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 María Portela Huerta

GRUPO C1.029

27/05/2024

Versión 1.0

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)

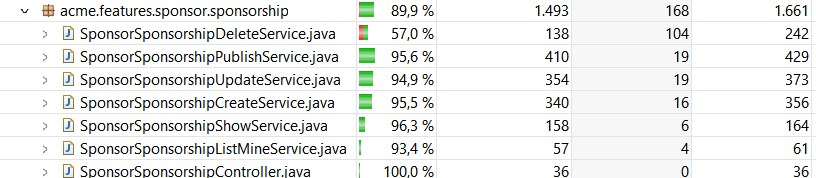
# Historial de versiones

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

# Capítulo 1 – Pruebas funcionales

## Sponsorship

Tras ejecutar todos los test, se puede observar que para sponsorship se cubre el 89,9%, valor que está muy próximo de la recomendación mínima del 90% que debería cubrir al menos todos los test. Pero, no lo hace por el delete, cuyo unbind baja.



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.

Texto

Descripción generada automáticamente

Voy a empezar hablando por el UpdateService.

Para el update service podemos observar que todo está en verde, menos alguna validación, que se ha probado y funciona igual en otros casos. Supongo que se me habrá pasado probarlo aquí, porque en los otros (como el create) funciona.

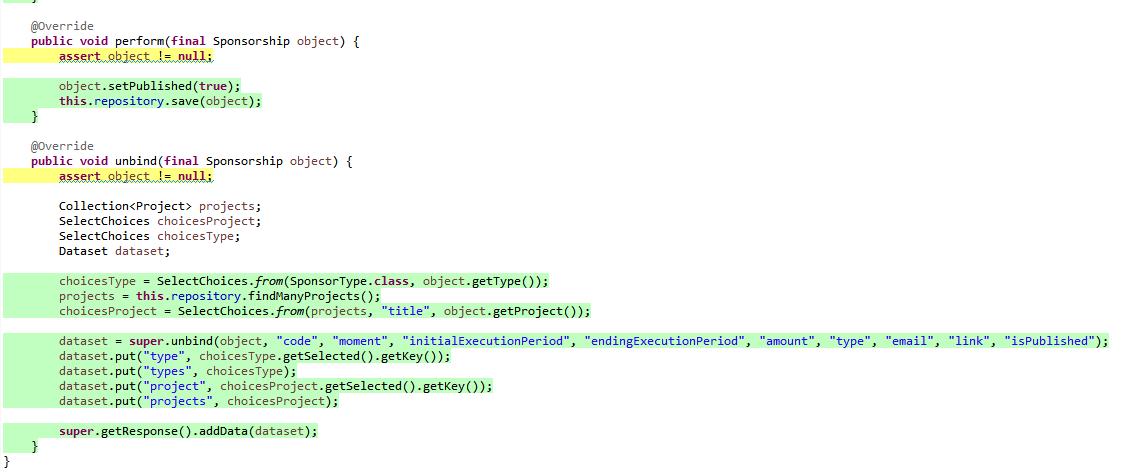


A continuación, vamos como el ShowService, el cual está todo en verde, menos el showInvoice que está implementado así para los casos de hackeo.



