El siguiente documento presenta el planeamiento, ejecución del experimento realizado sobre el sistema SATT construido.

**Justificación de escogencia de la arquitectura**

Se escogió usar la arquitectura implementada con JAX-RS pues, después del laboratorio llevado a cabo la semana pasada, dedujimos que el desempeño y la escalabilidad que permite manejar son adecuados para el número de solicitudes que esperamos manejar, el cual es alrededor de 4000 transacciones. Por otra parte, se ha trabajado anteriormente con la misma, lo cual nos implicaría cierto nivel de facilidad y agilidad a la hora de implementar y modificar el proyecto, a lo largo de las futuras iteraciones con los diferentes atributos de calidad solicitados.

**Pre-experimentación**

1. Problemática: Se formula una hipótesis en busca de resolver o poner a prueba la situación expuesta en el enunciado del experimento.

Se espera probar que el sistema, en su estado actual, puede manejar 2 tipos de transacciones distintas, cumpliendo con ciertos escenarios de calidad asociados al desempeño. En este contexto hablamos de un sistema que recibe notificaciones de estado y de eventos sísmicos, para poder generar boletines de alerta, lo cual implica un alto nivel de urgencia y por eso se plantean las siguientes latencias. En el caso de la recepción de mensajes y notificaciones de eventos sísmicos se espera alcanzar una latencia menor a medio segundo (para 4000 transacciones), al igual que para la generación de boletines de alerta. Nuestra hipótesis es que se pueden alcanzar las latencias anteriormente propuestas para los distintos escenarios de calidad.

1. Objetivo del experimento: Poner a prueba la hipótesis planteada en la arquitectura propuesta como solución al proyecto.

Se pondrá a prueba la hipótesis por medio de simulaciones de carga y estrés, usando la aplicación JMeter. Esta no sólo permite modelar escenarios específicos con tiempos y número de transacciones dados, también permite deducir datos resultantes en el experimento, como por ejemplo el tiempo de latencia máximo de las transacciones.

1. Descripción del experimento: Actividades a realizar, y datos a recoger.

Se utilizará JMeter para modelar los 2 contextos planteados en los 2 escenarios de calidad. Con la misma herramienta se ejecutarán los modelos y se obtendrán los datos resultantes. De estos datos, para los 2 escenarios, tomaremos la latencia por transacción máxima, que nos permitirá deducir el tiempo máximo que tomó llevar a cabo una transacción, y lo compararemos con el tiempo deseado para afirmar si se cumple el escenario de calidad o no.

1. Artefactos a construir: Cualquier elemento de software que se construya y/o se utilice.

A parte del software desarrollado y las pruebas en JMeter no se dispuso de algún elemento adicional.

1. Recursos de la experimentación: Especificaciones, de hardware y software, con las que se va a desarrollar el experimento.

Para el despliegue de la aplicación se usará NetBeans IDE 8.1 y GlassFish Server 4.1.1. Para la ejecución de la prueba se usará Apache JMeter 2.13.

1. Resultados esperados: Describir el comportamiento de implementación parcial de la arquitectura propuesta.

Se espera que las latencias máximas para cada escenario sean menores a las establecidas, para todos los casos la latencia máxima es de medio segundo

