ACME-L3-D04

S5 Test Report

| Grupo C1.02.08 |

| GitHub: https://github.com/mariaescalante/Acme-L3-D04-24.1.0 |

José Luis Cobo Ariza (joscobari@alum.us.es)

| 25/05/2023 |

Versión	Descripción	Fecha
1.0	Versión inicial	25/05/2023
1.1	Versión final	26/05/2023

Índice

Introducción:

En este documento se detalla la información relativa al informe de test individual del integrante José Luis Cobo Ariza. Se explicarán las pruebas unitarias y de rendimiento del proyecto.

Pruebas unitarias :

Aquí podemos comprobar las pruebas unitarias del código. Por una parte, encontramos en Audit los siguientes test:

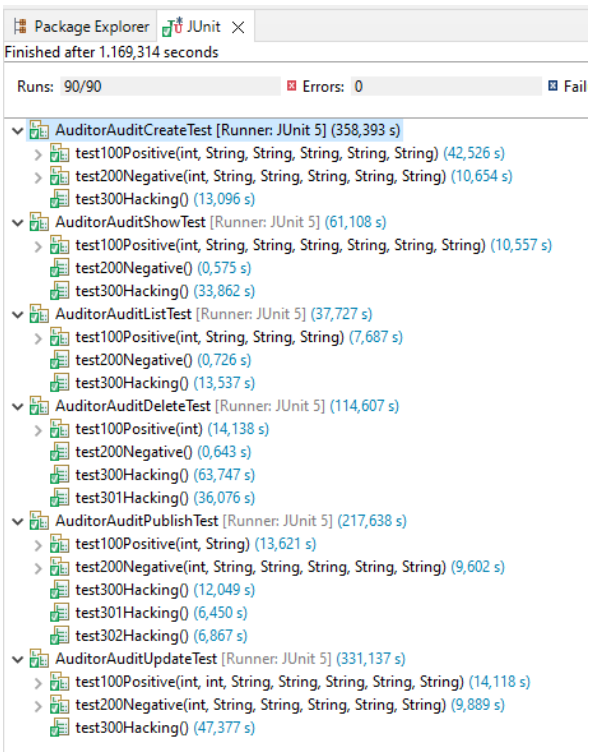
- Create: Comprobamos que a la hora de crear se cumplan las características implementadas del Audit.
- Show: Comprobamos que al mostrar los Audits, se hace correctamente.
- List: Comprobamos que al listar los Audits, se hace correctamente.
- Delete: Comprobamos que al borrar un Audit, se hace correctamente.
- Publish: Comprobamos que al publicar un Audit, se hace correctamente.
- Update: Comprobamos que a la hora de actualizar se cumplan las características implementadas del Audit.

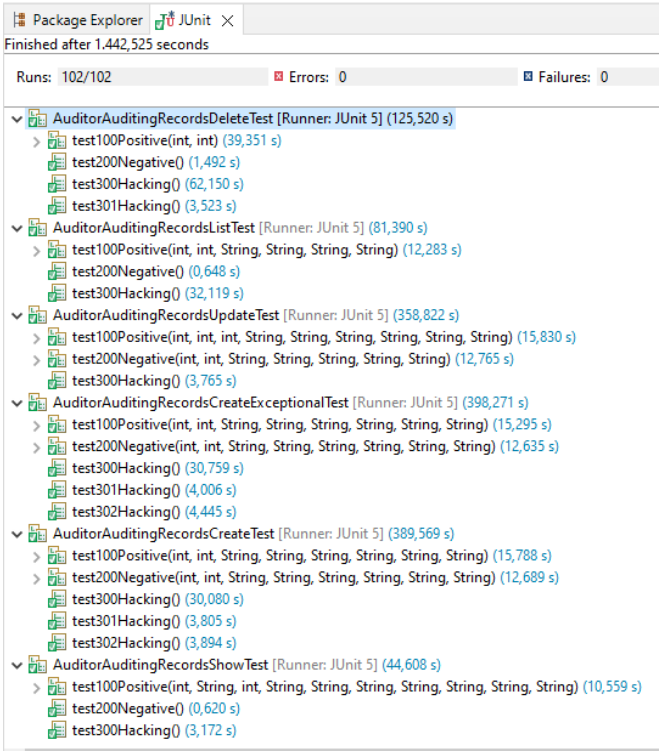
Para los Auditing Records, he comprobado los mismos test con las mismas funcionalidades y además, he añadido el siguiente test:

- Create excepcional: Comprobamos que a la hora de crear se cumplan las características implementadas del Auditing Records.

