**Documentación Entrega Final – Experimento 3**

**Grupo: 10**

**Pre-experimentación:**

**Problemática:** Hipótesis:El experimento es realizado en JAX-RS con una capa de servicios REST que soporta 1000 peticiones con información de los 3 sensores y tiene un tiempo de respuesta menor a 1000 ms, los cuales pueden ser consumidos desde aplicaciones web y móviles.

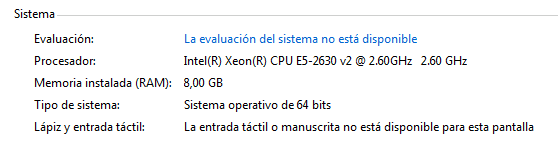
**Objetivo:** El objetivo de este experimento es aplicar patrones de diseño que favorezcan los atributos de calidad de modificabilidad y usabilidad. Desarrollar las aplicaciones móviles para los actores del sistema que así las requieran.

**Descripción del experimento:** El experimento consiste en implementar 3 tácticas de arquitectura de software vistas en la sesión magistral las cuales garantizan que el proyecto sea modificable, tenga una alta interoperabilidad y usabilidad. Finalmente se desea desarrollar una aplicación web y aplicaciones móviles para los usuarios que lo requieran.

**Recursos de la experimentación:** Los recursos descritos a continuación son iguales tanto en las máquinas que prestan los servicios del sistema y las del balanceador de carga.

Maquina Fisica:

Hardware:



Software:

Windows 7, JDK8, Glashfish 4.1.0.

