Experimento 3.

Bibiana Gamba - 201423538
Juan Manuel Domínguez - 201532819
Nicolás Acevedo - 201530726
Nicolás Cabrera - 201530947
Juan Camilo Sánchez - 201519337

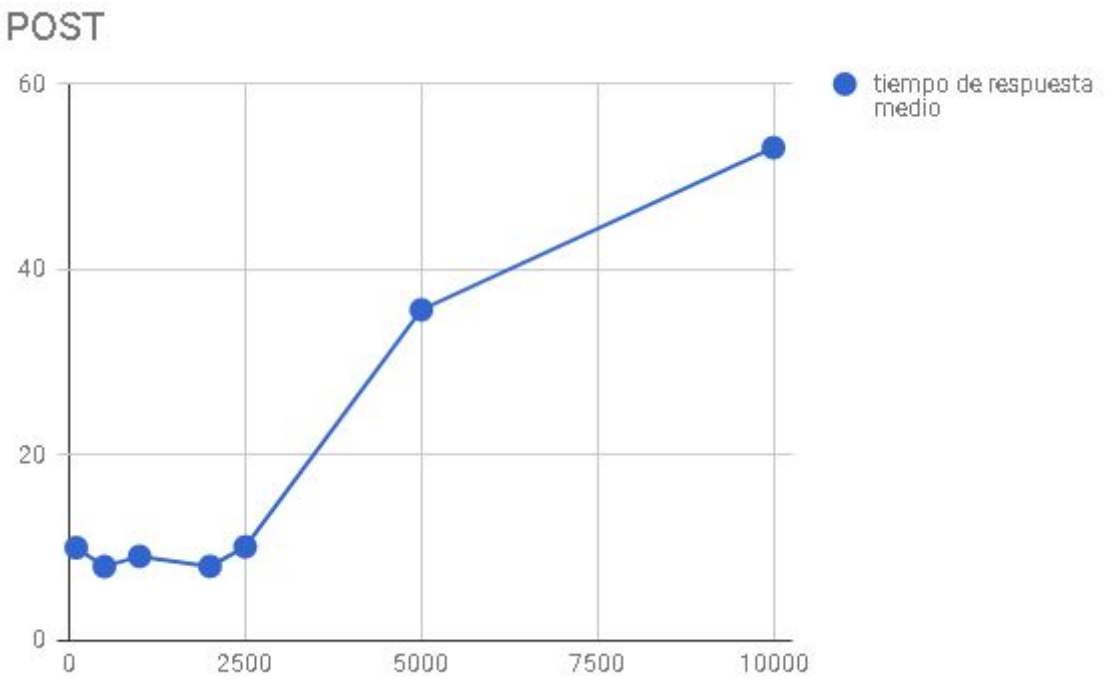
Kelly Garces
Arquitectura y Diseño de Software

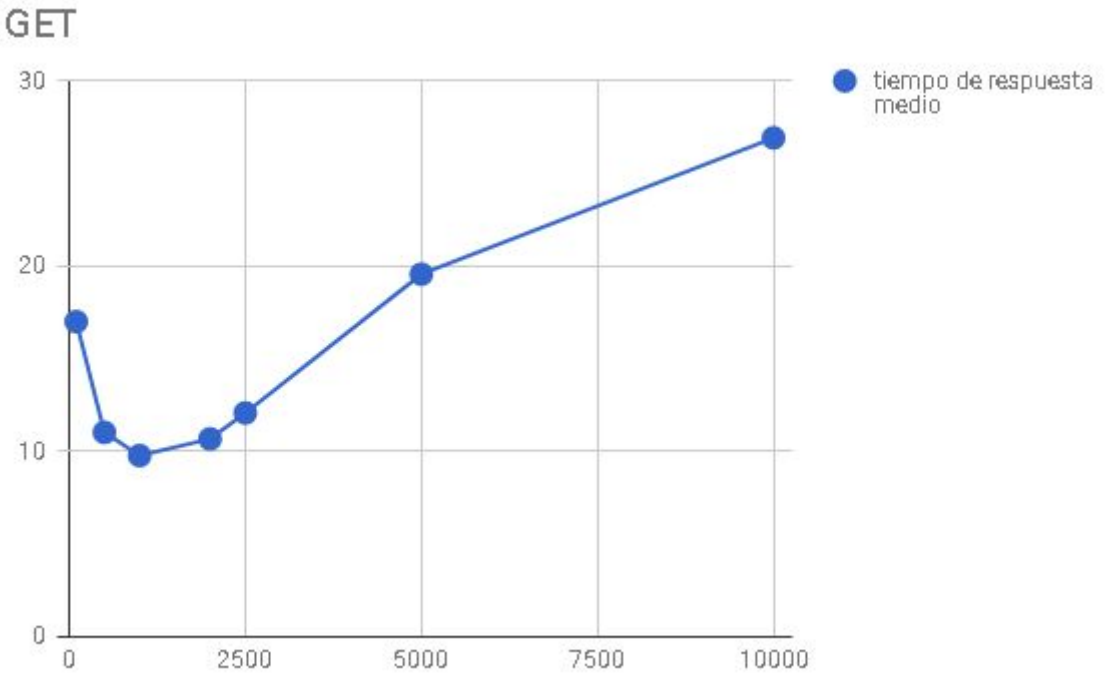
Universidad de los Andes
7 de diciembre del 2017

- Análisis y comparación de los resultados de las pruebas de carga de la aplicación con y sin uso de las tácticas de redundancia. El reporte debe incluir descripción de los escenarios de experimentación, gráficos para soportar el análisis y conclusiones.

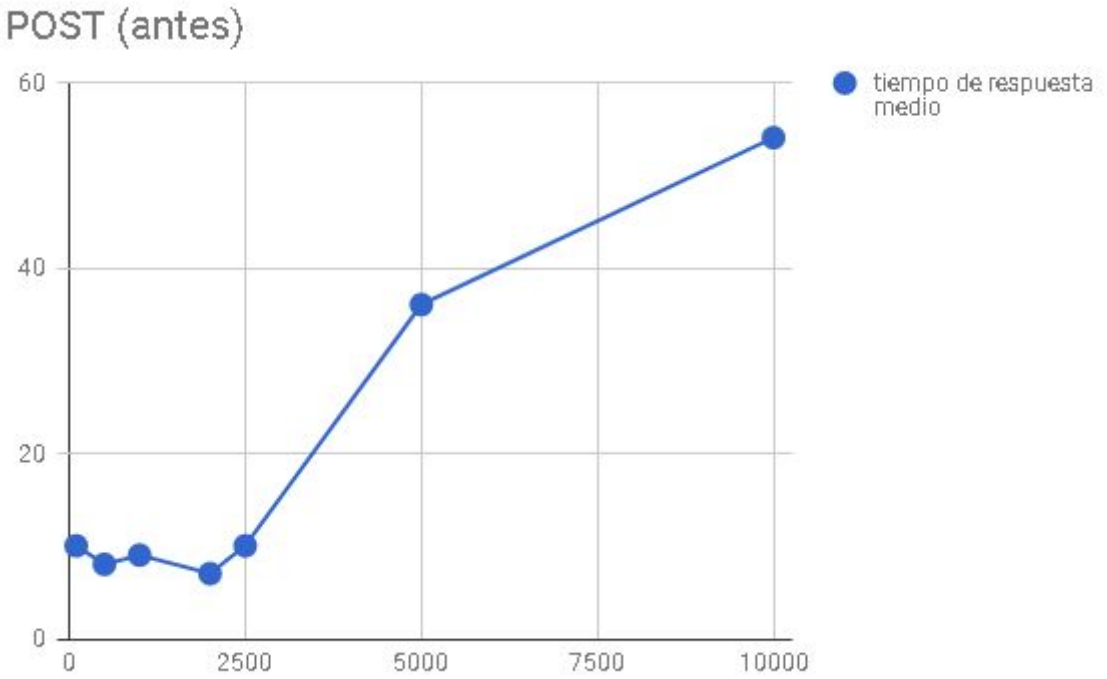
Al utilizar tácticas de redundancia, pudimos observar que hubo un aumento en el desempeño y el número de solicitudes que podía soportar la aplicación. Esto se debe a que al usar la redundancia en Kafka y configurando zookeeper pudimos distribuir las diferentes solicitudes que se hacen. De esta manera, el número de errores también disminuye cuando se tienen más threads ya que estan 3 Kafkas disponibles para responder a estas solicitudes. Además, para tener una disponibilidad más alta, estos están situadas en máquinas diferentes para que en caso que falle alguna las otras sigan respondiendo las solicitudes. Esto se ve tanto para el GET como para el POST.

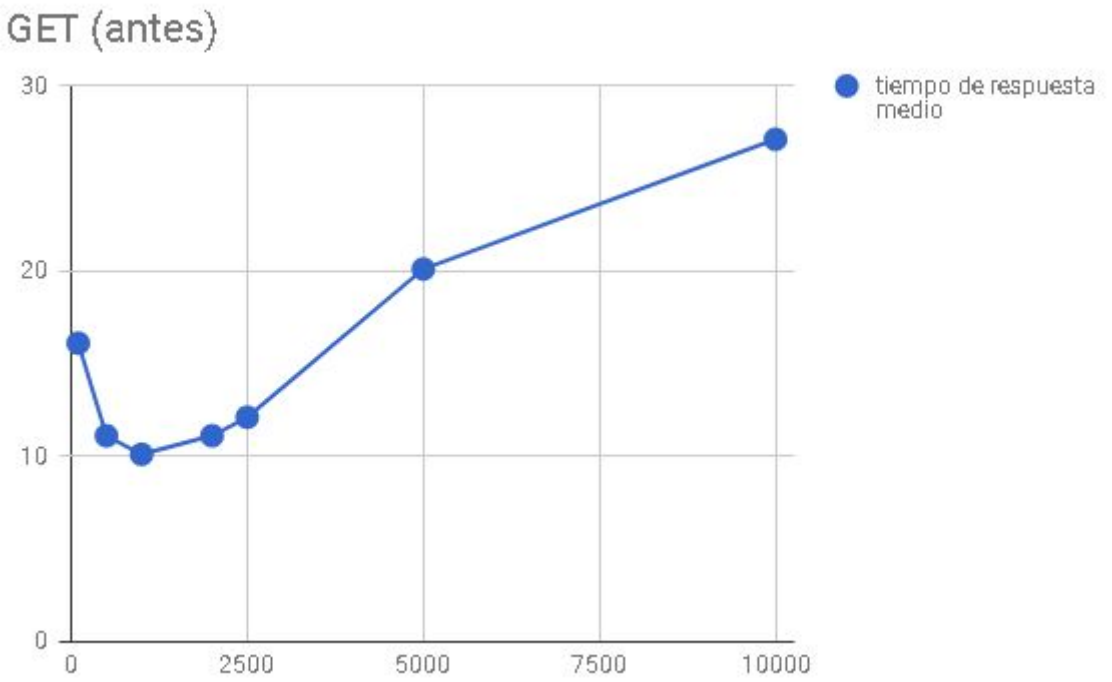
Con disponibilidad





Sin disponibilidad



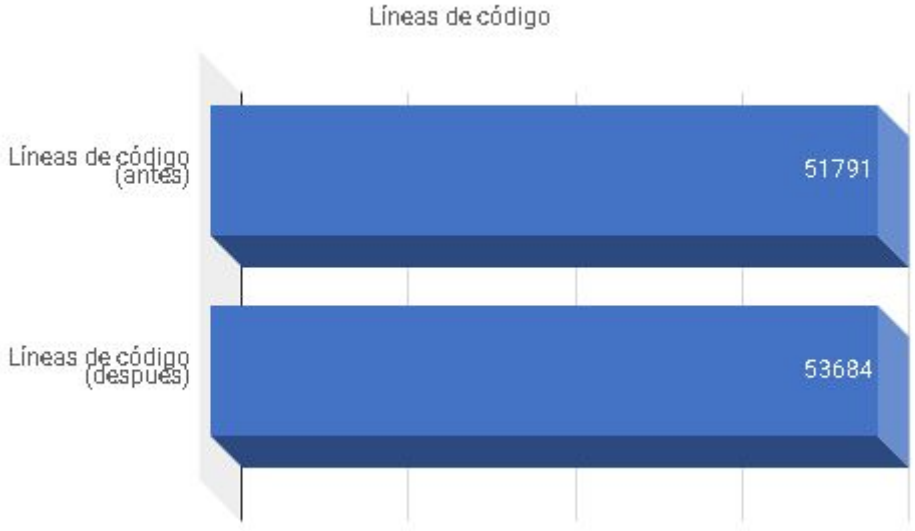


- Análisis de métricas (p.ej., las que da sonarqube) antes y después de aplicar patrones

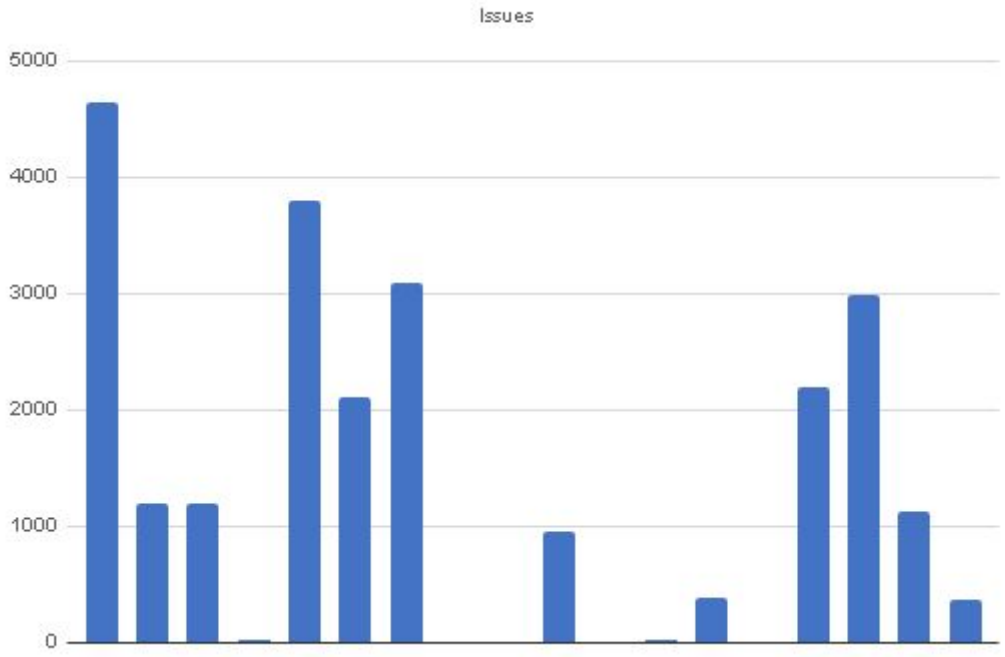
Para las métricas utilizamos ebertApp, donde pudimos observar que al implementar los patrones de modificabilidad tuvimos mejores resultados en muchas de las métricas de las que teníamos información. Las principales diferencias que se encontraron en toda la clase de BUGs y issues acerca del código en toda la arquitectura (Front y Back). Sin embargo estas mejoras fueron de alrededor del 5% y máximo del 10%. Como se puede ver en las gráficas y tablas que se incluyen en el documento. Hubo otros aspectos que se conservador igual, ya que el uso de patrones no afectaba esta parte del código. Por último, lo más beneficio del uso de patrones fue la duplicación de código que bajo se forma significativa.

