

# Documento de experimentación

Grupo Arquisuave

Camila García

Edgar Margffoy

Jose Gabriel Tamura

Margarita Gómez

## Pre-experimentación

### Problemática

Se tienen 1200 pozos, cada uno con 4 sensores que envían información a intervalos diferentes de tiempo al sistema.

### Objetivo del experimento

Se pretende implementar un sistema de recepción, registro y almacenamiento de registros provenientes de un conjunto de sensores dispersos geográficamente, con una latencia mínima. El sistema debe generar reportes.

### Descripción del experimento

Se debe implementar el sistema que sea capaz de administrar usuarios, campos y pozos con sus respectivas relaciones, y que sea capaz de recibir la información de todos los sensores de la topología de los pozos. Se debe encontrar la manera adecuada de guardar los registros de estos sensores de manera que más adelante sea posible recuperar la información en reportes. Por otro lado, estos reportes deben poder ser producidos con un intervalo de fechas dado, una región dada y/o un jefe de campo dado.

### Artefactos a construir

Se construye la solución a partir de un servidor basado en una arquitectura asíncrona que responde a peticiones soportadas por REST y desplegada por Scala. Se deben crear controladores y modelos para representar las entidades de usuario, modelo, campo y registro, de manera que puedan ser utilizadas correctamente por el CRUD necesario. Se debe además representar la entidad de reporte, que pueda ser restringida por filtros como fecha, jefe, campo y región.

## Recursos de la implementación

Se recurrió al uso de tres máquinas virtuales basadas en Windows 7 (x86\_64), cuya capacidad corresponde a 8Gb de Memoria RAM, un procesador de 1.2Ghz, así mismo, se emplearon dos máquinas Unix, la primera basada en Linux y la segunda en Mac OS X. La base de datos empleada corresponde a PostgreSQL 9.5. El sistema fue desarrollado y desplegado sobre Play Framework, empleando actores y colas de Akka.

## Resultados esperados

Se espera que el sistema sea capaz de responder a las peticiones de todos los sensores incluso en el caso en que todas estas coincidan al mismo tiempo. Sin embargo, algunos errores son esperados en el caso en que todos los sistemas se encuentren. Si este error es pequeño, puede considerarse despreciable debido al hecho de que los reportes manejan promedios y un registro puede no estar presente sin mayor problema. Se espera que las peticiones de emergencia sean respondidas inmediatamente y que su error sea siempre máximo del 0%, debido a su carácter urgente.

## Duración y etapas estimadas

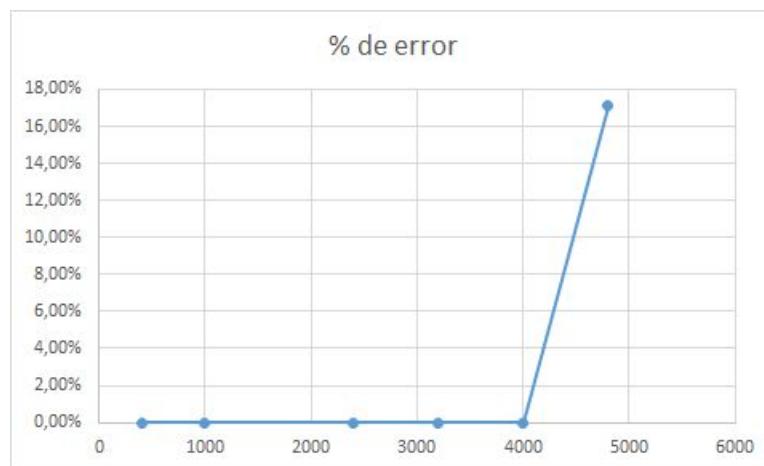
Etapas	Duración estimada
Planeación del modelo y la arquitectura	1/2 día
Desarrollo de la lógica de la aplicación	2 días
Pruebas de carga	1/2 día

