**LABORATORIO 1**

**ZAMIR ANDREI GARCÍA ROMERO**

**ÁLVARO DAVID LÓPEZ PINILLA**

**DIEGO ANDRÉS LOZANO ROLDÁN**

**WILMAN RINCON BAUTISTA**

**DEIVIS ENRIQUE VERGEL ARENAS**

**GRUPO STATUS QUO**

**ARQUITECTURA DE SOFTWARE**

**ESPECIALIZACIÓN EN CONSTRUCCIÓN DE SOFTWARE**

**UNIVERSIDAD DE LOS ANDES**

**2015**

# **Objetivo:**

El propósito de estos experimentos es establecer la solución más adecuada que nos permita cumplir con los requerimientos no funcionales de Latencia y Escalabilidad, bajo las métricas mencionadas en el enunciado.

# **Infraestructura usada:**

La infraestructura a continuación, describe el ambiente que será usado para la elaboración de las pruebas:

* Hardware:
  + Procesador: QuadCore AMD 2199 MHz Cache 512 KB
  + Memoria RAM: 1,5 GB
* Software:
* Contenedor: Glassfish 4
* D.B.: PostgreSQL
* O.S.: Centos 5

Las condiciones de conexión, estarán sujetas al servidor que se usó. Las pruebas se realizarán desde redes externas de donde se encuentre instalado el servidor.

# **Escenarios planteados:**

Los siguientes son los escenarios arquitecturales que se comprobaran, teniendo en cuenta que no se realizaran ajustes a la infraestructura, sino que se usaran diferentes patrones de diseño.

## Patrón 1:

La arquitectura de la primera aplicación a probar consiste en una Aplicación Web JEE en la que un servicio REST de Jersey utiliza un DAO implementado por un EJB instanciándolo por Lookup local. A su vez el DAO utiliza un EntityManager de JPA por Inyección de dependencia para administrar 4 entidades que mapean 4 tablas de una base de datos Postgres.

## Patrón 2:

La arquitectura de la segunda aplicación consiste en una Aplicación Empresarial JEE que contiene un artefacto WAR y un JAR-EJB empaquetados en un EAR. La aplicación WEB contiene únicamente un Servicio REST de Jersey que implementa un EJB Stateless que expone el método GET en que lista todos los pacientes registrados y el método POST que permite insertar un paciente. La aplicación de Datos y Negocio JAR-EJB se organizó en dos capas, una capa EJB de Negocio que encapsula lógica de negocio y una capa de persistencia que implementa unas entidades y unos DAO implementados en EJB, todas las relaciones de las capas se comunican por inyección de dependencia.

## Patrón 3:

La arquitectura de la tercera aplicación es una variante de la primer aplicación, que corresponde a solo una Aplicación Web JEE que contiene las 3 capas, la capa de presentación corresponde a un Servicio REST Jersey que a diferencia del patrón 1 implementa un EJB Stateless que permite realizar Inyección de dependencia a los EJB que implementan la capa de lógica de negocio que a su vez inyectan la capa de acceso a datos con sus entidades con JPA.

# **Ejecución del experimento:**

Durante este experimento se plantearon 3 hipótesis que creemos que pueden cumplir con los requisitos planteados:

## Patrón 1:

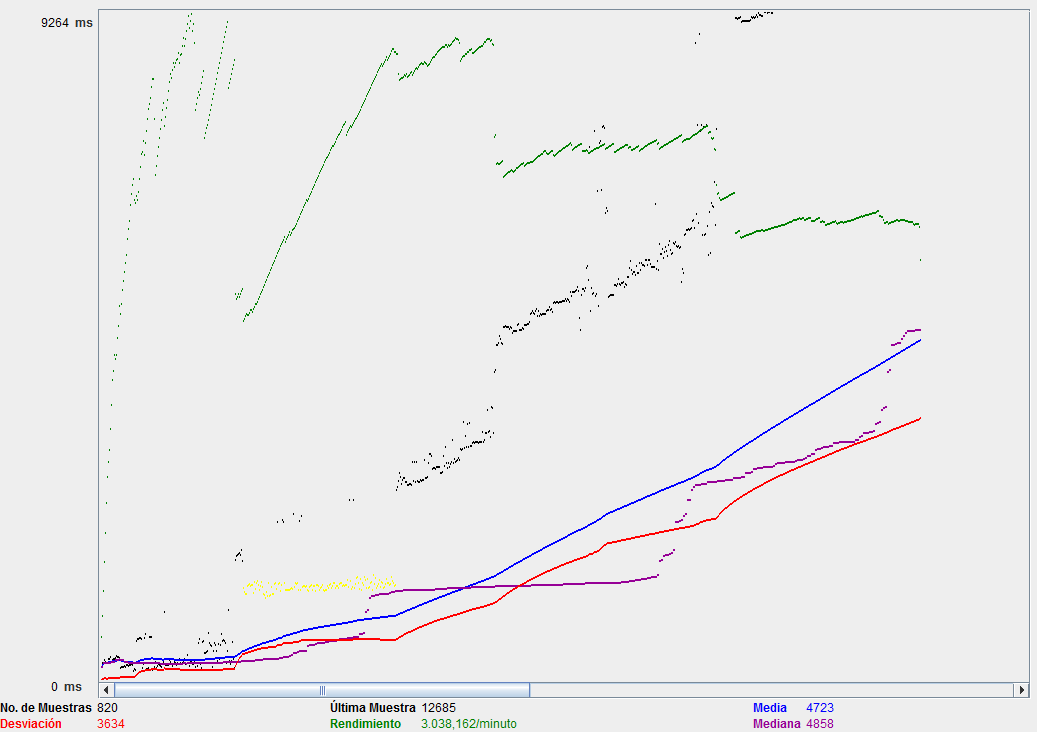
### Formato:

|  |  |
| --- | --- |
| Descripción del Experimento | |
| Título: Experimento 1 – 1 | ID: EXP 1 |
| Descripción: Validar las decisiones de diseño con respecto a la arquitectura del proyecto para cumplir con las metas de latencia *(“la arquitectura del sistema es capaz de procesar las funcionalidades de la aplicación en máximo 1 segundo”*) y escalabilidad *(“el sistema soporte hasta 13000 pacientes intentando sincronizar sus registros con el servidor principal en máximo 10 segundos”*. *“... determinar, mediante una gráfica, la curva de desempeño del sistema, indicando desde qué carga concurrente, el sistema deja de responder de manera adecuada.”*). | Responsable: StatusQuo |
| Propósito: | |
| ( ) Reparación, actualizar, clarificar | |
| (X) Obtener Información técnica | |
| ( ) Obtener información de negocio | |
| ( ) Otros: | |
|  | |
|  | |
| Propósito: Poner a prueba el primer escenario planteado, con el fin de conocer si la implementación planteada, es adecuada para los escenarios de calidad planteados (latencia y escalabilidad). | |
|
|
| Descripción del experimento: Se planea usar la primera configuración planteada, sobre una plataforma web que nos permita realizar peticiones al servicio de creación de pacientes, con el fin de realizar pruebas sobre los tiempos de respuesta del servidor, la cantidad de peticiones atendidas y el límite del servidor respecto a lo mencionado. | |
|
|
| Artefactos Creados: Se creó una implementación básica de una aplicación, con un servicio expuesto de creación de pacientes. Dicha aplicación cuenta con una base de datos, que soportara la operación. | |
|
|
| Criterio de terminación: No atención de solicitudes por parte del servidor, cuando se tenga al límite la infraestructura. | |
|
|
| Recursos Requeridos: Se requiere una máquina que soporte la operación, partimos de la configuración que se indicó anteriormente. | |
|
|
| Duración estimada: La duración estimada está sujeta al tiempo de ejecución de las pruebas, el tiempo de elaboración de los ambientes y de creación de los artefactos. | |
|
|
| Resultados del Experimento | |
| Resumen de los resultados: Se realizó la prueba, desplegando la aplicación en dos servidores locales en diferentes ubicaciones en la ciudad de Bogotá. Se realizaron pruebas locales y remotas, sin embargo en ninguna prueba se pudieron completar las 13000 solicitudes simultáneas, en todas las ocasiones el JMeter generaba una Excepción de memoria que obligaba a forzar el cierre sin obtener resultados. Para el ejercicio se esperaba 1 minuto de inactividad para detener manualmente la prueba. Logró ejecutar un total de 953 peticiones antes de dejar de responder. | |
|
|
| Duración Real: 3 minutos durante la prueba. 1 día para la creación e implementación de las aplicaciones y ambientes de prueba. | |
|
|
| Recursos Reales: 95% Servidor y 100% Cliente con JMeter | |
|
|
|
| Recomendaciones: Se deben utilizar otras técnicas como balanceador de carga y procesamiento en nodos paralelos. | |
|
|