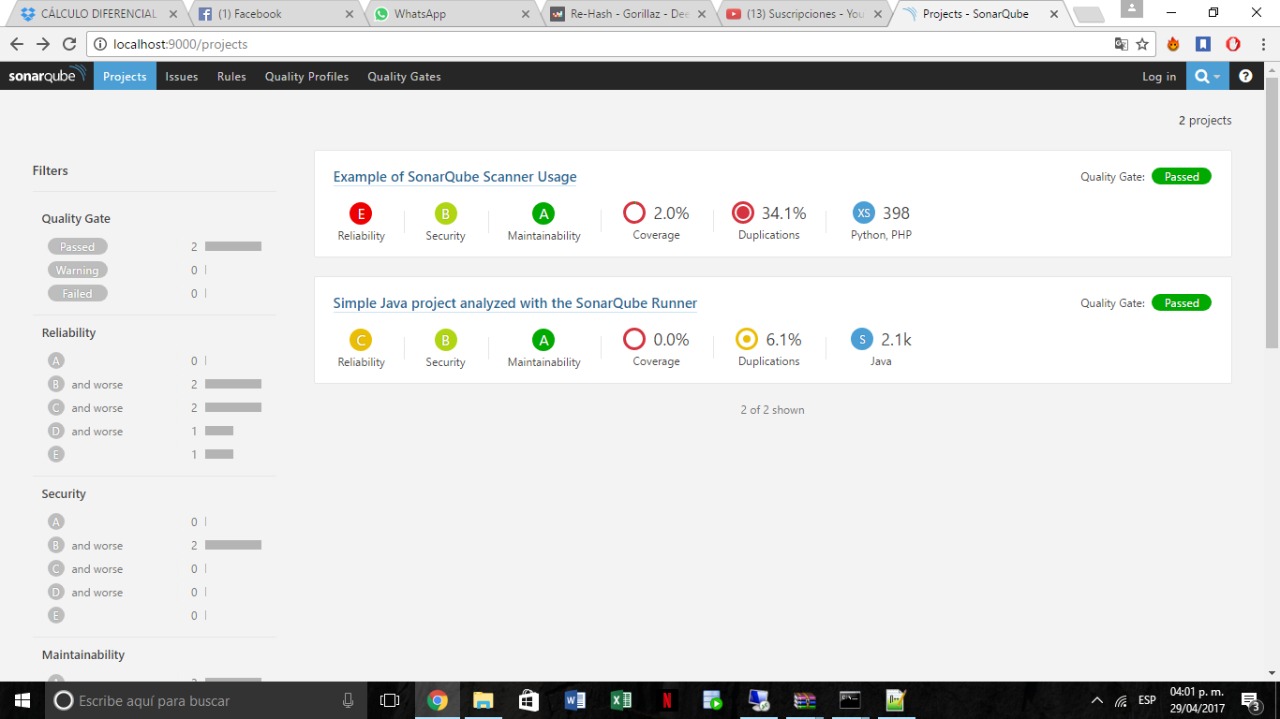
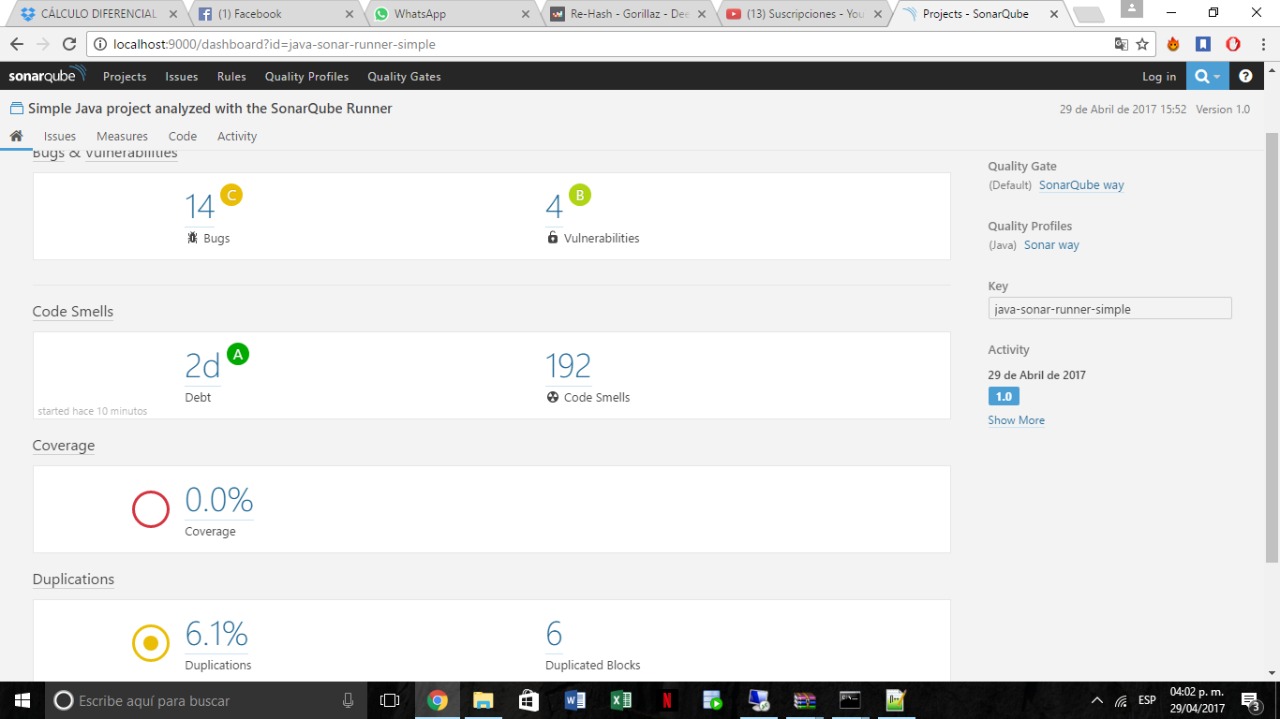
Dispositivo Movil:

Android 4.0 o mayores

**Resultados esperados:** Se espera que el experimento supere las pruebas y que la hipótesis sea verdadera.

**Post-experimentación**

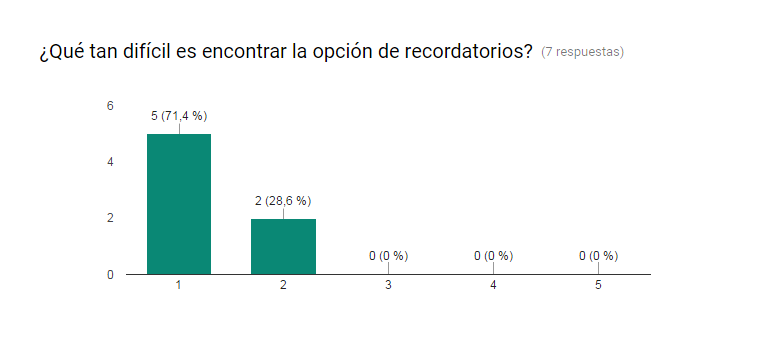
**Análisis de métricas antes y después de aplicar patrones**

****

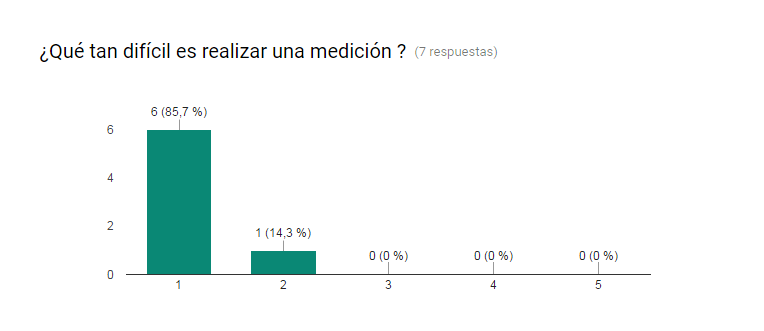
**Instrumento de crítica constructiva**

El instrumento diseñado para recibir una crítica constructiva por los usuarios fue una encuesta virtual; la cual califica cada una de las opciones ofrecidas en la aplicación, a continuación se muestra un análisis de los resultados obtenidos y además el link a la encuesta realizada: <https://goo.gl/forms/F4vQ32e39Zd9VbHI3>

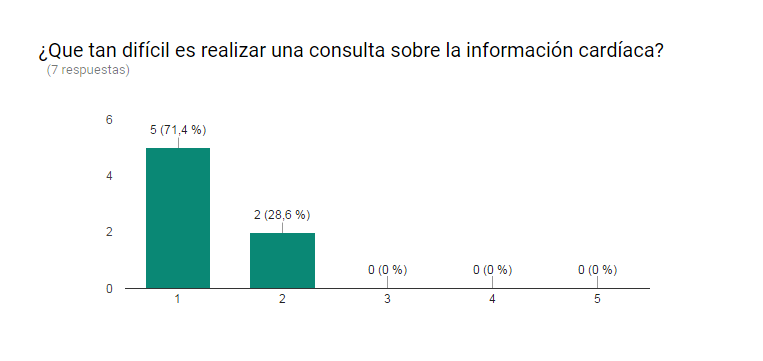
Esta encuesta fue realizada a 7 posibles pacientes que podrían estar interesados en usar la aplicación.

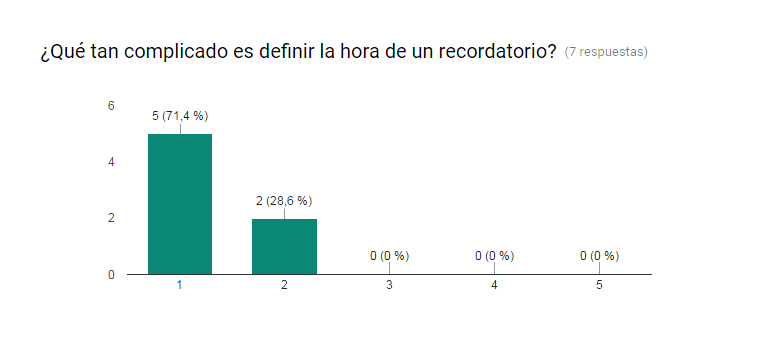


**P1:** En general es muy facil acceder a la opcion de recordatorios.

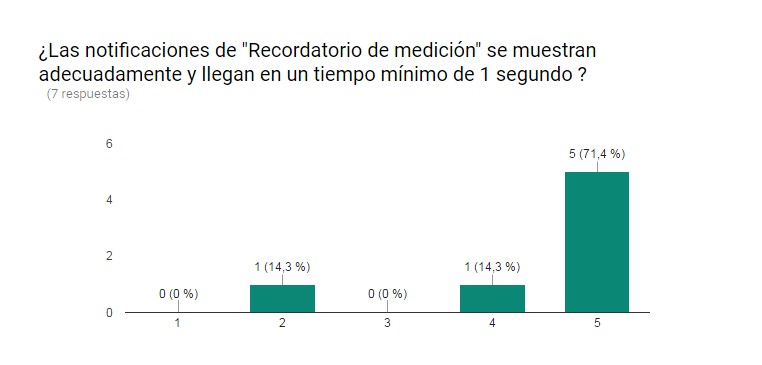


**P2:** En general es muy facil acceder a la opcion de realizar una medición.

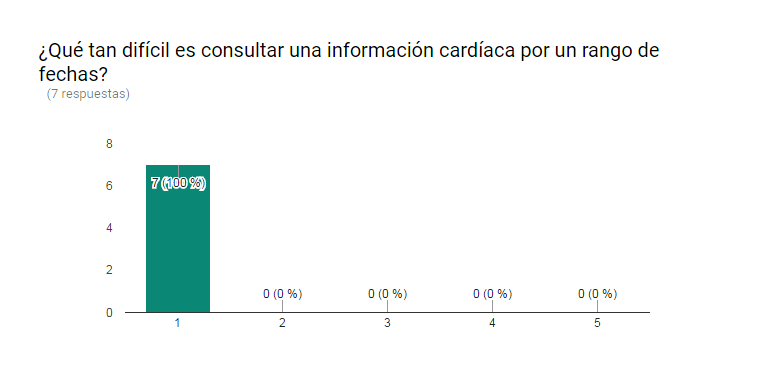
 **P3:** En general es muy facil acceder a la opcion de consultar la información cardíaca.



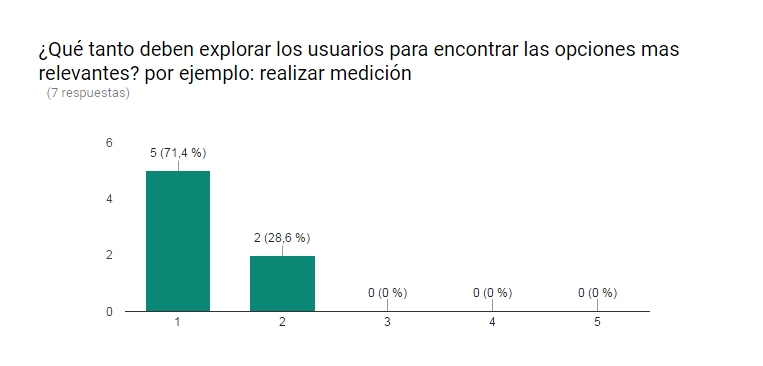
**P4:** En general es muy facil acceder a la opcion para definir la hora de un recordatorio.



