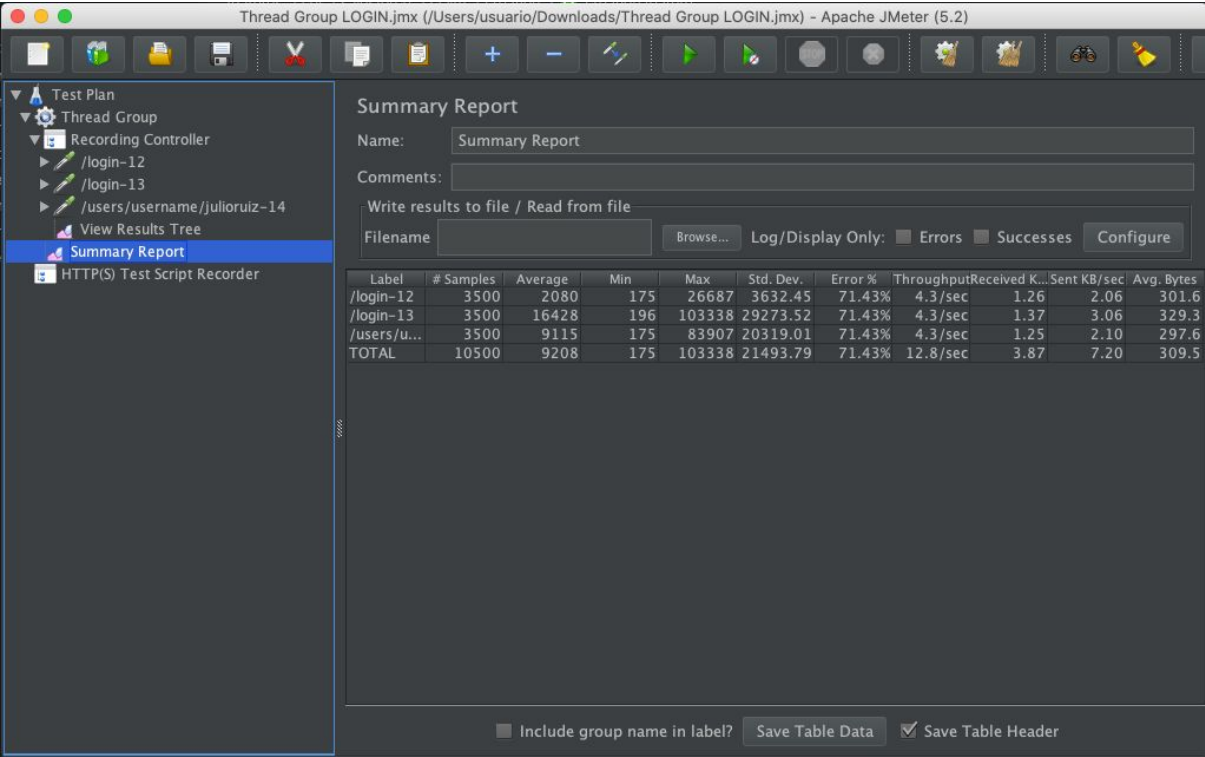
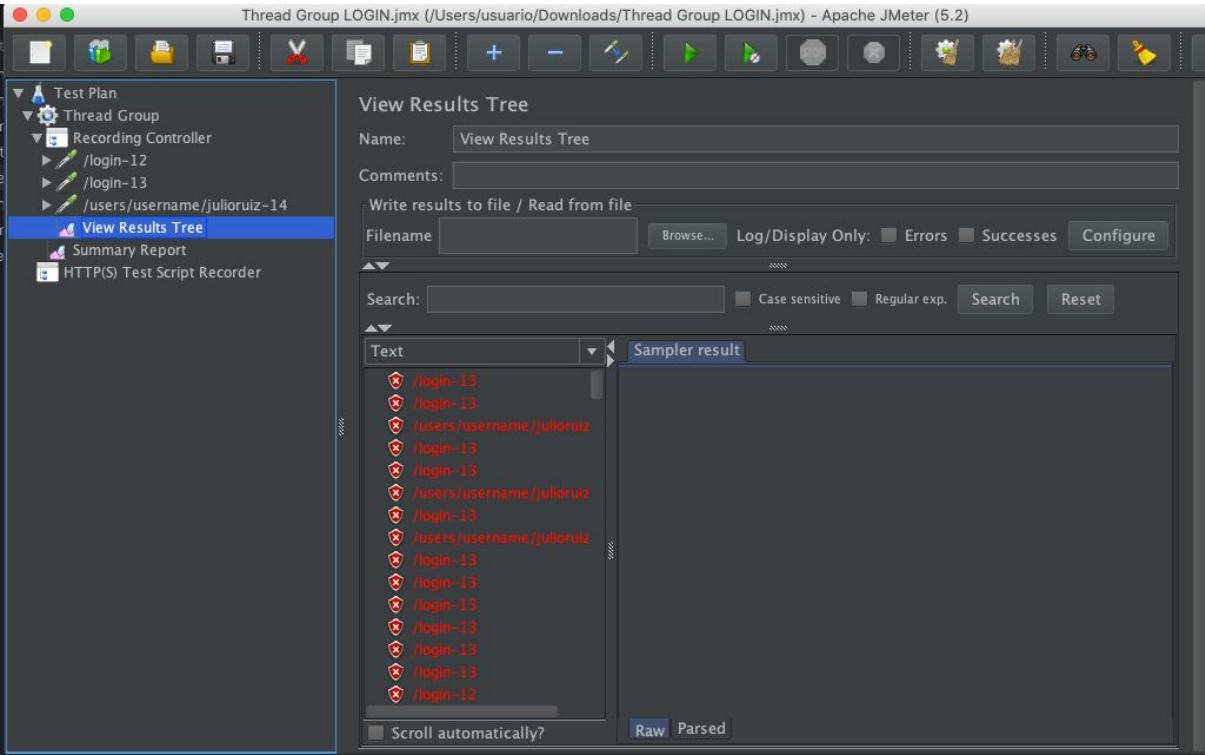
</

