

**DOCUMENTO DE EXPERIMENTACIÓN
ENTREGA PARCIAL 1**

Nombre del grupo:

DevAndes

Integrantes:

Daniel Althviz

José Daniel Fandiño

Camilo Mendoza

Maria Camila Remolina

Profesora:

Kelly Garces Pernet

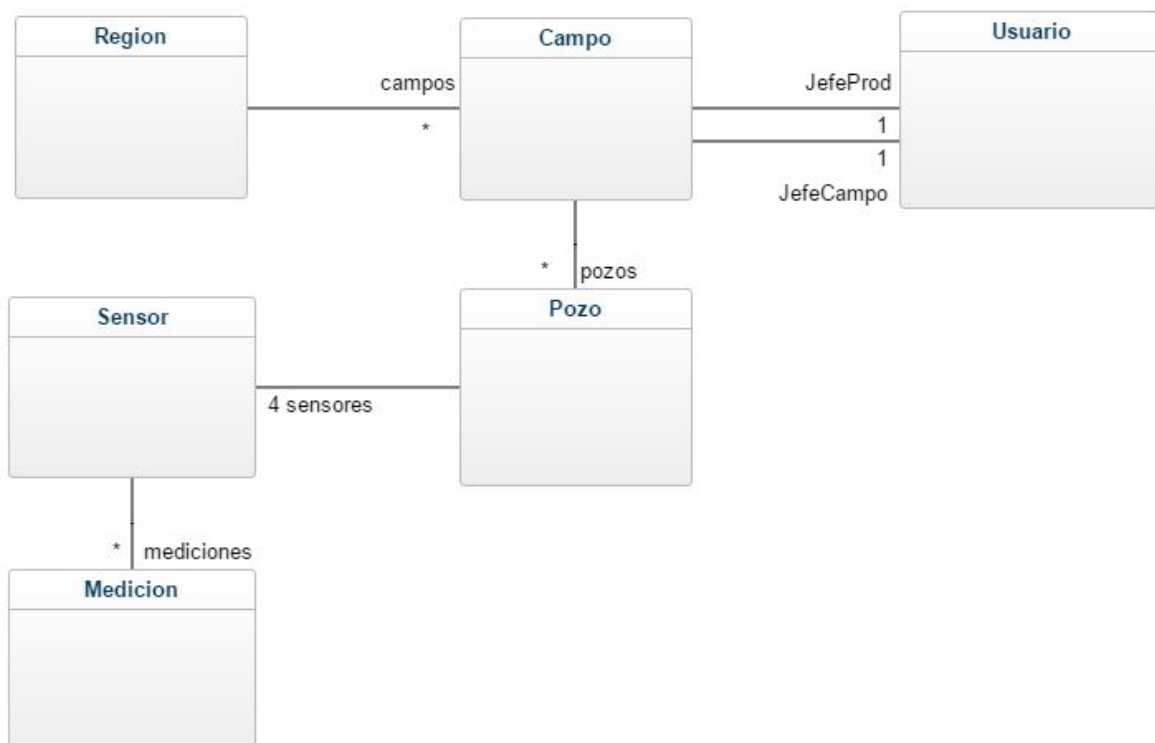
**Universidad de los Andes
Agosto 31, 2016**

PRE-EXPERIMENTACIÓN

1. Problemática:

El contexto del proyecto es, en resumen, que una compañía que extrae petróleo (OilCol) quiere implementar un sistema de mediciones automáticas para monitorear el estado de cada uno de sus pozos. Lo hará por medio de sensores que envían sus datos a la aplicación a desarrollar. Nuestro trabajo es implementar esta aplicación y ponerla a prueba para que satisfaga los atributos de calidad que demanda el proyecto.

El UML del proyecto es el siguiente:



2. Objetivo del experimento:

El objetivo de este experimento es validar las decisiones de diseño que nuestro equipo de trabajo DevAndes ha tomado con respecto a los requerimientos funcionales y no funcionales. Los requerimientos funcionales están claros en el enunciado ([enlace](#)). Los no funcionales corresponden a cumplir los atributos de calidad de *desempeño* y *escalabilidad*.