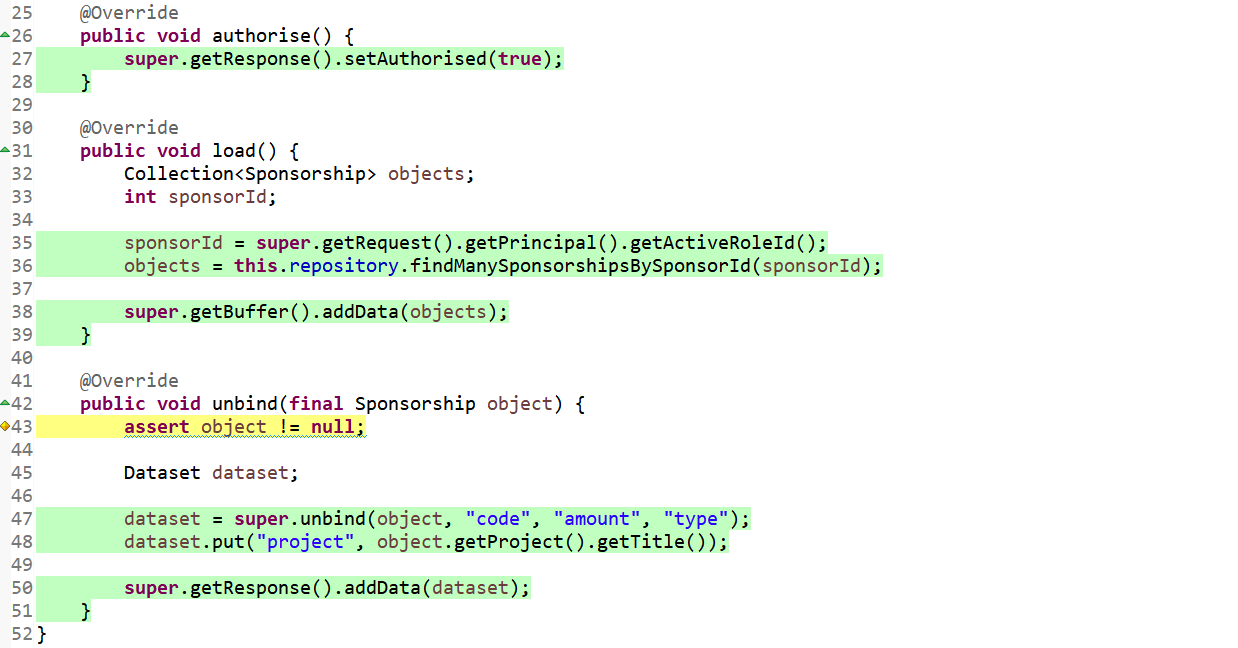
Continuamos con el publishService.





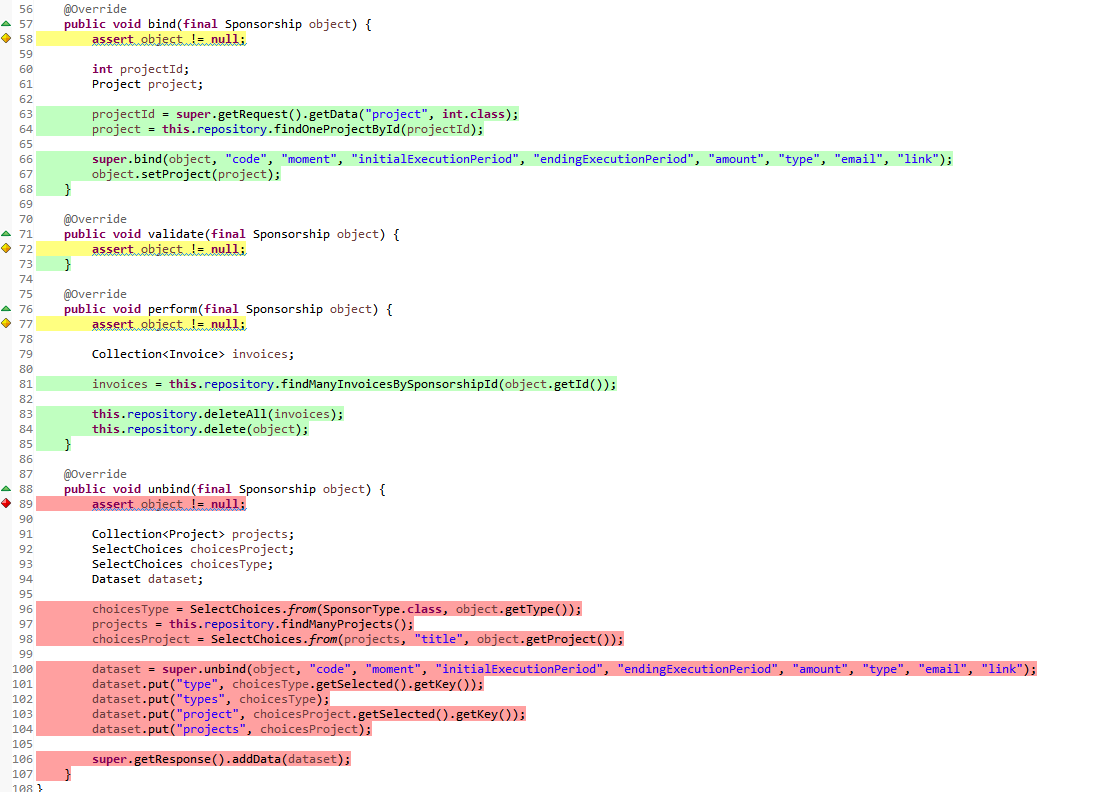
Como se puede comprobar todo en verde.