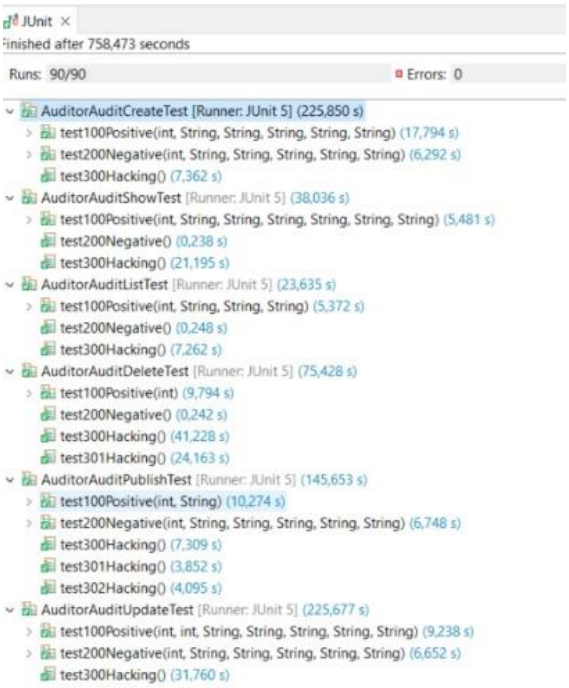
Pruebas de rendimiento:

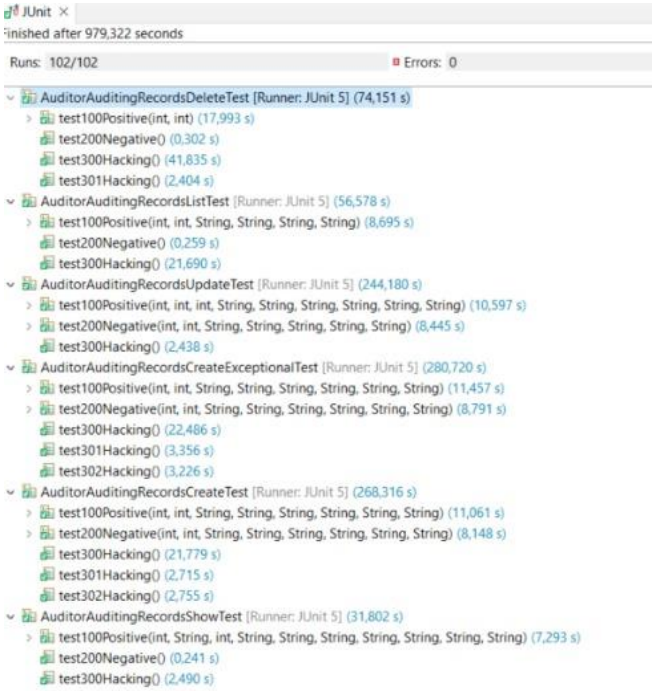
Aquí podemos comprobar las pruebas unitarias en dos ordenadores diferentes y comprobar la eficiencia en cada uno de ellos. Primero, vemos el testing del ordenador numero 1 el cual es más lento que el del ordenador 2:





Ahora, vemos el testing del ordenador 2 con el que los test se evaluan más rapidos:





Por tanto, se puede ver que en los dos ordenadores funcionan a la perfección los test.

Conclusión:

En conclusión, los resultados de las pruebas han sido exitosos tanto en un solo ordenador como en varios equipos para nuestro proyecto de programación. El capítulo de pruebas funcionales proporciona una lista organizada de casos de prueba, lo que facilita la identificación de errores y su eficacia en la detección de problemas. Por otro lado, el capítulo de pruebas de rendimiento incluye gráficos y un intervalo de confianza del 95% para el tiempo de respuesta del proyecto en diferentes computadoras, así como una comparación de hipótesis de confianza del 95% sobre la computadora más potente.

Además, hemos seguido un enfoque sistemático utilizando herramientas de planificación y Eclipse para producir los resultados de manera eficiente. Esto ha evitado que el proceso se convierta en una pesadilla y ha garantizado la coherencia y precisión de los informes de prueba.

Incluso en escenarios excepcionales donde un estudiante trabaja individualmente, se ha encontrado una solución para recopilar datos sobre una segunda computadora utilizando los laboratorios de la escuela. En caso de que esto no sea posible, se ha aplicado una simulación precisa utilizando un desplazamiento aleatorio del tiempo promedio de respuesta obtenido en la propia computadora.

En resumen, nuestros resultados de prueba demuestran que el proyecto de programación ha sido exitoso en diferentes entornos computacionales. Esto proporciona una base sólida para concluir que nuestro sistema es robusto y capaz de funcionar de manera efectiva en diversos contextos.