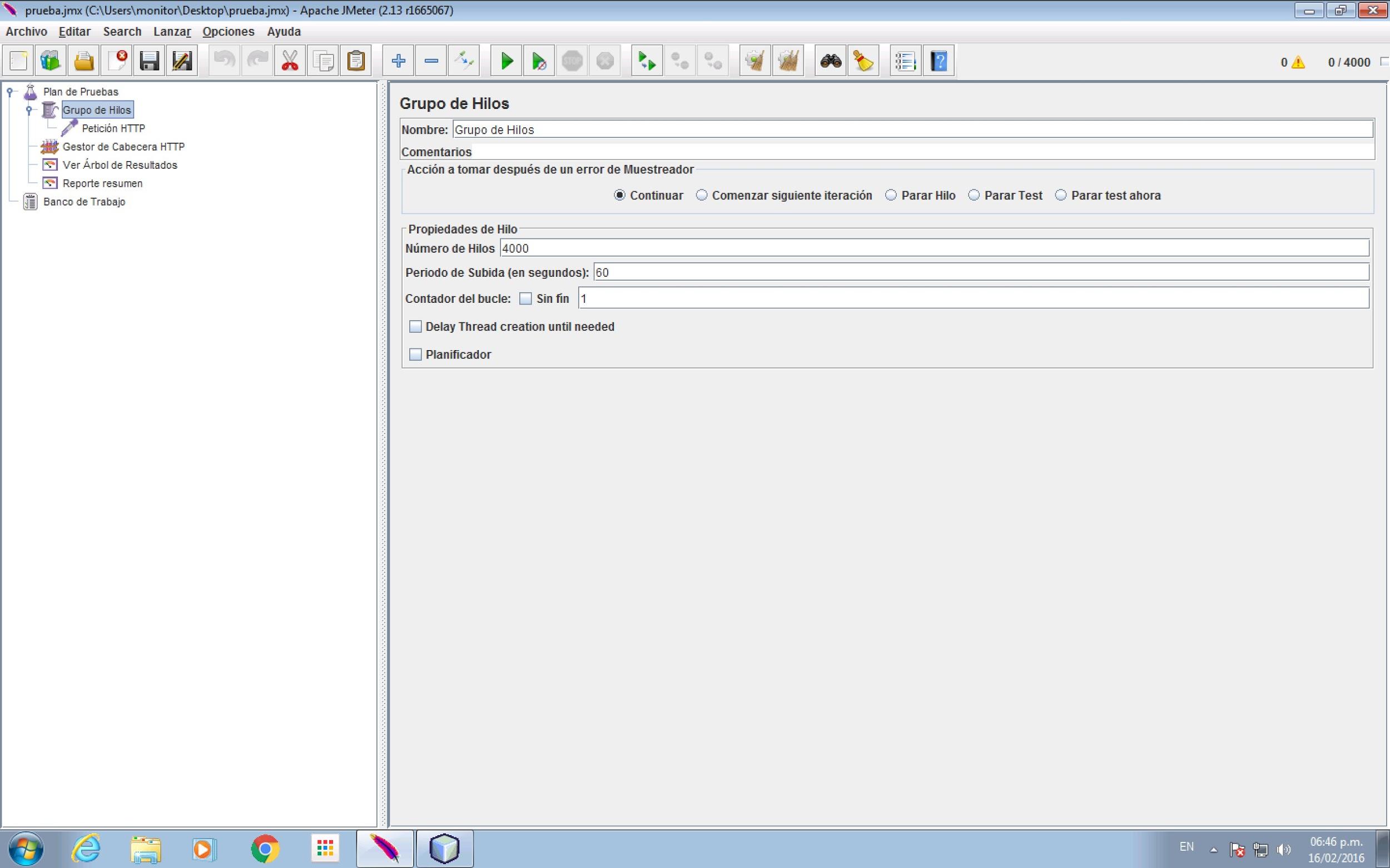
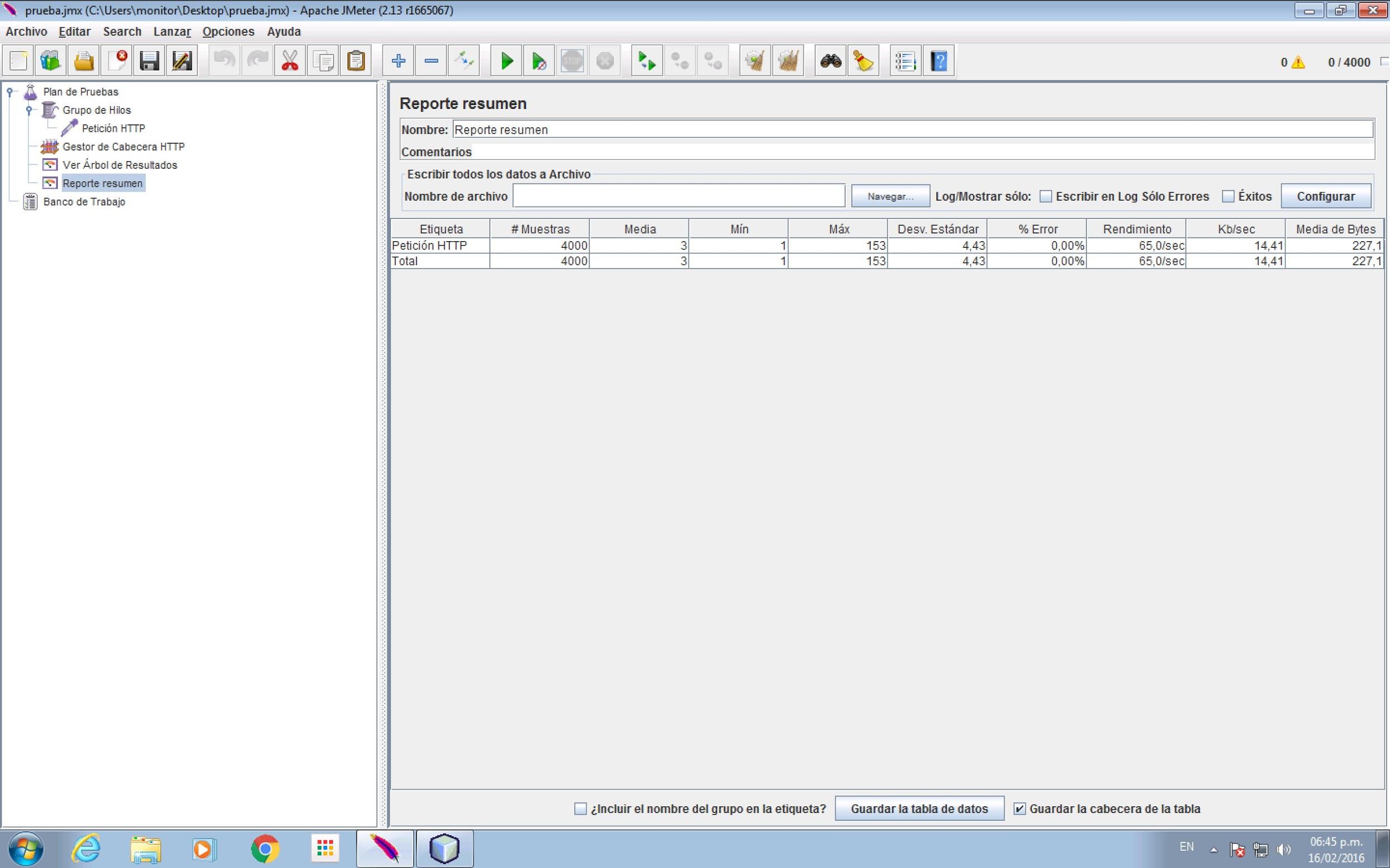
1. Duración y etapas: Planear y definir los entregables de cada etapa, con sus respectivos tiempos de desarrollo teóricos, del proceso de experimentación.