DISEÑO DE LOS CASOS DE PRUEBA			
1. Código del caso de prueba: <3 >		2. Nombre del caso de prueba: Login escalabilidad	
3. Identificación del escenario del Caso de Uso: JMETER concurrencia 2500 usuarios			
4. Descripción de la prueba: <ul style="list-style-type: none"> <li>Se realiza una prueba del login de la aplicación con la concurrencia un usuario.</li> </ul>			
5. Datos de entrada para ejecutar el caso de prueba:			
Datos	Resultados Esperados	Salida	Pasa / Falla

<a href="http://https://www.severetrip.cf/sign-in">http://https://www.severetrip.cf/sign-in</a>	Obtener un porcentaje no mayor a 0, la aplicación no responde todas las peticiones del usuario.	Obtener un porcentaje no mayor a 0, la aplicación no responde a todas las peticiones del usuario.	PASA
6. Comentarios:			
JMETER			
7. Responsable(s) de realizar la prueba:			
Katherine Sierra			
	Nombre: Katherine Sierra		
	Fecha: 1/10/2019		
	Rol: QA		

## Evidencia:





Gráfica:

# Usuarios	Escenario #1	
	Tiempo de Respuesta (ms)	Throughput (trans/min)
1	178	50,9
5	210	247,9
50	218	2463,1
75	432	3142,5
100	3310	1392,1
500	3850	6185,6
750	4678	7925,3
1000	6726	7766,0
2000		120000,0

- Se realizaron las pruebas después de la implementación del balanceador de carga

DISEÑO DE LOS CASOS DE PRUEBA			
1. Código del caso de prueba: <3 >		2. Nombre del caso de prueba: Post escalabilidad	
3. Identificación del escenario del Caso de Uso: JMETER concurrencia 1 usuarios			
• 4. Descripción de la prueba:  • Se realiza una prueba del login de la aplicación con la concurrencia un usuario.			
5. Datos de entrada para ejecutar el caso de prueba:			
Datos	Resultados Esperados	Salida	Pasa / Falla
http://https://www.severetrip.cf/sign-in	Obtener un porcentaje no mayor a 0, la aplicación de debe responder a todas las peticiones del usuario.	Obtener un porcentaje no mayor a 0, la aplicación de debe responder a todas las peticiones del usuario.	PASA

<b>6. Comentarios:</b>	
<b>JMETER</b>	
<b>7. Responsable(s) de realizar la prueba:</b>	
Katherine Sierra	
	Nombre: Katherine Sierra
	Fecha: 1/10/2019
	Rol: QA

## Evidencia:

Thread Group LOGIN.jmx (C:\Users\Kathe\Desktop\Thread Group LOGIN.jmx) - Apache JMeter (5.2.1)

File Edit Search Run Options Tools Help

Test Plan  
 Thread Group  
 Recording Controller  
 /login-12  
 /login-13  
 /users/username/julioruiz-14  
 View Results Tree  
**Summary Report**  
 HTTP(S) Test Script Recorder

### Summary Report

Name: Summary Report

Comments:

Write results to file / Read from file

Filename:  Browse...

Log/Display Only: ☐ Errors ☐ Successes

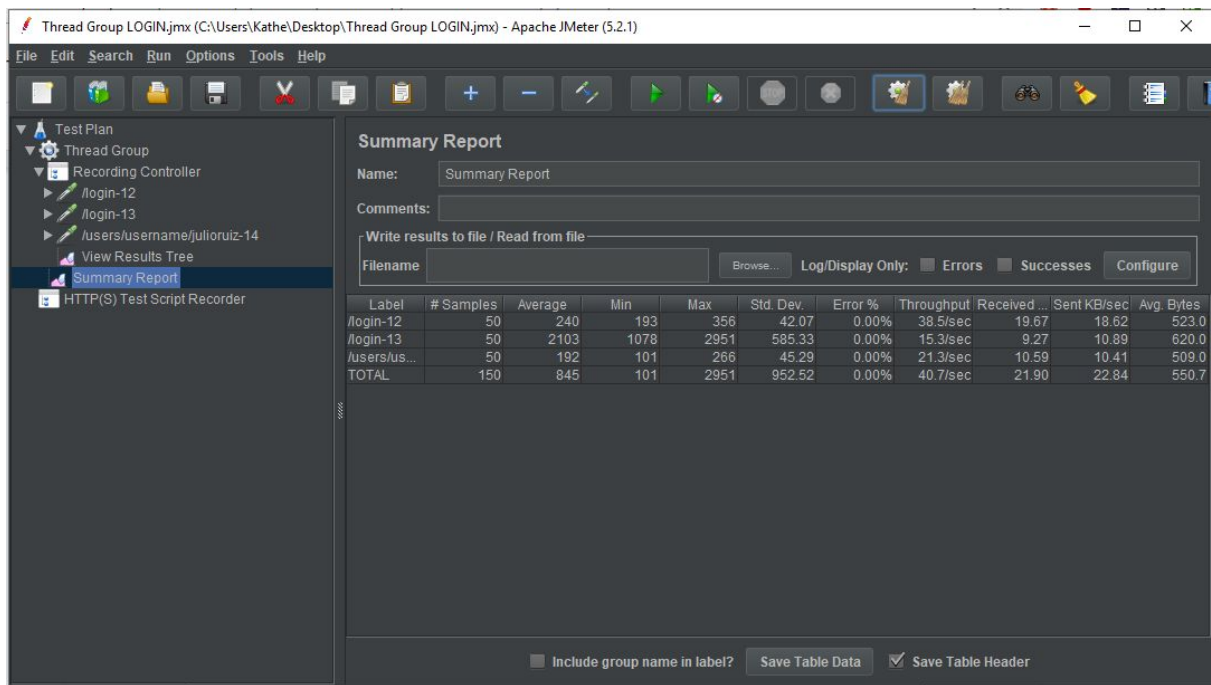
Label	# Samples	Average	Min	Max	Std. Dev.	Error %	Throughput	Received...	Sent KB/sec	Avg. Bytes
/login-12	1	195	195	195	0.00	0.00%	5.1/sec	2.62	2.48	523.0
/login-13	1	264	264	264	0.00	0.00%	3.8/sec	2.29	2.69	620.0
/users/us...	1	170	170	170	0.00	0.00%	5.9/sec	2.92	2.87	509.0
TOTAL	3	209	170	264	39.75	0.00%	4.8/sec	2.56	2.67	550.7

☐ Include group name in label?  ☒ Save Table Header



DISEÑO DE LOS CASOS DE PRUEBA			
1. Código del caso de prueba: <3 >		2. Nombre del caso de prueba: <b>Post escalabilidad</b>	
3. Identificación del escenario del Caso de Uso: JMETER concurrencia 50 usuarios			
4. Descripción de la prueba: <ul style="list-style-type: none"> <li>Se realiza una prueba del login de la aplicación con la concurrencia un usuario.</li> </ul>			
5. Datos de entrada para ejecutar el caso de prueba:			
Datos	Resultados Esperados	Salida	Pasa / Falla
<a href="http://https://www.severetrip.cf/sign-in">http://https://www.severetrip.cf/sign-in</a>	Obtener un porcentaje no mayor a 0, la aplicación de debe responder a todas las peticiones del usuario.	Obtener un porcentaje no mayor a 0, la aplicación de debe responder a todas las peticiones del usuario.	<b>PASA</b>
6. Comentarios: <b>JMETER</b>			
7. Responsable(s) de realizar la prueba: Katherine Sierra			
	Nombre: Katherine Sierra		
	Fecha: 1/10/2019		
	Rol: QA		

**Evidencia:**



DISEÑO DE LOS CASOS DE PRUEBA			
1. Código del caso de prueba: <3 >		2. Nombre del caso de prueba: <b>Post escalabilidad</b>	
3. Identificación del escenario del Caso de Uso: JMETER concurrencia 75 usuarios			
<ul style="list-style-type: none"><li>4. Descripción de la prueba:</li><li>Se realiza una prueba del login de la aplicación con la concurrencia un usuario.</li></ul>			
5. Datos de entrada para ejecutar el caso de prueba:			
Datos	Resultados Esperados	Salida	Pasa / Falla
http:// <a href="https://www.severetrip.cf/sign-in">https://www.severetrip.cf/sign-in</a>	Obtener un porcentaje no mayor a 0, la aplicación de debe responder a todas las peticiones del usuario.	Obtener un porcentaje no mayor a 0, la aplicación de debe responder a todas las peticiones del usuario.	<b>PASA</b>
6. Comentarios:  JMETER			

