**Pre-experimentación:**

* **Problemática:** Se plantea desarrollar una arquitectura por capas que use servicios REST para la comunicación con los sensores por medio de peticiones HTTP y que sea capaz de cumplir con requerimientos funcionales y no funcionales de la aplicación. Para la parte 2 del proyecto se le agrega unos requerimientos no funcionales los cuales son disponibilidad (desarrollada con múltiples copias de datos en servidores diferentes), confidencialidad (desarrollada con tecnología Shiro la cual permite manejar una serie de usuarios y asignarles ciertos permisos dentro de la información de la aplicación) e integridad de datos (desarrollada a base de un checksum el cual verifica este requerimiento no funcional).
* **Objetivo del experimento:** Probar la arquitectura planteada para verificar el cumplimiento de los nuevos requerimientos no funcionales.
* **Descripción del experimento:** Se desarrollará la capa de persistencia, la capa de presentación y se expondrán diferentes servicios con el fin de suplir otros requerimientos funcionales. Se usarán pruebas de carga con el fin de probar escalabilidad y latencia del software. De igual manera, los nuevos requerimientos se harán con pruebas de carga.
* **Artefactos a construir:** Se construirá la capa de persistencia dentro del backend del proyecto y en otra capa se realizará la interfaz gráfica para que desde allí se consuman los recursos de la aplicación. También se agregarán nuevos servicios de autenticación y validación por medio de Shiro y Stormpath y se generara el checksum de verificación de datos.
* **Recursos de la experimentación:** El experimento se desarrollará en el ambiente de ejecución NetBeans 8.1, como servidor para desplegar la aplicación se utilizará GlassFish 4.1.1 y para las pruebas de carga se utilizará JMeter.
* **Resultados esperados:** Se espera que el sistema pueda atender cerca de 4000 solicitudes en menos de un minuto, además que en caso de un evento sísmico sea capaz de generar un boletín de emergencia en menos de 500 ms, todo eso protegiendo en todo momento la integridad de los datos y la confidencialidad de los mismos.
* **Duración y etapas:** Esta etapa finalizará para el 29 de abril de 2016. Deberá incluir pruebas de carga de la arquitectura planteada y la incorporación de la confidencialidad e integridad de datos. Se realizará entregas semanales de avances progresivos. Inicialmente se llevará a cabo la incorporación de la persistencia.

**Experimentación:**

Para este experimento se desarrolló y ejecutó las pruebas de carga de los siguientes requerimientos con el fin de medir la disponibilidad del sistema:

* Generación de boletín (Latencia menor a 500 milisegundos, probado con 100 Threads)
* La actualización de la información de un sensor en cambios de altura de una zona específica (Latencia de 1 segundo, probado con 100 Threads)
* La actualización de la información de 4000 sensores (Escalabilidad)

A continuación, se muestran algunos resultados obtenidos para estas pruebas y sus respectivos análisis:

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Estado Experimento2,2** | |  |  | **Estado Experimento 2,1** | |  |  |
|  |  |  |  |  |  |  |  |
| # Muestras | Error | Rendimiento | KB/s | # Muestras | Error | Rendimiento | KB/s |
| 100 | 0,00% | 1,7 | 884,07 | 100 | 0,00% | 0,835 | 882,31 |
| 300 | 0,00% | 5 | 2629,57 | 300 | 0,00% | 4,9 | 521,67 |
| 500 | 0,00% | 8,3 | 4357,26 | 500 | 0,00% | 5,6 | 5945,25 |
| 800 | 0,00% | 13,2 | 6966,79 | 800 | 0,00% | 13,1 | 13862,56 |
| 1000 | 0,00% | 16,5 | 8697,57 | 1000 | 0,00% | 12,1 | 12831,81 |
| 3000 | 0,00% | 8,5 | 4468,81 | 3000 | 55,64% | 29,3 | 13774,34 |
| 4000 | 0,00% | 2,9 | 1538 | 4000 | 67,42% | 14,1 | 4863,58 |

**Post-experimentación:**

* **Resultados obtenidos:** Se obtuvieron resultados positivos para las dos pruebas de latencia. Sin embargo, para la prueba de escalabilidad no se cumplió con el requerimiento no funcional establecido.
* **Duración real:** A pesar que se cumplió con los tiempos establecidos, los resultados obtenidos no son del todo satisfactorios y la incorporación de las demás capas de la aplicación no se encuentra aún al 100%.
* **Artefactos construidos:** Se desarrollaron todos los artefactos mencionados en la pre-experimentación. Sin embargo, la incorporación de la persistencia no se ejecutó del todo, por un error en la actualización de tablas.
* **Análisis:** Se le atribuye los resultados negativos obtenidos en la prueba de escalabilidad al diseño de la arquitectura. En primer lugar, porque una petición tiene que viajar por las diferentes capas de la aplicación y además de esto una petición de inserción es más costosa que una de consulta. En segundo lugar, se les atribuye a los recursos computacionales con los que se cuenta.
* **Conclusiones:** Para que el tiempo de inserciones concurrentes en la aplicación disminuya es necesario acortar el camino por el que viaja el flujo de información de una petición.