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

## Post-experimentación

### Resultados obtenidos

No. De muestras	Tiempo promedio	Mínimo	Máximo	Desv. Estándar	% de error	Rendimiento	Kb/sec	Media de bytes
4800	3142	11	14786	2637,25	17,12%	269,0/sec	139,96	532,7
4000	1849	8	6126	1095,14	0,00%	394,5/sec	83,68	217,2
3200	1317	11	5209	875,8	0,00%	409,3/sec	86,8	217,2
2400	866	7	3489	597,32	0,00%	448,5/sec	95,12	217,2
1000	337	6	1186	223,11	0,00%	409,8/sec	86,86	217
400	70	3	428	80,08	0,00%	260,8/sec	55,18	216,7



## Duración real

Inicialmente, se esperaba un tiempo de desarrollo de alrededor 3 días, sin embargo, el proceso general tardó casi 4 días. Con respecto al proceso de diseño de la aplicación, fue necesario emplear alrededor de 6 horas. A continuación, se procedió a realizar la fase de configuración de entorno y definición de parámetros asociados a cada tecnología, ésta fase fue culminada tras 5 horas. Posteriormente, se dio inicio al proceso de implementación de la aplicación, conforme a un patrón de diseño MVC, el cual, fue finalizado tras 36 horas de esfuerzo grupal. Finalmente, se procedió al diseño y ejecución de pruebas de carga sobre la implementación realizada, con respecto a esta fase, es posible apreciar que el tiempo total de ejecución ronda alrededor de 9 horas.

## Artefactos construidos

Debido a que la aplicación se basa en un patrón de diseño MVC, fue necesario definir los modelos y controladores asociados a las entidades involucradas en el funcionamiento de la aplicación, en este caso, fue posible definir dos artefactos generados (Modelos y controladores, respectivamente), estos dos artefactos fueron definidos en torno al

funcionamiento de la aplicación, es decir, de acuerdo a los diferentes tipos de sensores que existen en cada uno de los pozos, así como los reportes y emergencias asociados a los mismos. Entre los artefactos no construidos se encuentra un generador de reportes, debido a que éste no contaba con una prioridad alta durante esta fase de desarrollo e implementación de la aplicación.

## Análisis

Los resultados indican que el servidor fue capaz de soportar la demanda de peticiones máxima (4.800 solicitudes por minuto), con un pequeño porcentaje de error. Esto significa que los patrones e implementación del diseño fueron suficientes para cumplir totalmente con los requisitos del sistema bajo condiciones normales y tener un excelente desempeño bajo condiciones de stress. Puede decirse que los resultados obtenidos se dieron debido a la implementación de una arquitectura reactiva, que mediante el uso de un thread pool, un dispatcher y una cola de mensajes, logran optimizar los tiempos de atender una sola solicitud delegando a otros hilos tareas más granulares. Esto se ve reflejado en los tiempos de respuesta del servidor, siendo menor a 4 segundos en el peor de los escenarios.

## Conclusiones

Fue claro al momento de hacer pruebas con un gran número de registros simultáneos que la idea original de hacer promedios en el momento de llegada del registro no es posible debido a la falta de atomicidad de la consulta a la base de datos. Es claro que esto necesita un cambio. Por otro lado, se descubrió que el sistema de emergencias, a pesar de tener una cola separada, tuvo algunos errores, los cuales no son despreciables debido a su urgencia. Por eso, debe encontrarse la manera de evitar que esto suceda en el futuro.

Finalmente, la configuración tanto del ambiente de trabajo como de pruebas tardó más de lo esperado y retrasó el flujo de trabajo. Esto afectó también la estimación de tiempo que se tenía planeada, por lo que nos obligó a trabajar más rápido y bajo más presión. Se recomienda hacer la configuración necesaria con anterioridad para los próximos experimentos, y tenerla en cuenta en el momento de estimar los tiempos de trabajo.