DP2 2024

Acme Software Factory

Repositorio: https://github.com/DP2-2024-C1-029/Acme-Software-Factory.git

Miembro:

* Jaime Varas Cáceres (jaivarcac@alum.us.es)

Tutor: José González Enríquez

GRUPO C1.029

25/05/2024

Versión 1.0

Índice

[Historial de versiones 3](#_Toc167744920)

[Capítulo 1 – Pruebas funcionales 4](#_Toc167744921)

[Code Audit 4](#_Toc167744922)

[Audit Record 6](#_Toc167744923)

[Capítulo 2 – Pruebas de desempeño 9](#_Toc167744924)

[Bibliografía 11](#_Toc167744925)

# Historial de versiones

|  |  |  |  |
| --- | --- | --- | --- |
| Fecha | Versión | Descripción | Entrega |
| 27/05/2024 | V1.0 | Inicio del documento | D04 |
|  |  |  |  |

# Capítulo 1 – Pruebas funcionales

## Code Audit

Tras ejecutar todos los test, se puede observar que para las codeAudit se cubre el 95.5%, valor que está por encima de la recomendación mínima del 90% que debería cubrir al menos todos los test.

Texto

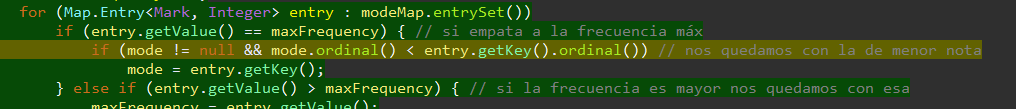
Descripción generada automáticamente

En primer lugar, para no repetirlo durante todo el documento, se va a comentar que las líneas que los “assert” siempre aparecen en amarillo, y que el “status” tampoco se puede poner en verde, porque hay un caso que nunca se puede probar.

Imagen que contiene Texto

Descripción generada automáticamente

Voy a empezar hablando por del Publish, en el que tenemos un condicional que no ha podido ser probado en todos los casos, esto aparece en todos los servicios en los que se calculan la moda de la notas:



Para el update service podemos observar que todo está en verde, menos una rama extra del cálculo de la moda:

Texto

Descripción generada automáticamente con confianza baja

Texto

Descripción generada automáticamente

En el ListService tenemos una línea en amarillo que corresponde a la internacionalización del valor del modo borrador:

Texto

Descripción generada automáticamente

En el controlador, todo está verde:

Texto

Descripción generada automáticamente

El resto de los servicios que no han sido nombrados es porque están cubiertas o las faltas de coberturas ya han sido mencionadas en otros servicios.

## Audit Record

En esta ocasión, no superamos el umbral del 90% por un 0.1%, pero se a unas líneas en el deleteService correspondientes al unbind que no se usan nunca y que se han dejado por seguir el estilo de AcmeJobs.

Pantalla de computadora con letras

Descripción generada automáticamente con confianza media

A continuación, el unbind del delete:

Texto

Descripción generada automáticamente

El update, a excepción de las dos líneas del authorise que están en amarillo siempre, esta todo en verde:

Texto

Descripción generada automáticamente

El list, tiene un par de líneas de las que no se han podido cubrir todas las ramas, una es el idioma y la otra es una comprobación para mostrar un botón:

Texto

Descripción generada automáticamente

Por último, el controlador y el resto de servicios no tienen nada que destacar ya que están cubiertos en su gran mayoría.

Pantalla de computadora con letras

Descripción generada automáticamente con confianza baja

Como conclusión se puede sacar que todo el “CodeAudit” ha sido probado de manera muy exhaustiva, probando todas las validaciones posibles. Al final de este capítulo se muestran capturas de un bloc de notas usado para la realización de los tests.

Texto

Descripción generada automáticamente

Para terminar este capítulo me gustaría comentar que he usado el Excel de SampleData para la realización de los tests .safe de los create, update, y publishl

# Capítulo 2 – Pruebas de desempeño

El desarrollo del software se ha ejecutado durante todo el cuatrimestre en mi PC personal. Obteniendo los resultados de ejecutar el replayer en eclipse, nos genera una batería de datos, los cuales, analizándolos mediante las técnicas enseñadas en clase, hemos podido obtener resultados claros.

Vamos a empezar por los promedios de los resultados de búsqueda.

Gráfico

Descripción generada automáticamente

Como se puede observar, hay alguna ruta que tiene peticiones que han tardado más, lo que es un resultado muy bueno, porque nos indica que las búsquedas se realizan de manera rápida. En este apartado tengo que comentar que este replayer ha sido realizado tras implementar índices y optimizar funciones ya que ya realicé análisis de los tests antes de este test, pero tuvieron que repetirse por un cambio en la base de datos. Estas optimizaciones eran principalmente streams que filtraban colecciones que podían ser filtradas en las queries del repositorio.

A continuación, vamos a observar el intervalo de confianza del PC personal frente al del PC 2 obtenido de una simulación con función random:

Tabla

Descripción generada automáticamente

Con estos datos se ha realizado un Z-Test, el cual se muestra a continuación.

Tabla

Descripción generada automáticamente

Podemos observar que Alpha es 0.05, y que el p-value es 0.438… por lo que podemos decir que los cambios **no** dieron como resultado ninguna mejora significativa; los tiempos de muestreo son diferentes, pero son globalmente iguales.

**También se ha replicado estas pruebas en otro ordenador** (PC2 – características similares) y he obtenido los siguientes resultados:

Tabla

Descripción generada automáticamente

Podemos observar que Alpha es 0.05, y que el p-value es 0.19… por lo que podemos decir que los cambios no dieron como resultado ninguna mejora significativa; los tiempos de muestreo son diferentes, pero son globalmente iguales.

Como **conclusión**, ninguno de los PCs muestra una diferencia significativa en el rendimiento (antes y después) a un nivel de variación del 5%. Por lo tanto, las diferencias observadas en las medias no son estadísticamente significativas, lo que sugiere que ninguno de los PCs es concluyentemente más rápido o lento que el otro según los datos proporcionados.

# Bibliografía

Diapositivas de Diseño y Pruebas 2 – Universidad de Sevilla.