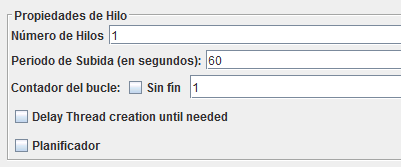
Se entregarán una serie de pantallazos mostrando los resultados de cada una de las pruebas con sus escenarios planteados. La ejecución de cada una de las pruebas no ha de tomar más de 5 minutos. Se llevará a cabo cada una de las pruebas por separado.

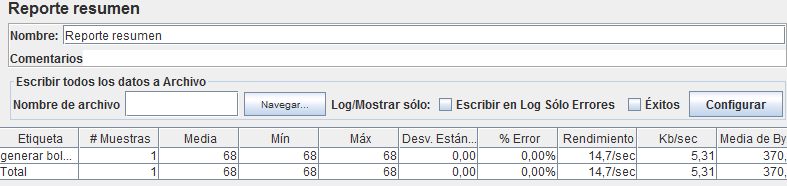
**Post-experimentación**

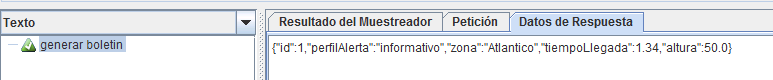
En esta fase se realiza el análisis del experimento. Es imperativo tener claro lo siguiente:

1. Resultados obtenidos

Prueba 1: Latencia máxima de 500 milisegundos, 4000 solicitudes de enviar mensaje con latitud, longitud, velocidad y altura de los sensores. El resultado fue 153 milisegundos (0,153 segundos), lo que quiere decir que el escenario de calidad se cumple, pues la latencia máxima fue menor a 500 milisegundos.







Prueba 2: Latencia en la generación de boletines menor a 500 milisegundos. El resultado que obtuvo para la generación de 1 boletín fue de un máximo de 68 milisegundos y el boletín se retorna como un objeto JSON, como se ve anteriormente, lo cual permite afirmar que el escenario de calidad se cumple.

Vale la pena aclarar que el estímulo de la prueba corresponde a 1 thread, ya que el boletín de alerta se genera una vez se registra un evento sísmico, por lo cual no es acorde con la realidad realizar una prueba como la anterior, donde se generen 4000 eventos sísmicos para producir 4000 boletines.

1. Duración real: Realizar una comparación con los tiempos teóricos y los reales de desarrollo para cada etapa del proceso de experimentación.

Para la prueba 1 el tiempo de ejecución total fue de 3 minutos 14 segundos. Esto fue inferior al tiempo planteado de 5 minutos, el cual se estimó aleatoriamente con base en ejecuciones previas. De la misma manera, la prueba del segundo escenario no tomó más de 1 minuto, por lo cual se puede afirmar que se tuvo un tiempo óptimo de realización de pruebas.

1. Artefactos construidos: Cuáles artefactos fueron construidos y cuáles no, explicando las causas por las que no fueron desarrollados.

Los únicos artefactos construidos fueron los escenarios en JMeter. No se llevo a cabo ninguna otra construcción pues las herramientas disponibles fueron suficientes para modelar los escenarios con todas sus cualidades establecidas y poder llevar a cabo las pruebas necesarias.

1. Análisis: Interpreta, explica y justifica los resultados obtenidos basado en patrones y tácticas arquitecturales implementadas en el experimento.

Podemos argumentar que la arquitectura escogida fue favorecedora para los escenarios de calidad que se deseaba alcanzar. Podemos afirmar esto pues estos escenarios requerían valores menores a ciertos máximos de latencia, para contextos específicos, y se pudo mantener los tiempos de respuesta por debajo de los máximos establecidos. Esto se debe a que la arquitectura JAX-RS, por su estructura, permite el acceso directo a los recursos necesarios para llevar a cabo la transacción, que aunque puede ser costoso a nivel de consumo de recursos favoreció el desempeño de la aplicación. En caso de manipularse información o modelos más pesados sería necesario considerar una arquitectura diferente, pero en este caso se manipulan objetos simples tales como los mensajes y boletines, que con pocos datos permiten el manejo de la lógica de todo el sistema.

1. Conclusiones: Teniendo en cuenta la hipótesis, plantea recomendaciones basadas en los resultados obtenidos en la etapa de experimentación.

En conclusión el experimento fue exitoso y se puede afirmar que, para los escenarios de calidad establecidos, la aplicación tiene un buen desempeño. La arquitectura escogida y el diseño implementado permiten obtener una aplicación capaz de cumplir con los requerimientos de calidad estipulados en este experimento y se espera que tenga un buen desempeño en futuros eventos. Para esto se pensó en la implementación de EJB’s sencillos, modelos desacoplados y servicios atómicos. Todo esto es una estrategia que nos permitirá modificar el proyecto acorde a nuevos requerimientos con un mínimo impacto en los ya cumplidos.