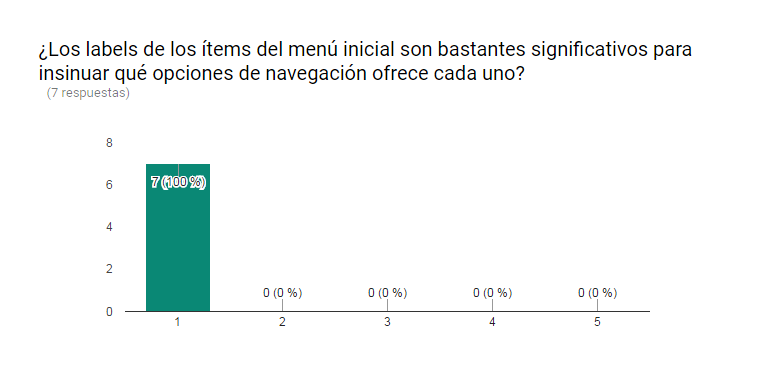
**P5:** En general las notificaciones en el celular recordando la medicion que debe realizar el usuario cumple con las espectativas, se muestran adecuadamente y en menos de un segundo.



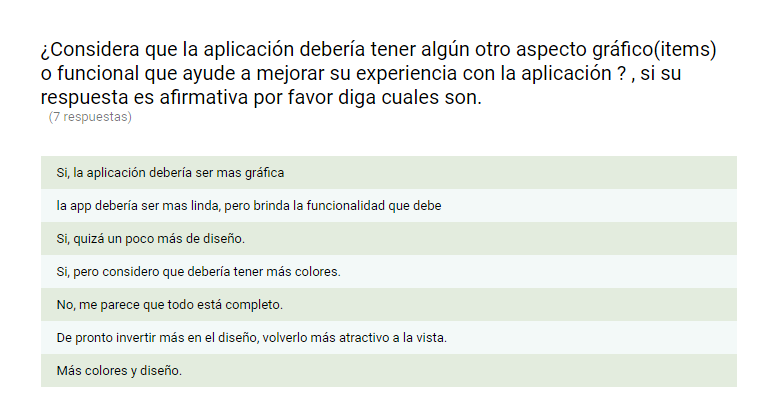
**P6:** En general consultar una información cardíaca por un rango de fechas es muy facil dado que se utiliza un calendario que facilita la usabilidad del cliente al quitarle el peso de escribir por simplemente seleccionar.



**P7:** En general los usuarios no deben explorar mas de 3 niveles para encontrar las opciones mas relevantes que deseen usar en un determinado momento.



**P8:** En general los labels de los items del menú inicial son bastantes significativos para insinuar que opcion ofrece cada uno para el usuario.



**P9:** En general las recomendaciones que proponen los usuarios son en la mejora grafica, es decir, darle mas estilo a la aplicación para que su aspecto visual sea mas agradable, no refutaron nada sobre la funcionalidad, lo cual hace pensar que cumple con lo necesario para el usuario final.

**Descripción de la Arquitectura y Proceso de desarrollo**

Con base en el laboratorio de escalabilidad y desempeño decidimos utilizar JAX-RS porque en términos de procesamiento de peticiones fue más ágil. Además, implementamos beans sin estado con el objetivo de favorecer la velocidad del sistema y procesar las 1000 peticiones en menos de un segundo; esto se debe a que el sistema no se bloquea si una petición no responde rápidamente por parte del cliente y puede seguir procesando más solicitudes al tiempo.

El sistema cuenta con una base de datos relacional provisionada por Amazon, la cual garantiza la persistencia de los datos del sistema y a su vez la duplicación de la misma, esta fue creada con su mayor normalización, es decir, un 70% de las relaciones creadas están en tercera forma normal.

Se implementó un balanceador de carga con el objetivo de aumentar la disponibilidad del sistema, pero esta ganancia fue castigada con la implementación de la seguridad, ya que reduce el desempeño, la modificabilidad, la usabilidad y la disponibilidad del sistema.

Finalmente para el último experimento se implementó las siguientes tácticas: La primera táctica es **Builder**; se implementó en las siguientes clases: Alerta, Configuración, Consejo, Dispositivo, Historial, Hospital y Medico. La segunda táctica es **Observer**; se implementó en la clase Configuración.