3. Descripción del experimento:

Se van a generar mediciones de 4800 sensores y se mandarán por medio de peticiones REST al servidor donde se encuentra la aplicación desplegada. Se recogerán datos de medición de tiempos de respuesta del sistema y el éxito de cada petición, ambas por medio de pruebas.

4. Artefactos a construir:

Se va a construir la capa de lógica de una aplicación que corre en Play. También se genera una conexión a una base de datos para persistir la información generada. Se usa también Derby para base de datos y JMeter para las pruebas.

5. Recursos de la experimentación:

Hardware:

Se usará como HW los sensores de las mediciones para cada pozo. Estos se comunicarán con la aplicación mandando peticiones REST al servidor local (del campo). Él se encargará de manejar toda la lógica y persistencia. El sensor solo manda la información, la aplicación la maneja.

Software:

La aplicación pedida va a correr en una aplicación java de escritorio. Para la entrega parcial solo se pide la capa de lógica, por lo cual se usará postman para las peticiones, pero en la entrega del experimento 1 ya se tendrá una UI. Se usarán las máquinas virtuales de admonsis. En particular, una máquina será el servidor (aplicación que recibe las peticiones de los sensores); otra máquina se encarga de representar la base de datos a la cuál se persisten los datos del servidor; y las otras dos máquinas se encargan de generar las peticiones.

6. Resultados esperados:

El software parcial atiende por medio de peticiones REST por lo menos 4800 mediciones en menos de 1 segundo con respuesta exitosa. Al igual debe ser capaz de manejar peticiones de emergencia con el protocolo pedido por el enunciado.

7. Duración y etapas:

Esta entrega corresponde a la parte inicial del experimento 1. La siguiente entrega corresponderá a la octava semana (de acuerdo al calendario del curso) y tendrá ya finalizada la capa de persistencia de la aplicación con las consultas optimizadas para mejorar sustancialmente los tiempos de acceso a la base de datos. Se finalizará el proceso de experimentación, comparando tiempos y resultados de ambas implementaciones. Se esperan mejores resultados en desempeño.

PRUEBAS DE CARGA CON JMETER

1. Configuración:

Se configuró JMeter para que recibiera un número incremental de peticiones, para poder medir el rendimiento de la aplicación y los tiempos de respuesta.

2. Resultados:

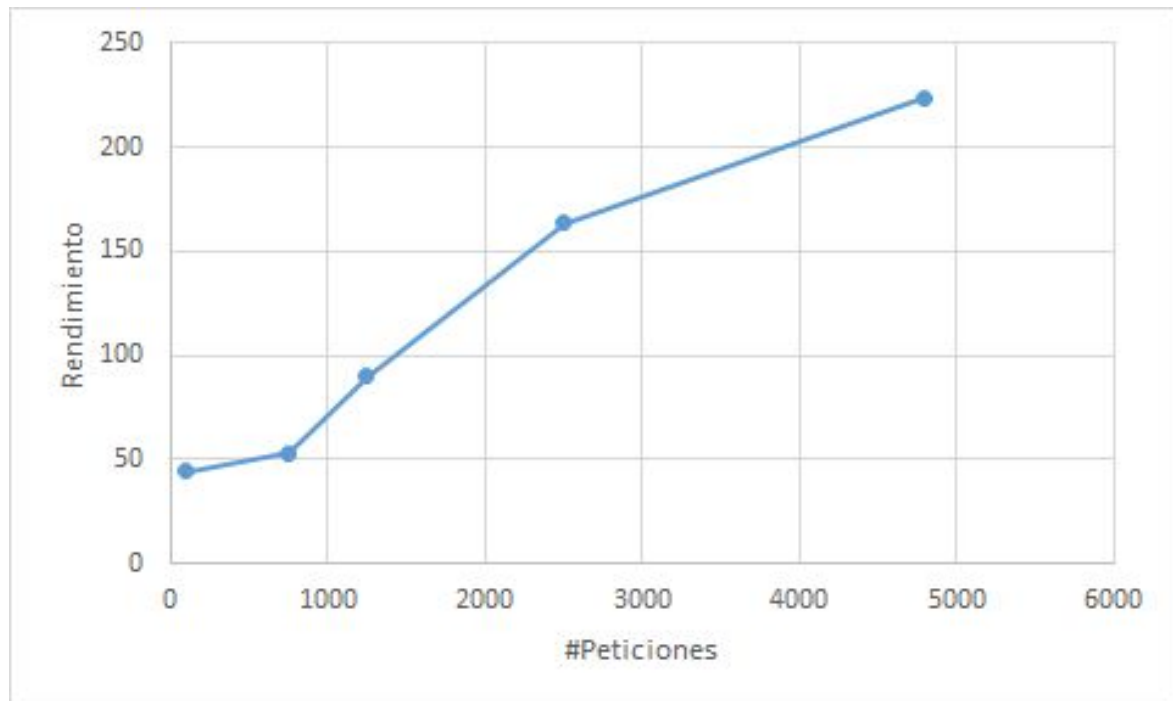
En la tabla a continuación se muestran los resultados obtenidos, donde cada fila corresponde a una carga de datos cada vez mayor.