### Resultados:

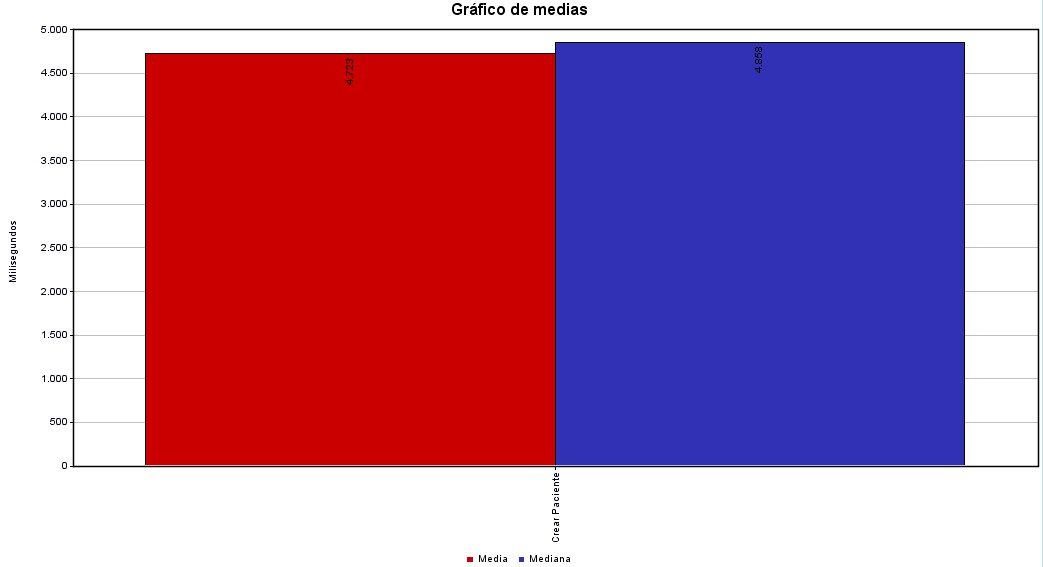
Los siguientes son los resultados obtenidos de la ejecución del experimento usando el patrón 1:

* Gráfico de resultados:

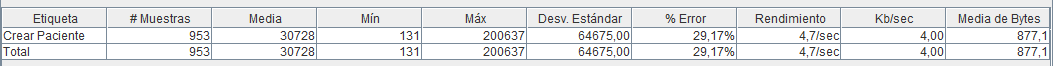


* Gráfico de medias:

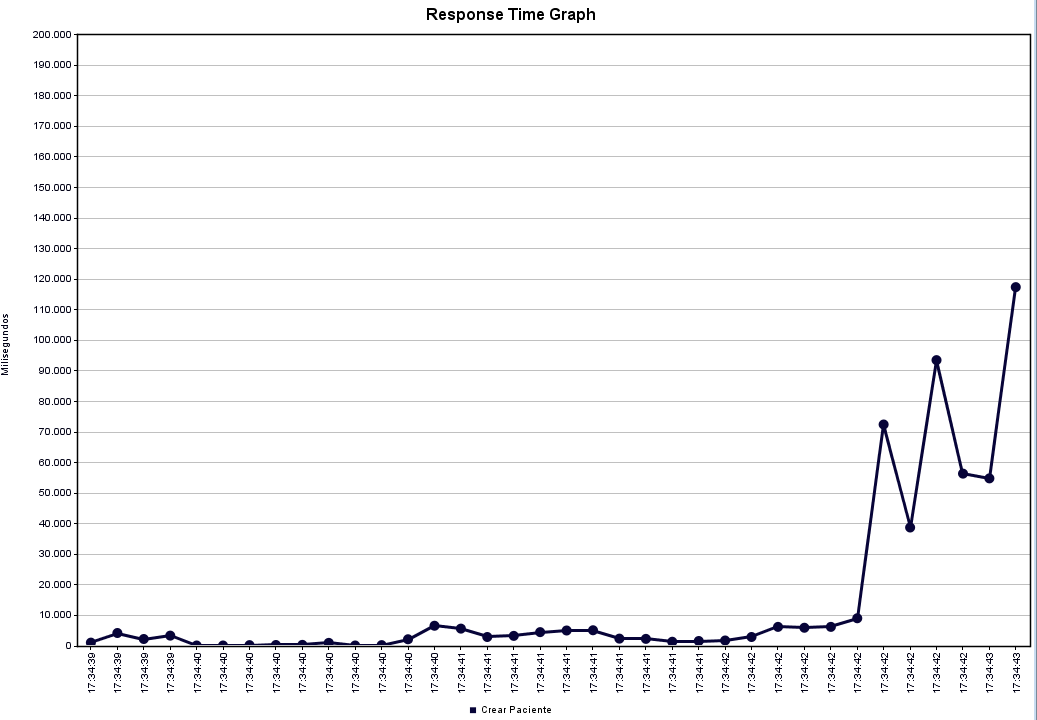




* Reporte resumen:



* Gráfico de tiempo de respuesta:



## Patrón 2:

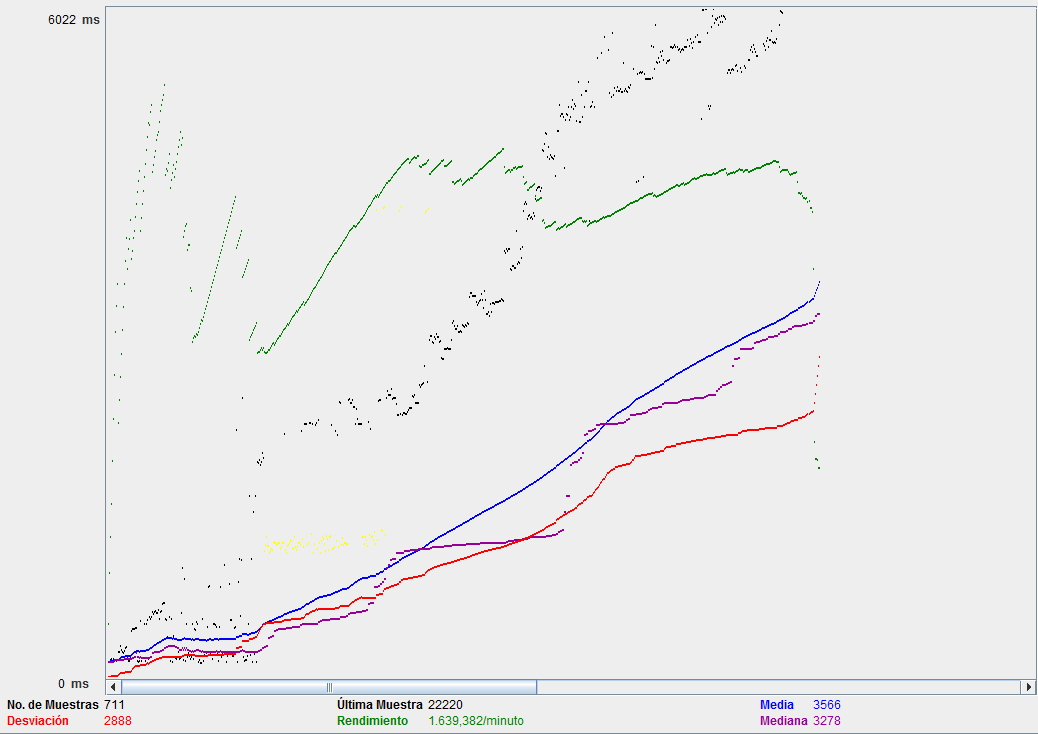
### Formato:

|  |  |
| --- | --- |
| Descripción del Experimento | |
| Título: Experimento 1 – 2 | ID: EXP 2 |
| Descripción: Validar las decisiones de diseño con respecto a la arquitectura del proyecto para cumplir con las metas de latencia *(“la arquitectura del sistema es capaz de procesar las funcionalidades de la aplicación en máximo 1 segundo”*) y escalabilidad *(“el sistema soporte hasta 13000 pacientes intentando sincronizar sus registros con el servidor principal en máximo 10 segundos”*. *“... determinar, mediante una gráfica, la curva de desempeño del sistema, indicando desde qué carga concurrente, el sistema deja de responder de manera adecuada.”*). | Responsable: StatusQuo |
| Propósito: | |
| ( ) Reparación, actualizar, clarificar | |
| (X) Obtener Información técnica | |
| ( ) Obtener información de negocio | |
| ( ) Otros: | |
|  | |
|  | |
| Propósito: Poner a prueba el primer escenario planteado, con el fin de conocer si la implementación planteada, es adecuada para los escenarios de calidad planteados (latencia y escalabilidad). | |
|
|
| Descripción del experimento: Se planea usar la segunda configuración planteada, sobre una plataforma web que nos permita realizar peticiones al servicio de creación de pacientes, con el fin de realizar pruebas sobre los tiempos de respuesta del servidor, la cantidad de peticiones atendidas y el límite del servidor respecto a lo mencionado. | |
|
|
| Artefactos Creados: Se creó una implementación básica de una aplicación, con un servicio expuesto de creación de pacientes. Dicha aplicación cuenta con una base de datos, que soportara la operación. | |
|
|
| Criterio de terminación: No atención de solicitudes por parte del servidor, cuando se tenga al límite la infraestructura. | |
|
|
| Recursos Requeridos: Se requiere una máquina que soporte la operación, partimos de la configuración que se indicó anteriormente. | |
|
|
| Duración estimada: La duración estimada está sujeta al tiempo de ejecución de las pruebas, el tiempo de elaboración de los ambientes y de creación de los artefactos. | |
|
|
| Resultados del Experimento | |
| Resumen de los resultados: Se realizó la prueba, desplegando la aplicación en dos servidores locales en diferentes ubicaciones en la ciudad de Bogotá. Se realizaron pruebas locales y remotas, sin embargo en ninguna prueba se pudieron completar las 13000 solicitudes simultáneas, en todas las ocasiones el JMeter generaba una Excepción de memoria que obligaba a forzar el cierre sin obtener resultados. Para el ejercicio se esperaba 1 minuto de inactividad para detener manualmente la prueba. Logró ejecutar un total de 711 peticiones antes de dejar de responder. | |
|
|
| Duración Real: 10 minutos durante la prueba. 1 día para la creación e implementación de las aplicaciones y ambientes de prueba. | |
|
|
| Recursos Reales: 100% Servidor y 100% Cliente con JMeter | |
|
|
|
| Recomendaciones: Se deben utilizar otras técnicas como balanceador de carga y procesamiento en nodos paralelos. | |
|
|