Seguimos con el ListService.



Está todo en verde.

También tenemos el deleteService, que está todo en verde, menos el unbind porque siempre que un usuario quiera eliminar un objeto, que pueda eliminar, lo va a eliminar.



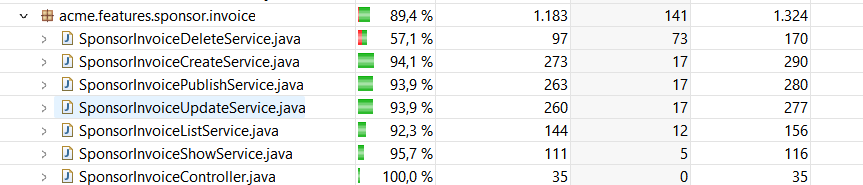
Por último tenemos el createService



Como conclusión se puede sacar que todo el “Sponsorship” ha sido probado de manera muy exhaustiva, probando todas las validaciones posibles. Al final de este capítulo se muestra una imagen con la batería de datos que se ha usado para probar todo el proyecto. 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 créate.

## Invoice

En esta ocasión, casi superamos el umbral del 90% sin embargo, se el porcentaje ha disminuido debido a que en el método delete no es posible acceder al unbind.



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

Texto

Descripción generada automáticamente



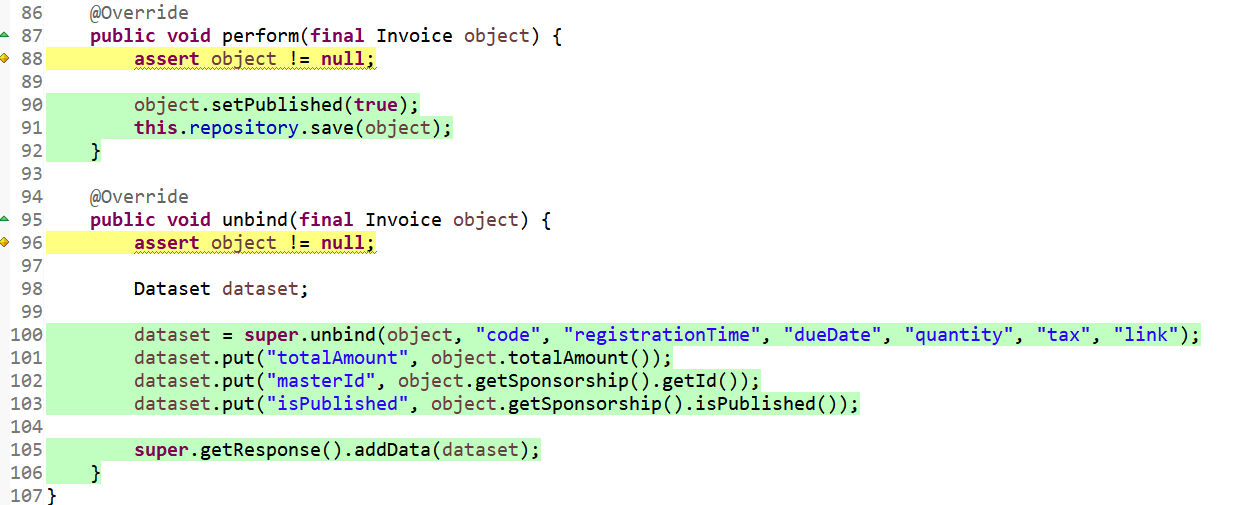
Podemos observar que está todo en verde y comprobado perfectamente.

Pasamos con el showService que está todo en verde también.



Seguimos con el publishService, que es exactamente igual que el updateService y que también es igual que el createService, el cual tiene todo en verde..