## 7. Responsable(s) de realizar la prueba:

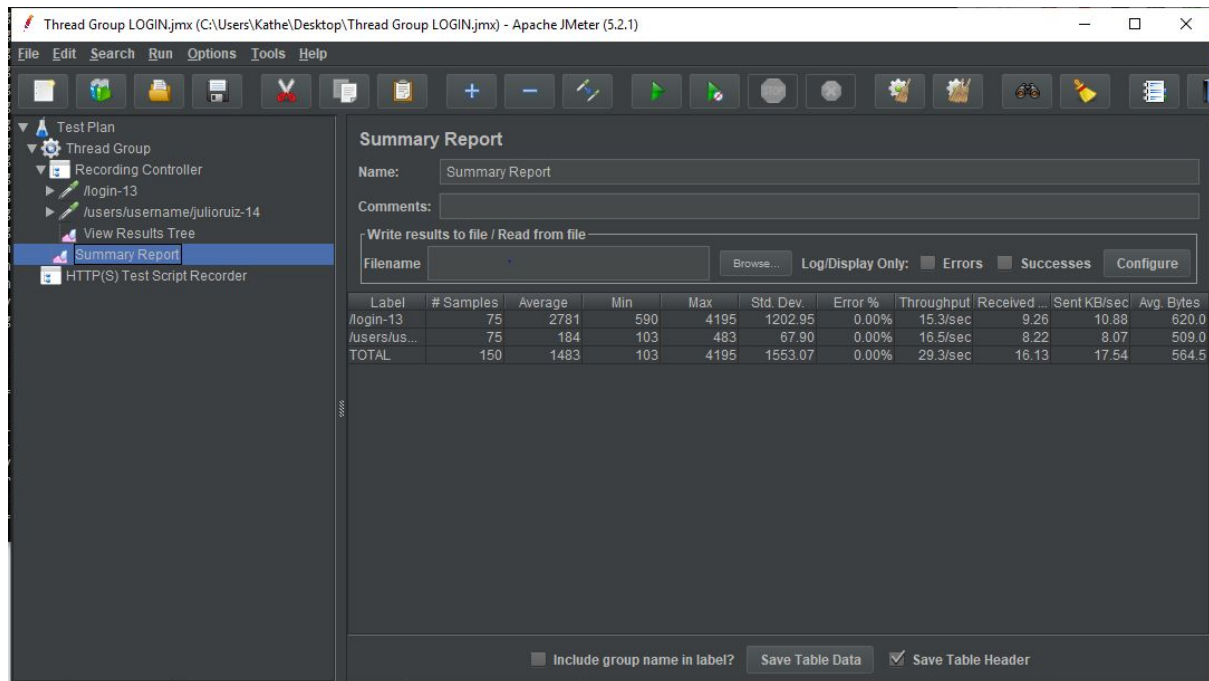
Katherine Sierra

Nombre: Katherine Sierra

Fecha: 1/10/2019

Rol: QA

## Evidencia:



## DISEÑO DE LOS CASOS DE PRUEBA

1. Código del caso de prueba: <3 >

2. Nombre del caso de prueba: **Post escalabilidad**

3. Identificación del escenario del Caso de Uso:

JMETER concurrencia 100 usuarios

• 4. Descripción de la prueba:

- Se realiza una prueba del login de la aplicación con la concurrencia un usuario.

5. Datos de entrada para ejecutar el caso de prueba:

Datos	Resultados Esperados	Salida	Pasa / Falla
<a href="https://www.severetrip.cf/s/ign-in">https://www.severetrip.cf/s/ign-in</a>	Obtener un porcentaje no mayor a 0, la aplicación de debe	Obtener un porcentaje no mayor a 0, la aplicación de debe	<b>PASA</b>

	responder a todas las peticiones del usuario.	responder a todas las peticiones del usuario.	
6. Comentarios: <b>JMETER</b>			
7. Responsable(s) de realizar la prueba: Katherine Sierra			
	Nombre: Katherine Sierra		
	Fecha: 1/10/2019		
	Rol: QA		

### Evidencia:

Thread Group LOGIN.jmx (C:\Users\Kathe\Desktop\Thread Group LOGIN.jmx) - Apache JMeter (5.2.1)

File Edit Search Run Options Tools Help

Test Plan  
 Thread Group  
 Recording Controller  
 /login-13  
 View Results Tree  
**Summary Report**  
 HTTP(S) Test Script Recorder

**Summary Report**

Name: Summary Report

Comments:

Write results to file / Read from file

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

Label	# Samples	Average	Min	Max	Std. Dev.	Error %	Throughput	Received...	Sent KB/sec	Avg. Bytes
/login-13	100	3762	1115	5360	1340.52	0.00%	16.6/sec	10.06	11.82	620.0
TOTAL	100	3762	1115	5360	1340.52	0.00%	16.6/sec	10.06	11.82	620.0

☐ Include group name in label?  ☒ Save Table Header

DISEÑO DE LOS CASOS DE PRUEBA	
1. Código del caso de prueba: <3 >	2. Nombre del caso de prueba: <b>Post escalabilidad</b>
3. Identificación del escenario del Caso de Uso: JMETER concurrencia 500 usuarios	

- **4. Descripción de la prueba:**
- Se realiza una prueba del login de la aplicación con la concurrencia un usuario.

**5. Datos de entrada para ejecutar el caso de prueba:**

Datos	Resultados Esperados	Salida	Pasa / Falla
http:// <a href="https://www.severetrip.cf/sign-in">https://www.severetrip.cf/sign-in</a>	Obtener un porcentaje no mayor a 0, la aplicación de debe responder a todas las peticiones del usuario.	Obtener un porcentaje no mayor a 0, la aplicación de debe responder a todas las peticiones del usuario.	PASA

**6. Comentarios:**

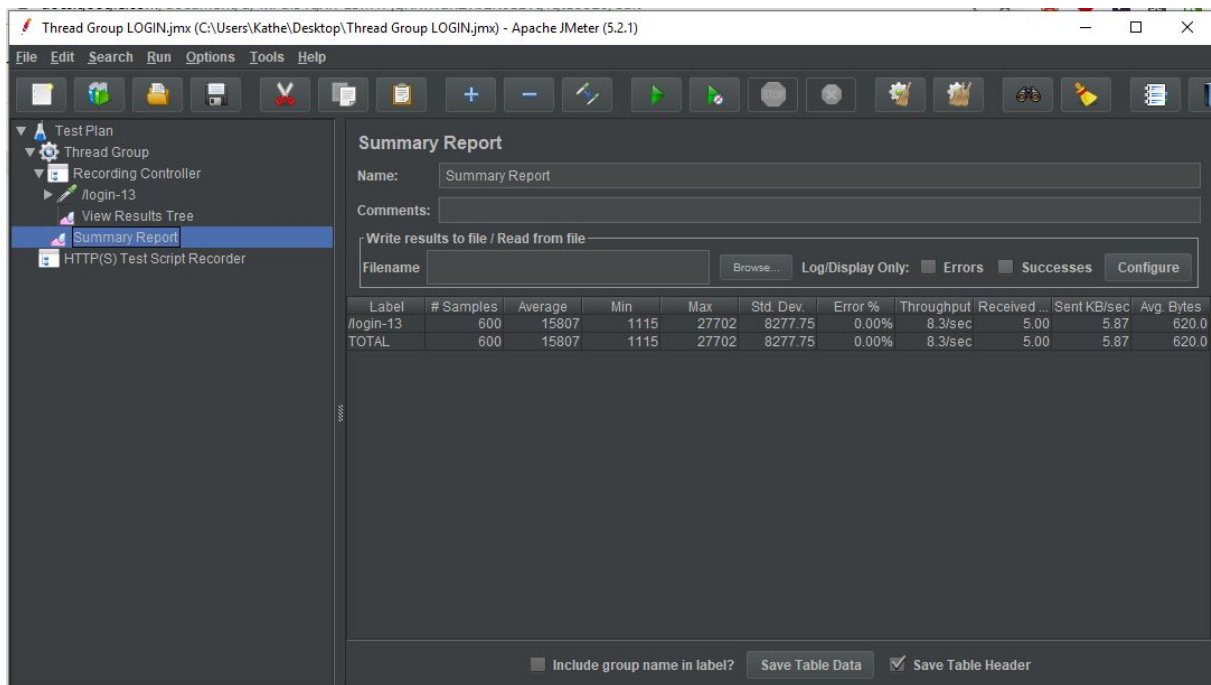
JMETER

**7. Responsable(s) de realizar la prueba:**

Katherine Sierra

	Nombre: Katherine Sierra
	Fecha: 1/10/2019
	Rol: QA

**Evidencia:**



DISEÑO DE LOS CASOS DE PRUEBA			
1. Código del caso de prueba: <3 >		2. Nombre del caso de prueba: <b>Post escalabilidad</b>	
3. Identificación del escenario del Caso de Uso: JMETER concurrencia 1000 usuarios			
<ul style="list-style-type: none"><li>4. Descripción de la prueba:</li><li>Se realiza una prueba del login de la aplicación con la concurrencia un usuario.</li></ul>			
5. Datos de entrada para ejecutar el caso de prueba:			
Datos	Resultados Esperados	Salida	Pasa / Falla
http:// <a href="https://www.severetrip.cf/sign-in">https://www.severetrip.cf/sign-in</a>	Obtener un porcentaje no mayor a 20, la aplicación de debe responder a todas las peticiones del usuario.	Obtener un porcentaje no mayor a 20, la aplicación de debe responder a todas las peticiones del usuario.	<b>PASA</b>
6. Comentarios:  JMETER			

