*Grupo proganation*

Iván Arturo Salazar; Juan Sebastián Palacios

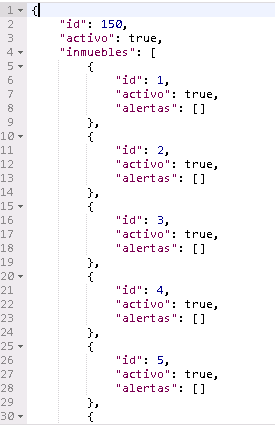
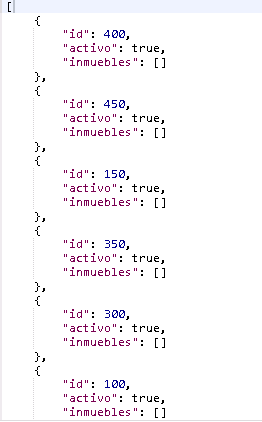
Sebastián Ramos; Salomón Heim Jaramillo

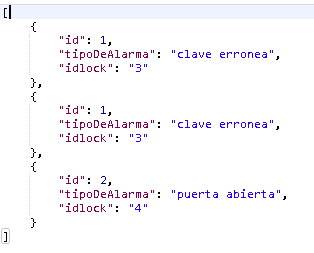
*Arquitectura y diseño de Software*

***Entrega final experimento 1:***

*Escenario de pruebas:*

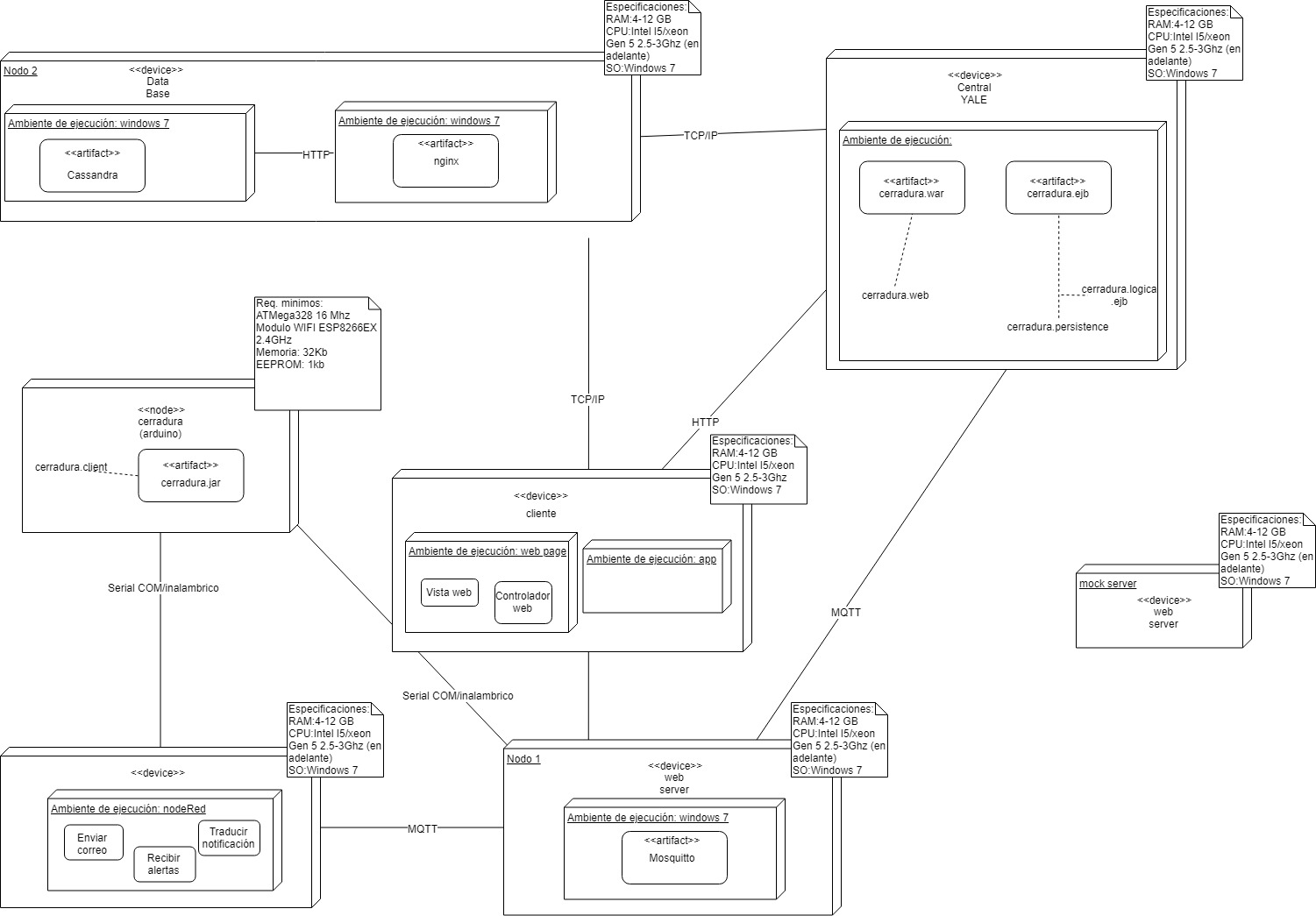
Se desarrollaron los siguientes escenarios de pruebas en la base de datos de cada integrante siguiendo una ruta tipo: unidadResidencial/inmueble/Alertas