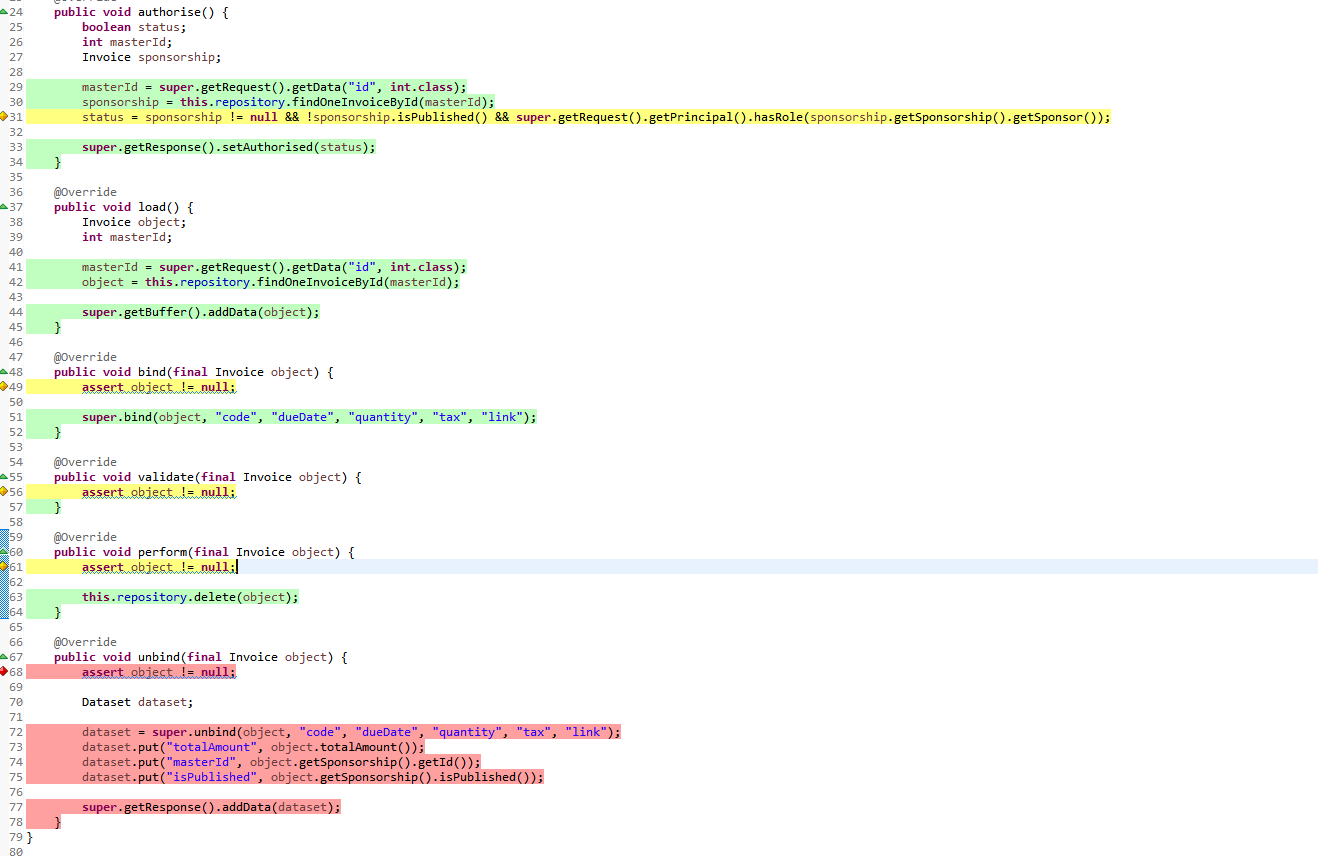




Texto

Descripción generada automáticamente

Por último, tenemos el deleteService, que como se puede observar, nos baja el porcentaje porque nunca llega al unbind.



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 se muestra una imagen con la batería de datos que se ha usado para probar todo el proyecto. 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.

Para terminar este capítulo me gustaría mostrar todos los datos que se han usado para las pruebas. He usado el Excel que se ha añadido en la actualización del framework para tomar de ahí los datos.

# Capítulo 2 – Pruebas de desempeño

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

Descripción generada automáticamente con confianza media

Como se puede observar, ningún promedio supera los 50 ms, 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 la primera vez que realicé el test, observé que había un par de métodos que llegaban a los 50 ms, así que gracias a eso pude darme cuenta de que el código que había implementado no estaba refactorizado correctamente, porque me traía colecciones de datos enteras, y luego recorría todos esos datos para obtener los resultados, lo que en una base de datos mucho más grande provocaría mucha perdida de tiempo, así que opté por solucionar el código y poner todo lo necesario directamente en las querys.

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

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 0.698… 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

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

Tabla

Descripción generada automáticamente

Podemos observar que Alpha es 0.05, y que el p-value es 0.751… 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.