# Muestras	Media	Mín	Máx	Desv. Estándar	% Error	Rendimiento	Kb/s	Media de Bytes
100	1191	327	1676	280,90	0,00%	44.6	8,27	189,8
750	4855	630	13917	3137,87	37,47%	53.4	43,05	826,0
1250	5172	463	13126	4329,73	47,52%	90.2	87,80	996,4
2500	4545	541	13491	4541,37	69,36%	163.5	218,24	1366,8
4800	5109	270	18523	4974,33	76,96%	223.7	326,66	1495,6

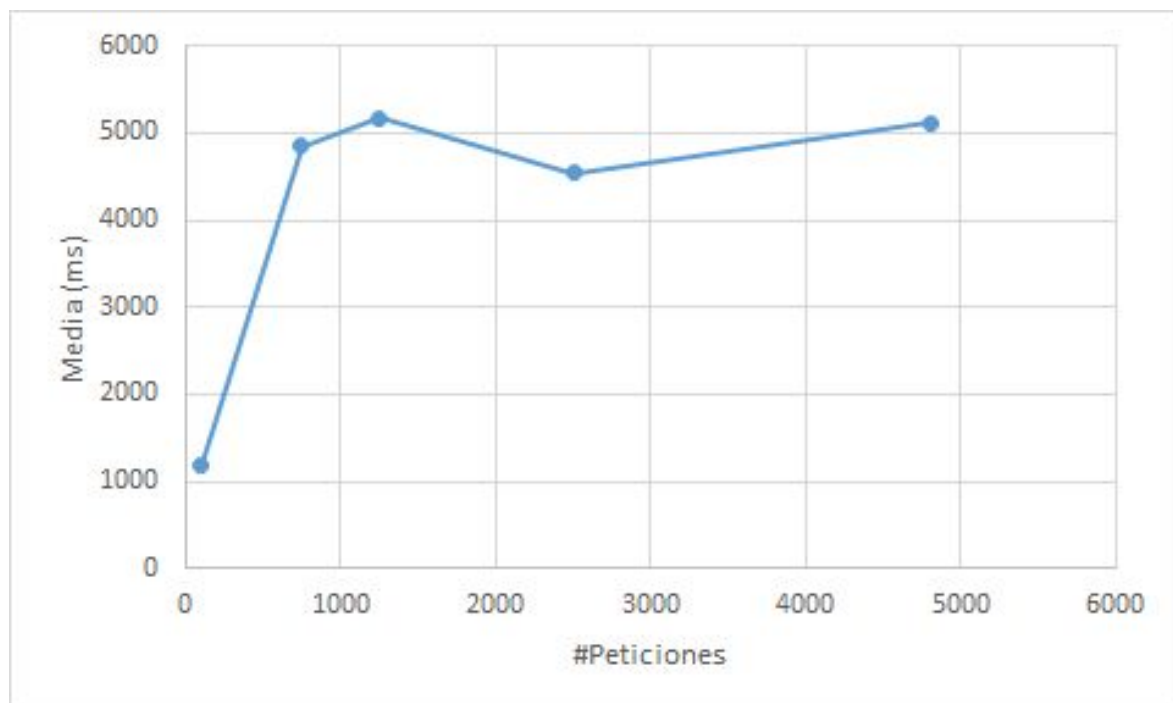
3. Gráficas:

Para visualizar mejor esta información realizaremos a continuación varias gráficas con su explicación.

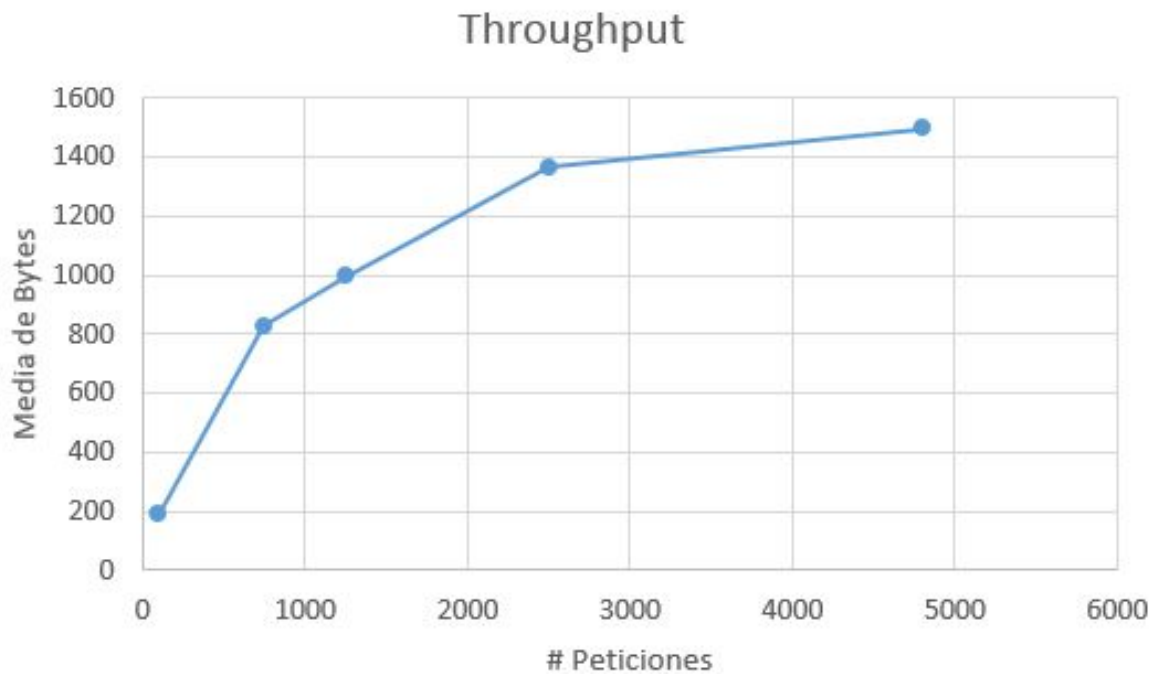
a. Rendimiento



b. Tiempo de respuesta medio



c. Throughput



4. Análisis:

Para un tiempo promedio de 5109 ms por 4800 peticiones, se tiene que la respuesta del sistema aunque no es ideal, es del mismo orden de magnitud de la esperada. Este tiempo puede ser mejorado al optimizar la consultas o incluso las relaciones entre las entidades y componentes del problema.

Sin embargo, JMeter ya está configurado para que acepte todas nuestras peticiones al tiempo. Al igual las peticiones son aceptadas correctamente y la lógica del sistema está funcionando detrás. Nuestro enfoque ahora será la capa de persistencia y lograr que las pruebas de carga mejoren significativamente.