Realizamos comandos POST desde una colección en Postman de tal forma que tuviéramos datos persistidos para realizar las pruebas pertinentes.

*Diagrama de despliegue:*



*Comparación resultados:*

*Satisfacción escenarios de calidad:*

Los escenarios de calidad que cumplimos se relacionaron a realización de:

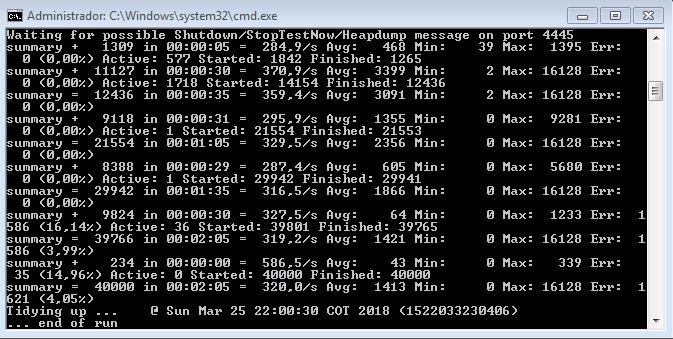
* Desempeño en las pruebas resueltas:
* Realizar servicios REST: Las consultas que logramos realizar cumplen con las enunciadas por el proyecto, pues pudimos realizar los métodos GET y POST en cada una de las entidades necesarios (unidades residenciales, inmuebles y alarmas), además de poder realizar un servicio de DELETE sobre las unidades residenciales.

Para el desarrollo futuro del proyecto debemos tener en cuenta la realización del cluster para completar las pruebas de JMETER, pues el limite de este no lograba abarcar la totalidad de pruebas de carga que los requerimientos pedían, cuando lo montamos sobre una sola máquina. También tenemos que mejorar la implementación funcional del balanceador de carga, el cual nos seguirá sirviendo para mejorar el desempeño en estas pruebas, además de ayudarnos a cumplir escenarios de escalabilidad necesarios.

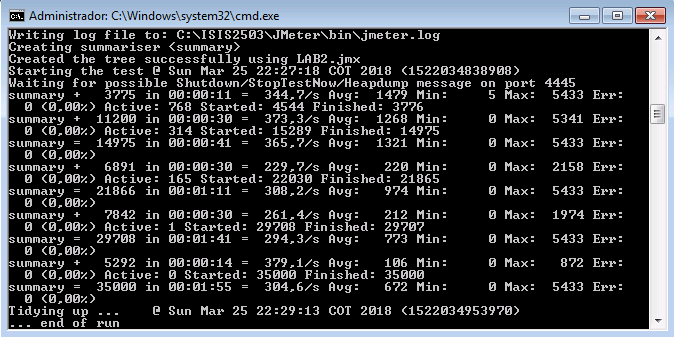
*Pruebas de carga:*

Para las pruebas de carga decidimos utilizar un balanceador de carga que nos permitiera enviar un porcentaje de las pruebas por máquina y así evitar que JMeter no soportara todas las peticiones. Para esto, utilizamos tópicos para distinguir las diferentes máquinas que suban las pruebas (en este caso 2) y configuramos la petición para que envie mas de una alerta por petición en caso de que sean lanzadas desde la misma unidad residencial, diferenciando el inmueble de cada alerta para evitar errores o confusión al momento de persistir los datos.

*Máquina 1:*



*Máquina 2:*

**

Como se puede evidenciar la máquina 1 al tener mayor número de peticiones tiene un tiempo mayor y un porcentaje de error diferente a 0 (4.05%), pero es mucho mejor a si usaramos una sola máquina donde no son capaces de soportar todas las peticiones o generan un error mayor al 50%.