En cuanto a las líneas de código, no hubo diferencias significativas debido a la naturaleza de los patrones que utilizamos.

Líneas de código



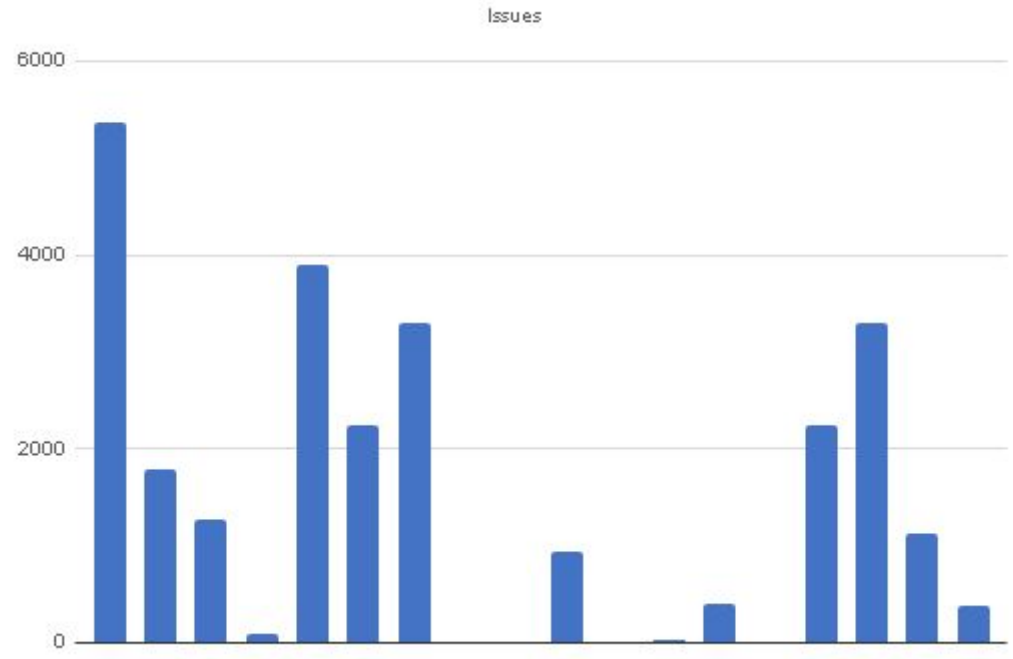
Issues de EbetApp para el Experimento 3 (sin patrones)



Quick Fixes Total	4653
Bug Risk Issues	1206
Compatibility	1201
Duplication Issues	21
Style Issues	3800
Coffe Script Issues	2105
CSS Issues	3097

HTML Issues	5
Java Issues	11
Javascript Issues	948
Less Issues	2
Markdown Issues	33
Shell Issues	383
XML Issues	7
CoffeeLintIssues	2207
CSSLintIssues	2987
CodeClimateIssues	1131
ShellCheckIssues	367
Lineas de codigo	51791

Issues de EbetApp para el Experimento 3 (con patrones)



Quick Fixes Total	5369
Bug Risk Issues	1787
Compatibility	1264
Duplication Issues	96
Style Issues	3907
Coffe Script Issues	2242
CSS Issues	3305
HTML Issues	5
Java Issues	13
Javascript Issues	951

Less Issues	2
Markdown Issues	31
Shell Issues	393
XML Issues	7
CoffeeLintIssues	2235
CSSLintIssues	3305
CodeClimateIssues	1125
ShellCheckIssues	389
Lineas de codigo	53684