DP2 2024

Acme Software Factory

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

Miembro:

* David Godoy Fernández (davgodfer@alum.us.es)

Tutor: José González Enríquez

GRUPO C1.029

27/05/2024

Versión 1.0

Índice

[Historial de versiones 3](#_Toc167536083)

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

[Contract 4](#_Toc167536085)

[Progress Log 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

## Contract

Tras ejecutar todos los test, se puede observar que para contract se cubre el 92,9%, 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.

Texto

Descripción generada automáticamente

Vamos a empezar hablando por el UpdateService.

Para el update service podemos observar que todo está en verde, menos una validación.

Texto

Descripción generada automáticamente

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

Por ejemplo, aquí tenemos el “project” que siempre tiene que ser publicado. Sin embargo, tal y como he desarrollado el formulario, el desplegable solo permite elegir entre proyectos publicados, por lo que manualmente no se pueden hacer pruebas. Sí hay que tener en cuenta esta validación por si se intenta hacer algún hack. El resto de las validaciones están en verdes.

A continuación, vamos como el ShowService, el cual está todo en verde.

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

Continuamos con el publishService.

Texto

Descripción generada automáticamente

Texto

Descripción generada automáticamente

Interfaz de usuario gráfica, Texto

Descripción generada automáticamente

Como se puede comprobar todo en verde, exceptuando la validación comentada anteriormente para evitar el hack, ya que por ejemplo en este caso, la entidad está configurada para que te muestre un desplegable con solo los proyectos publicados, por lo que solo con postman se podría colocar una ID de un proyecto publicado para que se cumpla la validación.

Seguimos con el ListService, el cual está todo en verde.

Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

Descripción generada automáticamente

Para el deleteService, todo en verde, además cabe destacar que la función del unbind se ha eliminado ya que esta no es necesaria, esto se observaba claramente en el coverage ya que salía de color rojo, por lo que nunca se ejecutaban.

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

Por último, tenemos el createService.

Interfaz de usuario gráfica, Texto

Descripción generada automáticamente

Texto

Descripción generada automáticamente

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

Como podemos comprobar le ocurre lo mismo que al publish y al update, que aparece la línea en amarillo ya que no se puede comprobar sin realizar postman.

Como conclusión se puede sacar que todo el “Contract” 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 muchas veces como pueden ser en los casos del publish, update o create.

## Progress Log

Tras ejecutar todos los test, se puede observar que para progress log se cubre el 94,6%, valor que está por encima de la recomendación mínima del 90% que debería cubrir al menos todos los test.

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

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

Texto

Descripción generada automáticamente

Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

Descripción generada automáticamente

Podemos observar que está todo en verde a excepción de la línea del if para “publishedContract”, esto se debe a que no hay un campo en el formulario con ese nombre es simplemente indicativo, sin embargo, observamos que la validación se cumple correctamente.

Pasamos con el showService que está todo en verde también.

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

Seguimos con el publishService, el cual tiene todo en verde a excepción de los campos “recordId”, “registrationMoment”, y “publishedContract”. En el caso de publishedContract se comentó anteriormente la razón sin embargo en los otros 2 casos no se entiende la razón de que la línea esté en amarillo ya que la validación está siendo complicado correctamente como indica el converage, y las 2 condiciones de las líneas en amarillo se cumplen por lo que estas no deberían aparecer en amarillo.

Imagen que contiene Texto

Descripción generada automáticamente

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

Descripción generada automáticamente

Seguimos con el createService, se observa la línea amarilla comentada anteriormente debido a la inexistencia de un campo publishedContract, sin embargo aparece otra validación en amarillo, esto se debe a que como estamos realizando un create no se puede comprobar que la id del objeto sea igual a una existente ya que este no tiene una id asignada todavia.

Texto

Descripción generada automáticamente

Interfaz de usuario gráfica, Texto, Aplicación, Chat o mensaje de texto

Descripción generada automáticamente

Para el listService, se observa todo el código en verde.

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

Por último, tenemos el deleteService, que como se puede observar, en esta entidad el unbind también ha sido eliminado ya que era código incensario que aparecía en rojo en el coverage.

Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

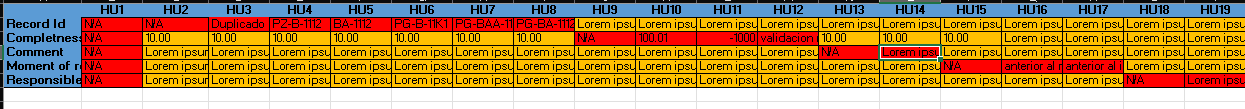
Descripción generada automáticamente

Como conclusión se puede sacar que todo el “Progress log” 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 muchas veces como pueden ser en los casos del publish, update o create.

Para terminar este capítulo se muestra a continuación 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, y lo he pegado en un bloc de notas para tenerlo a mano a la hora de realizar las pruebas.

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente



Interfaz de usuario gráfica, Texto, Aplicación, Tabla, Excel

Descripción generada automáticamente

# 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

Como se puede observar, existen 3 picos, esto se debe a que hemos implementado una api para gestionar las currency obtenido así el valor de la moneda a tiempo real, esto hace que el tiempo aumente mucho en los servicios que se ha utilizado, también se ha decidido implementar los índices solo en la primera entidad debido a que la segunda entidad tenía tiempos mucho más inferiores debido a que no utilizaba la api y apenas haría diferencia. Por lo demás se observa que las búsquedas no han tardado más de 50 ms lo que nos indica que las búsquedas se realizan de manera rápida siempre y cuando no dependan de la api.

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

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 0.733 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 haciendo una simulación** (Esto se ha realizado ya que no se disponía de otro Pc para realizar las pruebas por lo que se ha utilizado la simulación definida en el anexo) y he obtenido los siguientes resultados:

Interfaz de usuario gráfica, Aplicación, 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 0.644… 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**, la simulación no 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 la simulación no es concluyentemente más rápido o lento que el real según los datos simulados.

# Bibliografía

Diapositivas de Diseño y Pruebas 2 – Universidad de Sevilla.s