**SISTEMA DE TRANSPORTE PÚBLICO - EXPERIMENTO 1**

**Pre-experimentación**

1. **Problemática**

¿Es posible desarrollar un servidor de sistema de transporte que cumpla con unos requerimientos funcionales y que al mismo tiempo pueda satisfacer los atributos de calidad de escalabilidad y desempeño mediante el uso del framework Play!, aprovechando el estilo arquitectural que este ofrece?

1. **Objetivo del experimento**

Medir la escalabilidad y desempeño de una aplicación desarrollada usando un estilo arquitectural asincrónico, asegurando el cumplimiento de unos escenarios de calidad establecidos.

1. **Descripción del experimento**

**Actividades**

* 1. Definir casos de uso para determinar qué requerimientos funcionales debe satisfacer el servidor de sistema de transporte.
  2. Elaborar modelo de clases que permita distinguir las responsabilidades de cada entidad identificada en el problema.
  3. Implementar la capa de lógica de la aplicación haciendo uso del framework Play!
  4. Exponer los servicios REST necesarios para realizar las medidas.
  5. Diseñar las pruebas con respecto a los escenarios de calidad.
  6. Ejecutar las pruebas en un ambiente controlado.
  7. Analizar resultados obtenidos

**Datos a recoger**

En las pruebas se recogen las siguientes medidas:

* Latencia
* Porcentaje de error
* Rendimiento
* Número de solicitudes

1. **Artefactos a construir**

* Aplicación usando el framework the Play! la cual hará de servidor central del sistema.
* Aplicaciones StandAlone, implementados con Java Swing, quienes se encargaran de mandar la solicitudes a la aplicación web central.

1. **Recursos de experimentación**

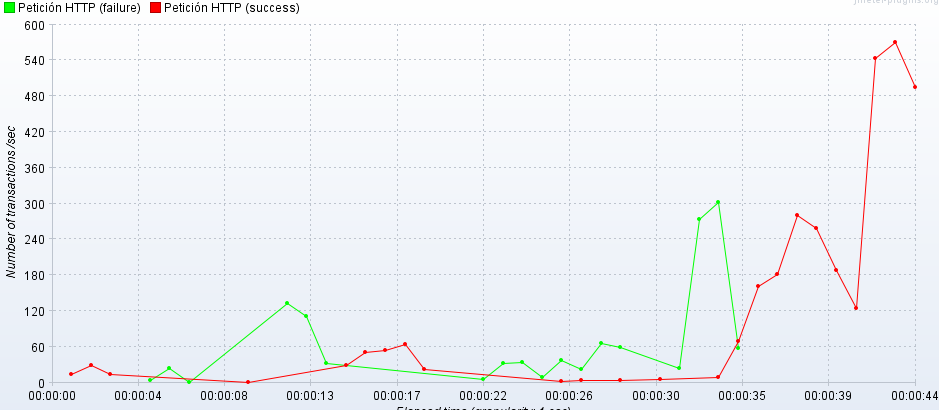
* Máquinas virtuales, en estas se cuenta con la herramienta Apache JMeter, por medio de la cual mediremos latencia y escalabilidad.
* Las máquinas virtuales cuentan con 8gb de memoria principal y dos procesadores de 2.40GHz cada uno.

1. **Resultados esperados**

* Se espera que el sistema permita conocer la posición actual de todos los vehículos y su estado
* Se espera controlar la ruta de los conductores de mobibus
* Se espera manejar de forma eficiente el proceso de manejo de accidentes en los tranvía
* Se espera que el sistema pueda permitir tener control sobre el prestamo y devolucion de los vcub
* Se espera que el sistema permita mantener reporte en tiempo real de los servicios que presta TBC
* Se espera que la información sea transmitida, recibida y procesada de forma inmediata
* Se espera que el sistema sea seguro y no permita el manejo de la información a personas no autorizadas
* Se espera que el sistema sea altamente disponible para los usuarios
* Se espera que el tiempo de respuesta de las funcionalidades de la aplicaciones no exceda 1 s
* Se espera que el sistema soporte hasta 4500 vehículos (4000 VCubs, 250 tranvías, 250 Mobibus) intentando enviar información (posición, alarma, etc.) al servidor principal en 5 s

**Post-experimentación**

1. **Resultados obtenidos:**



1. **Artefactos construidos:**

Solo fue construido el artefacto principal del sistema, la aplicación web, pues es el núcleo de todo, y donde se encuentra la arquitectura. El despliegue en cloud se dejó para más adelante, se consideró que era importante lograr el despliegue de nuestra aplicación en un servidor común y corriente, sin aumentar nuestros recursos, para tratar de cumplir los escenarios de calidad solo con la arquitectura del framework Play!. Las aplicaciones standalone aún no eran necesarias.

1. **Análisis:**

De acuerdo a la gráfica con los resultados obtenidos podemos observar que el sistema soporta hasta 600 transacciones por segundo sin exceder un segundo como tiempo de respuesta. Después de esa cantidad de solicitudes, vemos como el porcentaje de errores aumenta, y al mismo tiempo la latencia. Esta prueba fue realizada con peticiones PUT, pues eran las que más demanda le ejercen a la arquitectura No se cumplen ninguno de los escenarios de calidad, usando recursos base y a pesar de tener una configuración óptima para la arquitectura. .Lo anterior indica el sistema al ser sometido a pruebas de carga con JMeter, bajo escenarios de alta demanda de solicitudes muestra una baja latencia, satisfaciendo el atributo de calidad de desempeño, que se esperaba no superará un segundo de respuesta.

1. **Conclusiones:**

El estilo arquitectural que se use para desarrollar una aplicación tiene un gran impacto en la capacidad de ésta para satisfacer los atributos de calidad, por este motivo es de gran importancia seleccionar una arquitectura que permita asegurar el cumplimiento de unos escenarios de calidad, teniendo en cuenta el trade-off que esta selección de arquitectura pueda significar. Teniendo en cuenta esto, se deben evaluar los recursos actuales para ver si aun con una buena arquitectura, la aplicación se comporta de manera adecuada, y si no es así se debería darle un pequeño empujón agregando más recursos. Aquí vemos la importancia del hardware para un sistema, en términos de disponibilidad y rendimiento.