DP2 2024

Acme Software Factory

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

Miembro:

* José María Portela Huerta (josporhue@alum.us.es)

Tutor: José González Enríquez

GRUPO C3.029

22/10/2024

Versión 3.0

Índice

[Historial de versiones 3](#_Toc167536083)

[Capítulo 1 – Pruebas funcionales 4](#_Toc167536084)

[Training Module 4](#_Toc167536085)

[Training Session 8](#_Toc167536086)

[Capítulo 2 – Pruebas de desempeño 14](#_Toc167536087)

[Bibliografía 16](#_Toc167536088)

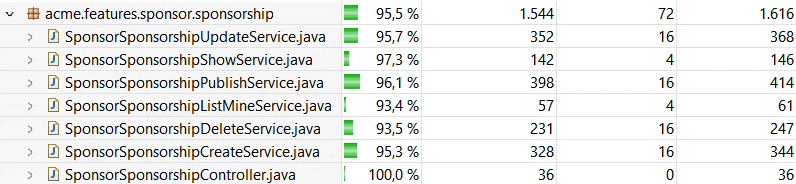
# Historial de versiones

|  |  |  |  |
| --- | --- | --- | --- |
| Fecha | Versión | Descripción | Entrega |
| 27/05/2024 | V1.0 | Inicio del documento | D04 |
| 08/07/2024 | V2.0 | Revisión y cambios para la segunda Convocatoria | D04 2nd Call |
| 22/10/2024 | V3.0 | Revisión y cambios para la tercera Convocatoria | D04 3rd Call |

# Capítulo 1 – Pruebas funcionales

## Sponsorship

Tras ejecutar todos los test, se puede observar que para sponsorship se cubre el 95,5%, valor que está por encima de la recomendación mínima del 90% que debería cubrir al menos todos los test. Y este porcentaje es así de alto porque en el delete el unbind, el método se va a ejecutar al intentar eliminarse cuando hay entidades hijas, invoices, publicadas. Esto ha cambiado con respecto a la entrega 2, como lo pedían en los requisitos de la 2nd Call, corrigiendo el fallo que tenía entonces.



En primer lugar, para no repetirlo durante todo el documento, se va a comentar que las líneas que los “assert” siempre aparecen en amarillo. Esto se debe a que, como nos comentan en las clases teóricas, es recomendable añadirlos al inicio de los proyectos software con el fin de asegurarnos que estamos trabajando con elementos no nulos.



Ahora, voy a empezar hablando por el UpdateService.

Para el update service podemos observar que todo está en verde. Esto lo conseguimos en esta clase y en el resto de ellas, porque las pruebas de hacking nos ayudan a pasar por esas condiciones (del status) por las que normalmente un usuario normal no pasaría.

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

Texto

Descripción generada automáticamente con confianza media

Texto

Descripción generada automáticamente con confianza mediaInterfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

A continuación, vamos como el ShowService, el cual está también todo en verde. Para evitar repetirme por todo el documento, concreto ahora que solo se resaltará si ha habido algún caso en que no hemos conseguido todo verde.

Texto

Descripción generada automáticamente con confianza media

Continuamos con el publishService.

Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

Descripción generada automáticamente

Imagen que contiene Texto

Descripción generada automáticamenteInterfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

Descripción generada automáticamente

Seguimos con el ListService.

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

También tenemos el deleteService, que está todo en verde. Incluido el bind por lo comentado anteriormente, al corregirse el error de la segunda entrega.

Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

Descripción generada automáticamente



Por último tenemos el createService. Aquí mencionar que se ha tomado una decisión de diseño en el authorise, que es para evitar un hacking de que actualice otro elemento por el id y la versión, saltándose todas las validaciones que se encuentran en el update y publish, no se permite el acceso si el elemento creado tiene un id distinto a 0 (en caso de que se esté haciendo post. Si no, no tendrá dicho campo id).

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

Texto

Descripción generada automáticamenteInterfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

Descripción generada automáticamente

Como conclusión se puede sacar que todo el “Sponsorship” ha sido probado de manera muy exhaustiva, probando todas las validaciones posibles. Para concluir, decir que en la tabla Excel añadida entre los reportes, llamada DatosPruebaTesting.xlsx, se encuentran todos los datos de pruebas, tanto de casos positivos y negativos en la hoja Sponsorships, como de hacking en la hoja HACK-Sponsorships. Se añaden ahí todos los ejemplos y no en este documento porque se ilustran mejor los datos probados con una descripción en este formato, lo cuál debe hacer más sencillo visualizar todos los casos de prueba. A mí, en particular, me sirvió para darme cuenta de si me faltaba algún caso alguna vez, y hacía más mecánico el repetir los tests si se me había olvidado hacer algún detalle en alguno de ellos.

## Invoice

En esta ocasión 90% por los pelos, con el porcentaje disminuido de nuevo porque en el método delete no es posible acceder al unbind, como antes.

Imagen que contiene Tabla

Descripción generada automáticamente

Como en el apartado anterior, vamos a comenzar con el updateService.

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

Descripción generada automáticamente

Pasamos con el showService.

Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

Descripción generada automáticamente

Seguimos con el publishService.

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

Imagen que contiene Texto

Descripción generada automáticamente

Imagen que contiene Texto

Descripción generada automáticamente

Luego está el listService.

Texto

Descripción generada automáticamente

También tenemos el deleteService, con su unbind en rojo por lo explicado anteriormente.

Texto

Descripción generada automáticamente con confianza media

Y, para terminar, el createService.

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamenteInterfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

Descripción generada automáticamente

Como conclusión se puede sacar que todo el “Invoice” ha sido probado de manera muy exhaustiva, probando todas las validaciones posibles. Al final de este capítulo comento de nuevo la tabla de Excel, con la batería de datos que se ha usado para probar todo el proyecto. También tiene datos para Invoice y sus hacking. Se ha de comentar que no solo se ha hecho una petición, si no que todas las operaciones se han repetido muchas veces, en algunas ocasiones se ha probado entre 30-40 veces como pueden ser los casos del publish, update o create.

# Capítulo 2 – Pruebas de desempeño

Antes que nada, mencionar que con respecto a la segunda entrega, no ha habido cambios significativos. Se han realizado las pruebas de desempeño, pero se ha notado que no ha habido ninguna distinción con las pruebas anteriores, con lo que se ha dejado las anteriores porque son idénticas. Y esto es normal porque solo se ha añadido una validación en la eliminación de la sponsorship, en caso de tener invoices publicadas que no elimine nada.

El desarrollo del software se ha ejecutado durante todo el cuatrimestre en el PC1. 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 resultaros claros.

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

Gráfico, Gráfico de barras

Descripción generada automáticamente

Como se puede observar, ningún promedio supera los 40 ms, lo que es un resultado muy bueno, porque nos indica que las búsquedas se realizan de manera rápida. Comentar que esta foto es después de refactorizar y corregir el código, dejando todo en las querys. Esto ha ayudado a que el sistema haya podido optimizarse y no acumule retrasos en entornos más grandes.

A continuación, vamos a observar el intervalo de confianza para **el PC 1** tanto antes de introducir índices, como después de meter índices.

Interfaz de usuario gráfica, Aplicación, Tabla, Excel

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 0.03897…, por lo que podemos decir que los cambios **sí** dieron como resultado cambios importantes (no demasiado porque están algo próximos a Alpha, pero no está junto a éste); los tiempos de muestreo son diferentes, siendo la media de después mejor. Por lo que concluimos que mejora el sistema con índices.

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

Tabla

Descripción generada automáticamente

Con estos resultados se ha hecho de nuevo un **Z-Test (PC 2)** para analizar correctamente los datos, y he obtenido:

Aplicación, Tabla

Descripción generada automáticamente

Podemos observar que Alpha es 0.05, y que el p-value es 0.05168051… 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**, solo el PC1 muestra una diferencia significativa (ligeramente) 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 solo el primero 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.