## 7. Responsable(s) de realizar la prueba:

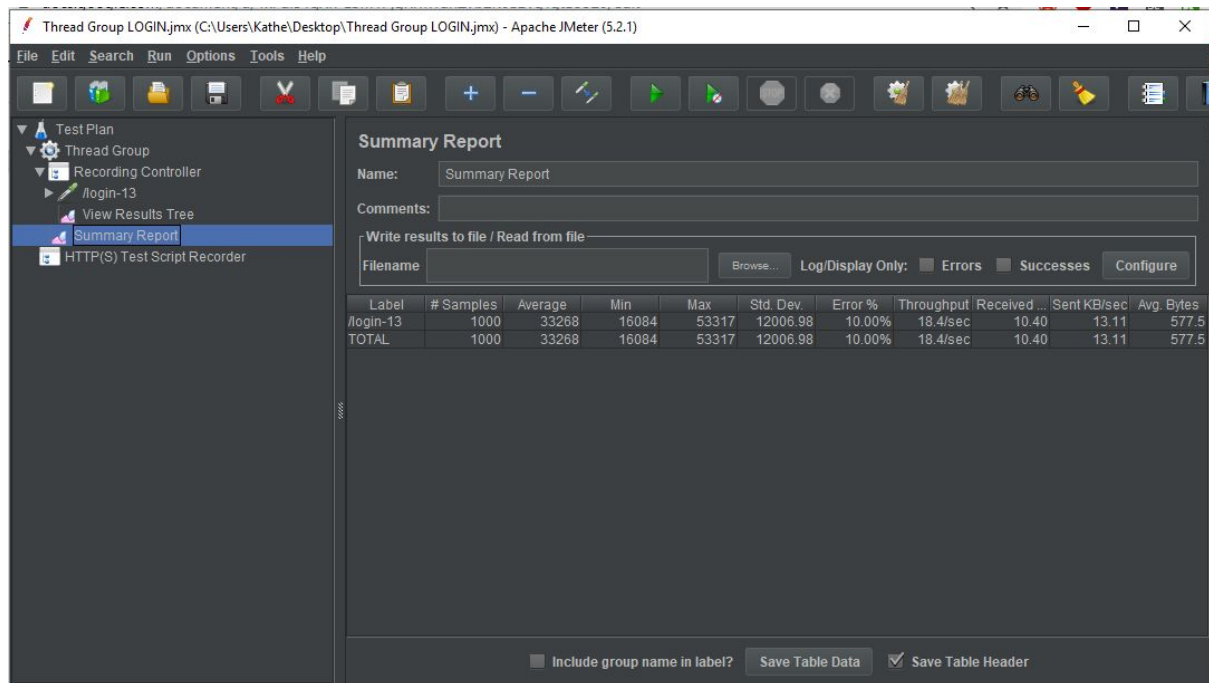
Katherine Sierra

Nombre: Katherine Sierra

Fecha: 1/10/2019

Rol: QA

## Evidencia:



## DISEÑO DE LOS CASOS DE PRUEBA

1. Código del caso de prueba: <3 >

2. Nombre del caso de prueba: **Post escalabilidad**

3. Identificación del escenario del Caso de Uso:

JMETER concurrencia 2500 usuarios

• 4. Descripción de la prueba:

- Se realiza una prueba del login de la aplicación con la concurrencia un usuario.

5. Datos de entrada para ejecutar el caso de prueba:

Datos	Resultados Esperados	Salida	Pasa / Falla
<a href="http://https://www.severetrip.cf/s-ign-in">http://https://www.severetrip.cf/s-ign-in</a>	Obtener un porcentaje no mayor a 50, la	Obtener un porcentaje no mayor a 50, la	<b>PASA</b>

	aplicación de debe responder a todas las peticiones del usuario.	aplicación de debe responder a todas las peticiones del usuario.	
<b>6. Comentarios:</b> <b>JMETER</b>			
<b>7. Responsable(s) de realizar la prueba:</b> Katherine Sierra			
	Nombre: Katherine Sierra		
	Fecha: 1/10/2019		
	Rol: QA		

### Evidencia:

Label	# Samples	Average	Min	Max	Std. Dev.	Error %	Throughput	Received	Sent KB/sec	Avg. Bytes
/login-13	2500	39075	6577	63805	16775.43	49.72%	38.1/sec	15.32	27.11	411.5
<b>TOTAL</b>	<b>2500</b>	<b>39075</b>	<b>6577</b>	<b>63805</b>	<b>16775.43</b>	<b>49.72%</b>	<b>38.1/sec</b>	<b>15.32</b>	<b>27.11</b>	<b>411.5</b>

# Usuarios	Escenario #1		Escenario #2	
	Tiempo de Respuesta (ms)	Throughput (trans/min)	Tiempo de Respuesta (ms)	Throughput (trans/min)
1	178	50,9	293	46,4
5	210	247,9	410	212,8



50	218	2463,1	1285	1312,9
75	432	3142,5	1930	1535,8
100	3310	1392,1	2560	1685,4
500	3850	6185,6	7590	3492,4
750	4678	7925,3	10460	3926,7
1000	6726	7766,0	16084	3512,1
2000		120000,0	60683	1945,4

