

**Experimento Número 2 Parcial**

12.03.2017

**─**

Juan Pablo Otalora cod: 201425525

Ana María Espinosa cod: 201425031

Fabio Andrés López cod: 201423782

Grupo numero 2

Universidad de los Andes

Bogotá, Colombia

# **Vision**

Este documento se basa en la idea de comparar y ver los cambios significativos que obtuvimos de las pruebas del experimento 1 con las pruebas de la entrega parcial del segundo experimento. Se tienen en cuenta nuevos elementos agregados a la arquitectura de la aplicación y su impacto en el desempeño, escalabilidad y disponibilidad de esta.

# **Objetivos**

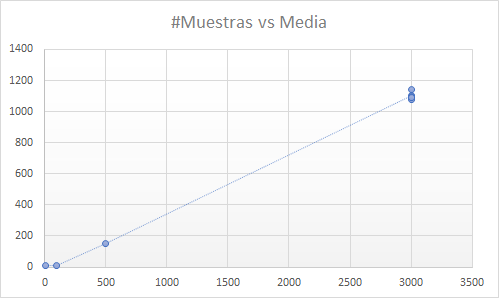
1. Observar el impacto de añadir un balanceador de carga sobre la aplicación y los requerimientos de calidad anteriormente garantizados.
2. Documentar los hechos más importantes de la implementación de un balanceador de carga a partir de políticas de manejo de recursos bien definidas dentro de la aplicación.

# **Cambios realizados**

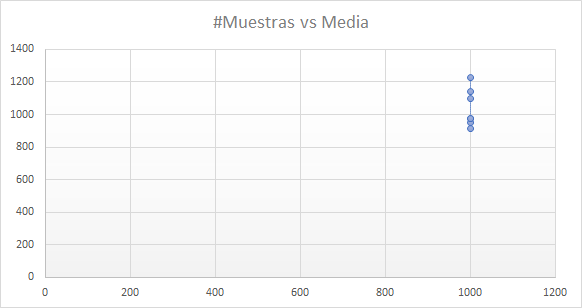
Para esta entrega se implementa un balanceador de carga con la metodología Least-Con para poder satisfacer el requerimiento de calidad de disponibilidad. Para realizar esto se utiliza la herramienta Ngnix que permite implementar un balanceador de carga HTTP. En la configuración de Ngnix se establece un servidor como backup, para que se ejecute en caso de que ninguno de los otros dos servidores se encuentren disponibles. También, se incluyen elementos como la IP de cada máquina, el puerto por el cual se comunican y la siguiente linea de codigo para que las peticiones se redireccionen a alguna de las IPs anteriormente establecida, “proxy\_pass http://myapp1; ”.

# **Resultados obtenidos**

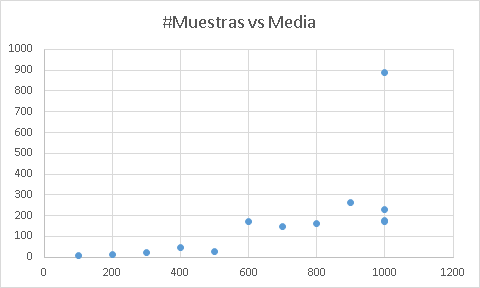
En la entrega parcial del experimento 1 se obtienen los siguientes resultados:



Después, en la entrega final del experimento 1 se obtienen los siguientes resultados:



Finalmente en esta entrega parcial del experimento 2 se obtienen los siguientes resultados:



A partir de los resultados obtenidos se puede decir que, tras la implementación correcta del modelo de actores en Play framework y el uso de un balanceador de carga junto con el algoritmo least\_conn el atributo de calidad desempeño se ve fuertemente mejorado. Los tiempos de latencia ya no superan los mil milisegundos, y las pruebas realizadas no generan error en las transacciones en ninguno de sus intentos.

La implementación del balanceador de carga no solo permite un mejor desempeño al hacer que haya varios servidores trabajando múltiples transacciones al mismo tiempo. Además permite una disponibilidad mayor en la medida que si alguno de estos no responde estarán los otros nodos para soportarlo, y en última instancia se tiene un backup en el nodo central.

Tras la realización de la primera parte del experimento 2 se concluye que la implementación de un balanceador de carga funciona como herramienta para garantizar desempeño y disponibilidad de la aplicación para la Clínica Santafé.