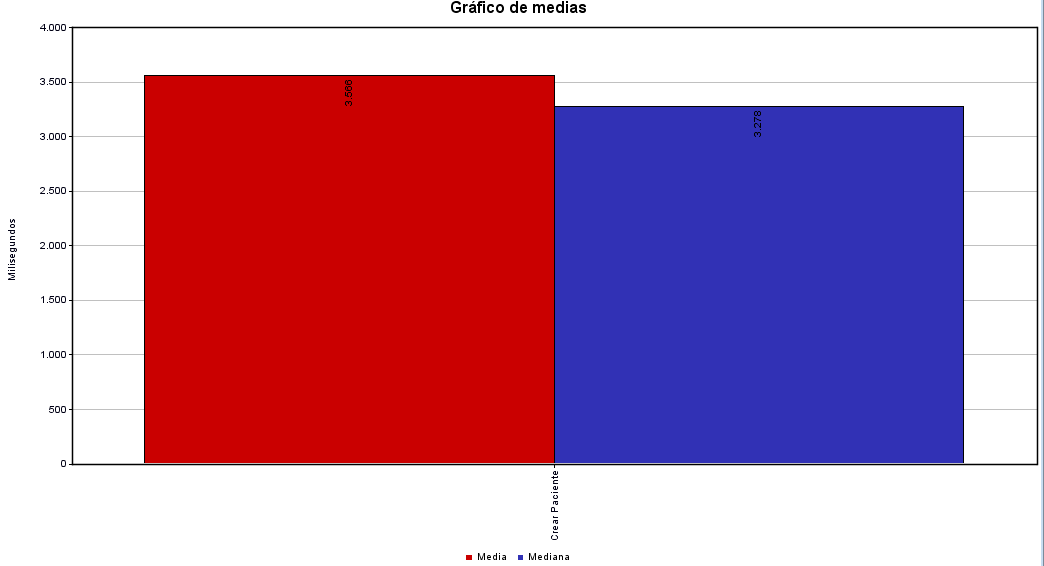
### Resultados:

Los siguientes son los resultados obtenidos de la ejecución del experimento usando el patrón 2:

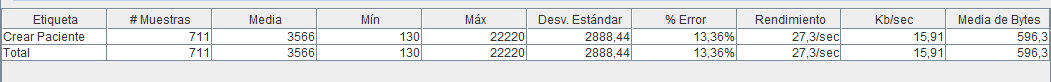
* Gráfico de resultados:



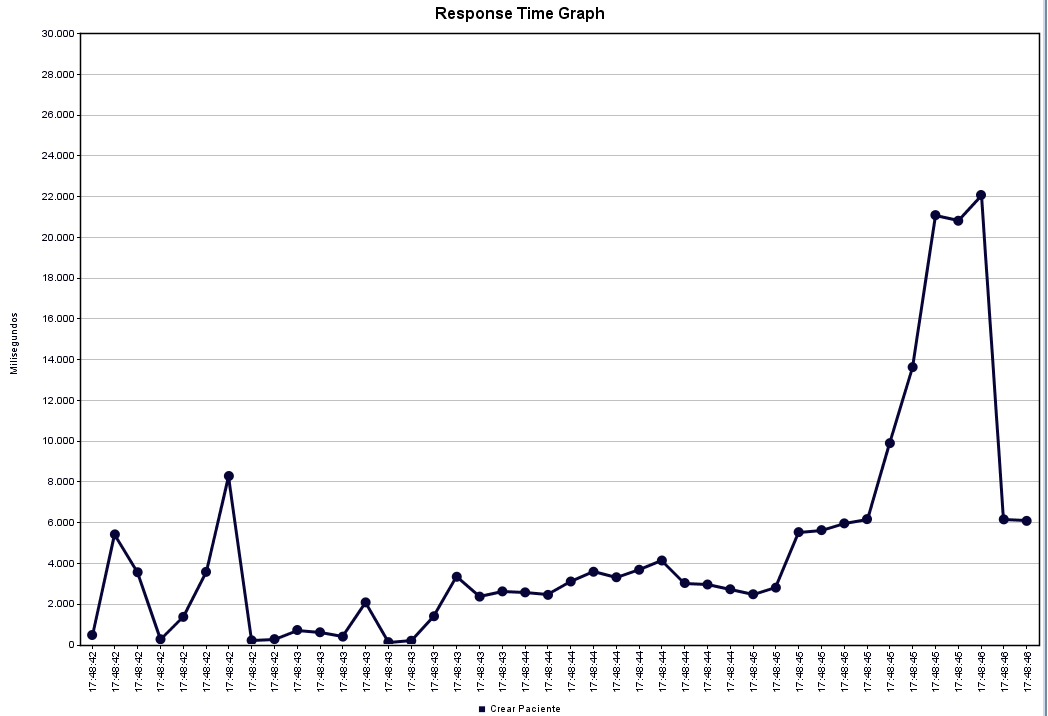
* Gráfico de medias:



* Reporte resumen:



* Gráfico de tiempo de respuesta:



## Patrón 3:

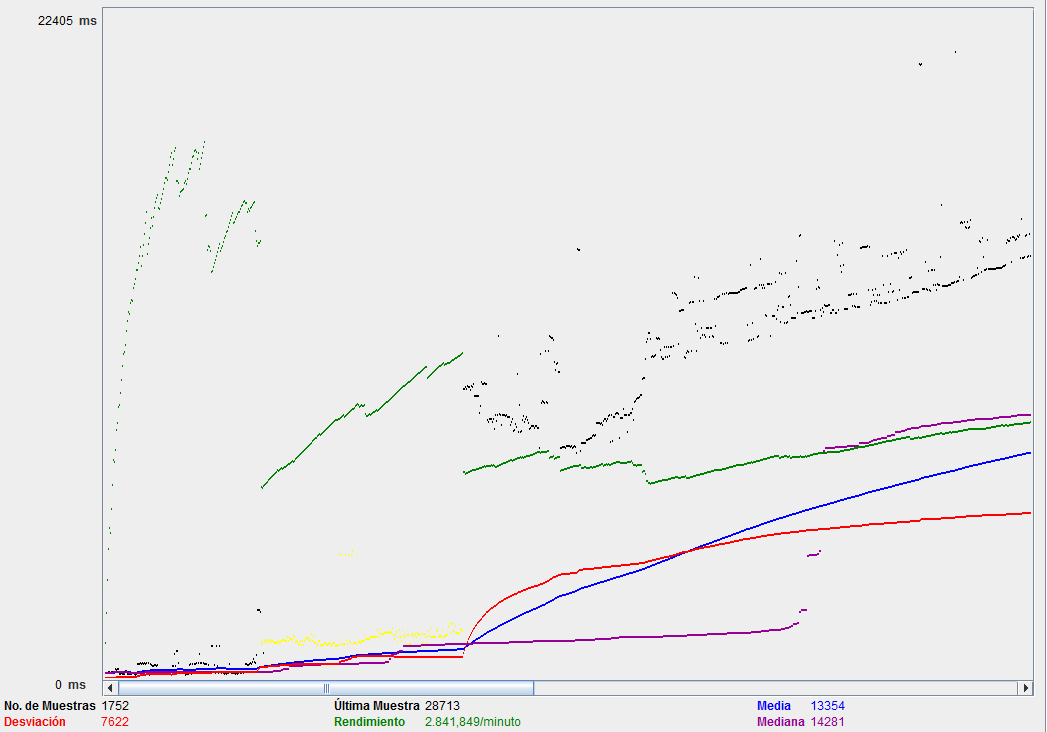
### Formato:

|  |  |
| --- | --- |
| Descripción del Experimento | |
| Título: Experimento 1 - 3 | ID: EXP 3 |
| Descripción: Validar las decisiones de diseño con respecto a la arquitectura del proyecto para cumplir con las metas de latencia *(“la arquitectura del sistema es capaz de procesar las funcionalidades de la aplicación en máximo 1 segundo”*) y escalabilidad *(“el sistema soporte hasta 13000 pacientes intentando sincronizar sus registros con el servidor principal en máximo 10 segundos”*. *“... determinar, mediante una gráfica, la curva de desempeño del sistema, indicando desde qué carga concurrente, el sistema deja de responder de manera adecuada.”*). | Responsable: StatusQuo |
| Propósito: | |
| ( ) Reparación, actualizar, clarificar | |
| (X) Obtener Información técnica | |
| ( ) Obtener información de negocio | |
| ( ) Otros: | |
|  | |
|  | |
| Propósito: Poner a prueba el tercer escenario planteado, con el fin de conocer si la implementación planteada, es adecuada para los escenarios de calidad planteados (latencia y escalabilidad). | |
|
|
| Descripción del experimento: Se planea usar la primera configuración planteada, sobre una plataforma web que nos permita realizar peticiones al servicio de creación de pacientes, con el fin de realizar pruebas sobre los tiempos de respuesta del servidor, la cantidad de peticiones atendidas y el límite del servidor respecto a lo mencionado. | |
|
|
| Artefactos Creados: Se creó una implementación básica de una aplicación, con un servicio expuesto de creación de pacientes. Dicha aplicación cuenta con una base de datos, que soportara la operación. | |
|
|
| Criterio de terminación: No atención de solicitudes por parte del servidor, cuando se tenga al límite la infraestructura. | |
|
|
| Recursos Requeridos: Se requiere una máquina que soporte la operación, partimos de la configuración que se indicó anteriormente. | |
|
|
| Duración estimada: La duración estimada está sujeta al tiempo de ejecución de las pruebas, el tiempo de elaboración de los ambientes y de creación de los artefactos. | |
|
|
| Resultados del Experimento | |
| Resumen de los resultados: Se realizó la prueba, desplegando la aplicación en dos servidores locales en diferentes ubicaciones en la ciudad de Bogotá. Se realizaron pruebas locales y remotas, sin embargo en ninguna prueba se pudieron completar las 13000 solicitudes simultáneas, en todas las ocasiones el JMeter generaba una Excepción de memoria que obligaba a forzar el cierre sin obtener resultados. Para el ejercicio se esperaba 1 minuto de inactividad para detener manualmente la prueba. Logró ejecutar un total de 1752 peticiones antes de dejar de responder. | |
|
|
| Duración Real: 2 minutos durante la prueba. 1 día para la creación e implementación de las aplicaciones y ambientes de prueba. | |
|
|
| Recursos Reales: 90% Servidor y 100% Cliente con JMeter | |
|
|
|
| Recomendaciones: Se deben utilizar otras técnicas como balanceador de carga y procesamiento en nodos paralelos. | |
|
|

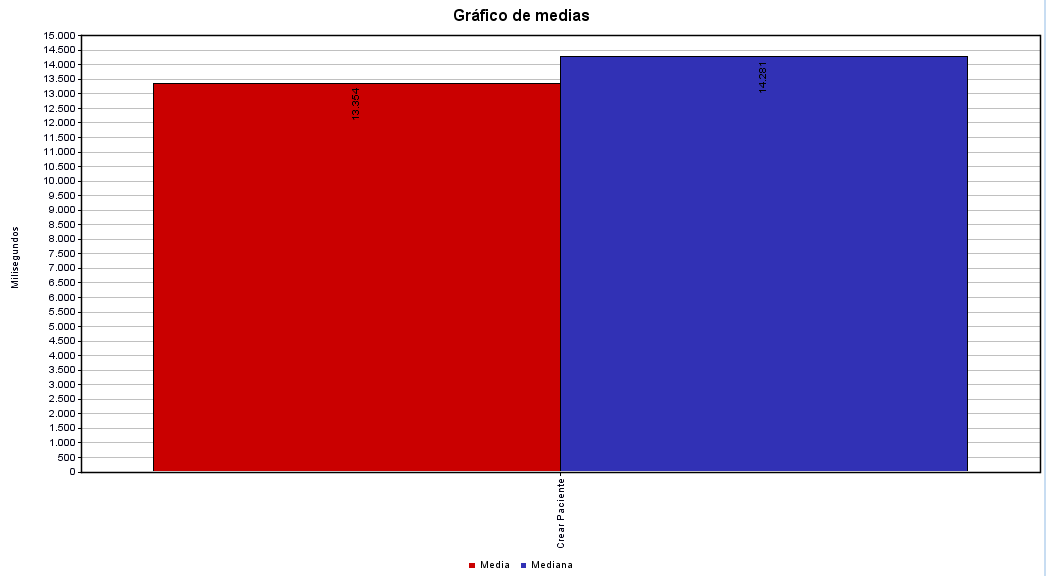
### Resultados:

Los siguientes son los resultados obtenidos de la ejecución del experimento usando el patrón 2:

* Gráfico de resultados:



* Gráfico de medias:



* Reporte resumen:



* Gráfico de tiempo de respuesta:

