INFORME DE PRUEBAS

GRUPO C1.02.03

Autores:

- **Pablo Mera Gómez** – pabmergom@alum.us.es

Repositorio:https://github.com/Alejandrocg024/Acme-L3-D04.git

Fecha del documento: 25/05/2023

Tabla de contenidos.

Resumen general.

A lo largo de este documento se han registrado todas las pruebas que hemos realizado sobre nuestra aplicación, buscando de alguna manera cubrir toda la funcionalidad implementada durante el desarrollo del proyecto. Hay un total de 45 pruebas, cubriendo tanto casos positivos, negativos, como posibles hackeos.

Considerando el rendimiento con el que se han ejecutado los tests en 2 máquinas distintas, se llega a la conclusión de que nuestra aplicación de media puede servir cualquier petición en un intervalo entre 0,0074119 y 0,00660123 segundos.

Tabla de revisión.

Fecha	Versión	Descripción
19/05/2023	V1	Versión final.

Introducción.

Para asegurarnos que la implementación de todos los requisitos ha sido adecuada, hemos realizado un conjunto de tests que cubren el buen funcionamiento de todas aquellas tareas propuestas durante el proyecto.

En este documento explicaremos todos los distintos tests que hemos realizado y la intención por la que fueron creados. Además, mostraremos el tiempo en ejecutar cada petición y cada uno de los tests. Por último vamos a mostrar un gráfico que permita identificar la capacidad de respuesta del sistema.

Contenido.

Las pruebas realizadas para comprobar el correcto funcionamiento de la aplicación han sido:

Para las auditorías:

Create:

- Test100Positive: Este test autentica a un auditor, lista sus auditorías y crea una nueva, confirmando que se ha creado de manera correcta.
- Test200Negative: Este test intenta crear una auditoría con datos incorrectos.
- Test300Hacking: Este test intenta crear una auditoría usando roles no apropiados.

Delete:

- Test100Positive: Nos logueamos como auditor1 y borramos algunas auditorias de auditor1 no publicadas.
- Test200Negative: No hay caso negativo.
- Test300Hacking: Intentamos borrar las auditorías del auditor2, siendo auditor1, administrator o lecturer1.
- Test301Hacking:Nos logueamos como auditor1 e intentamos borrar sus auditorías publicadas, cosa el sistema no debe permitir.

List:

- Test100Positive: Nos logueamos como auditor1 y listamos todas sus auditorías.
- Test200Negative: No hay caso negativo.
- Test300Hacking: Intentamos listar auditorías sin ser auditor

Publish:

- Test100Positive: Nos traemos de la base de datos una auditoría que cumpla las condiciones para ser publicada y la publicamos.
- Test200Negative: Intentamos publicar auditorías que no se pueden publicar por las siguientes razones: No tienen registros de auditorías.
- Test201Negative: Intentamos publicar auditorías que no se pueden publicar por las siguientes razones: Todos sus registros de auditorías no están publicados, es decir, alguno de sus registros está sin publicar.
- Test300Hacking: Intentamos publicar una auditoría sin ser los creadores de dicha auditoría.
- Test301Hacking: Intentamos publicar auditorías que no se pueden publicar debido a que ya lo están.

Show:

- Test100Positive: Nos logueamos como auditor1 y mostramos alguna de sus auditorías.
- Test200Negative: No hay caso negativo.
- Test300Hacking: Intentamos ver las auditorías de un auditor que no es el que está logeado.

Update:

- Test100Positive: Este test autentica a un auditor, lista sus auditorías y actualiza una existente, confirmando que se ha actualizado de manera correcta.
- Test200Negative: Este test intenta actualizar una auditoría con datos incorrectos.
- Test300Hacking: Este test intenta actualizar una auditoría usando roles no apropiados.

Para los registros de auditoría:

Create:

- Test100Positive: Este test autentica a un auditor, lista sus auditorías, selecciona una para listar sus registros de auditoría y crea uno nuevo, confirmando que se ha creado de manera correcta.
- Test101Positive: Este test autentica a un auditor, lista sus auditorías, selecciona una ya publicada para listar sus registros de auditoría y crea uno nuevo excepcional, confirmando que se ha creado de manera correcta.
- Test200Negative: Este test intenta crear un registro de auditoría con datos incorrectos.
- Test201Negative: Este test intenta crear un registro de auditoría excepcional sin pulsar la confirmación.
- Test300Hacking: Este test intenta crear una auditoría usando roles no apropiados.
- Test301Hacking: Este test intenta crear un registro de auditoría para una auditoría que no fue creada por el auditor logeado.

Delete:

- Test100Positive: Nos logueamos como auditor1 y borramos su único registro de auditorias no publicado.
- Test200Negative: No hay caso negativo.
- Test300Hacking: Intentamos borrar los registros de auditoría del auditor1, siendo auditor2, administrator o lecturer1.
- Test301Hacking: Nos logueamos como auditor1 e intentamos borrar sus registros de auditorías publicadas, el sistema no debe permitirlo.

List:

- Test100Positive: Nos logueamos como auditor1 y listamos todas sus registros de auditoría.
- Test200Negative: No hay caso negativo.
- Test300Hacking: Intentamos listar registros de auditoría sin ser auditor.

Publish:

- Test100Positive: Buscamos un registro de auditoría sin publicar y los publicamos.
- Test200Negative: No hay casos negativos

- Test300Hacking: Intentamos publicar un registro de auditoría sin ser los creadores de dicho registro.
- Test301Hacking: Intentamos publicar registro de auditoría que no se pueden publicar debido a que ya lo están.

Show:

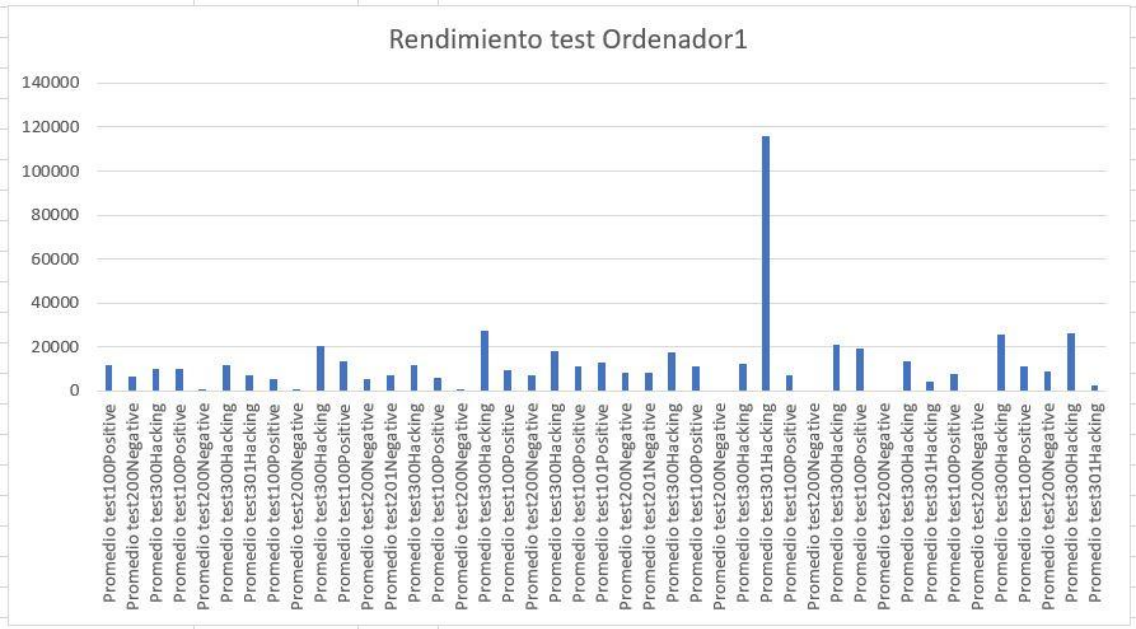
- Test100Positive: Nos logueamos como auditor1 y mostramos alguno de sus registros de auditoría.
- Test200Negative: No hay caso negativo.
- Test300Hacking: Intentamos ver los registros de auditoría de un auditor que no es el que está logeado.

Update:

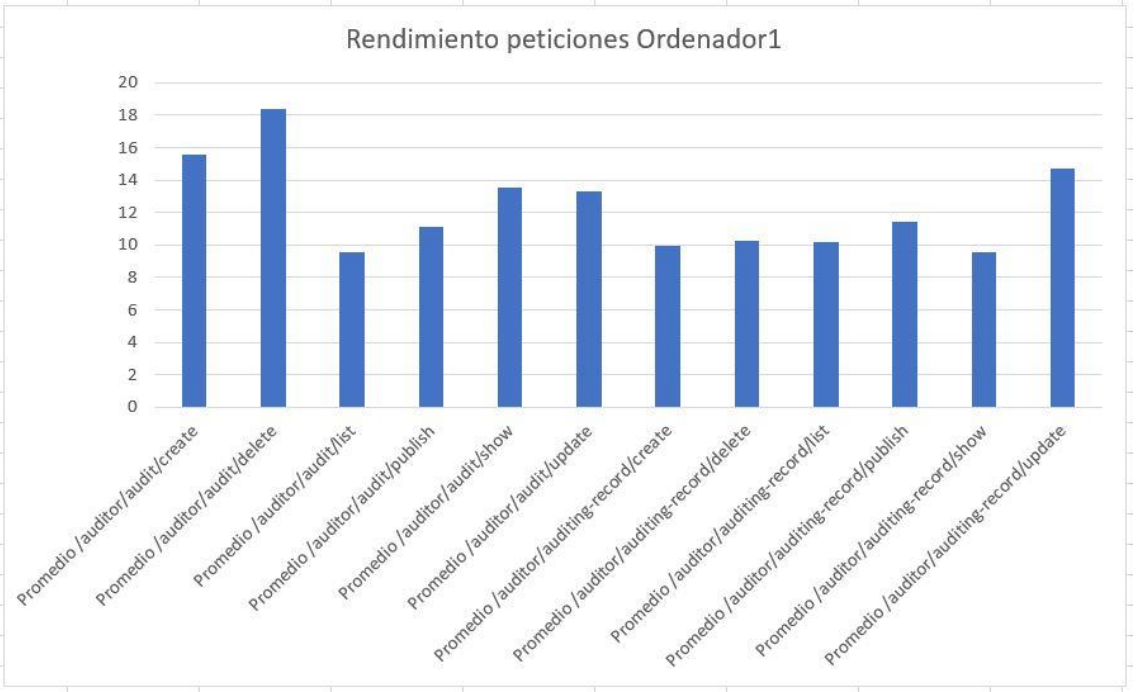
- Test100Positive: Este test autentica a un auditor, lista sus auditorías, escoge un registro y lo actualiza, confirmando que se ha actualizado de manera correcta.
- Test200Negative: Este test intenta actualizar un registro de auditoría con datos incorrectos.
- Test300Hacking: Este test intenta actualizar un registro de auditoría usando roles no apropiados.
- Test301Hacking: Este test intenta actualizar un registro de auditoría que ya ha sido publicado.

Una vez comentadas las distintas pruebas del sistema, es el momento de ver cuál es el rendimiento de las pruebas en sí, para ello analizaremos distintos datos sobre las pruebas y sobre las peticiones realizadas al sistema mientras que estas se ejecutan.

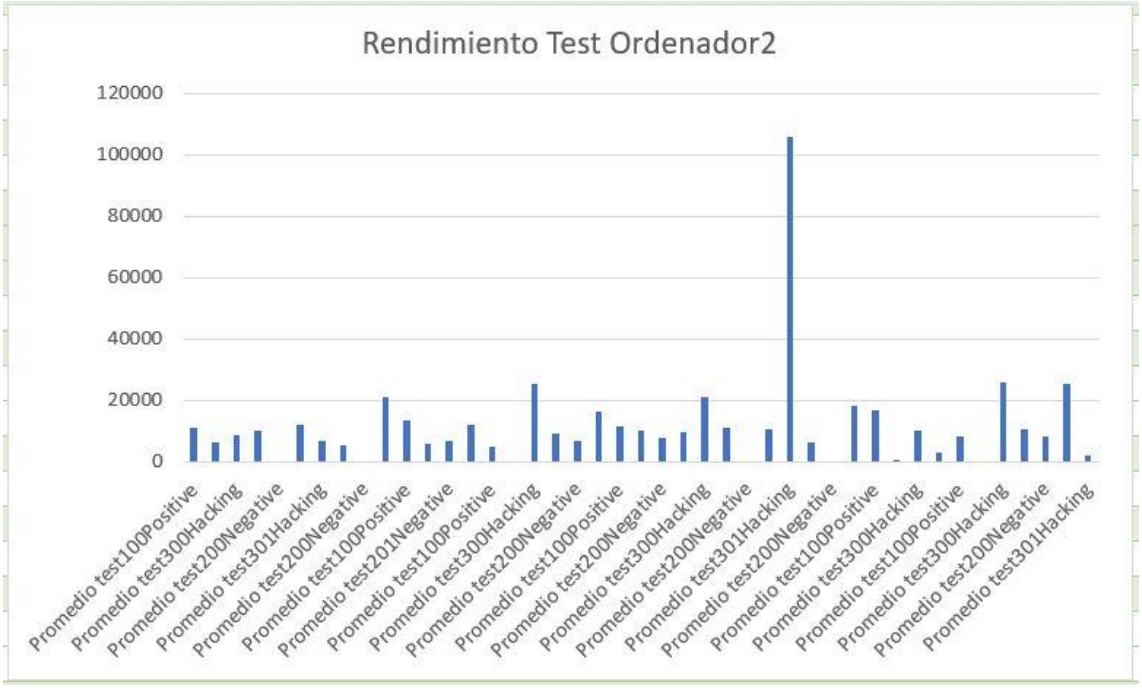
Los resultados en una máquina cuyo hardware es un procesador Intel(R) Core(TM) i5-10210U CPU 2.10 GHz y 8 Gigas de RAM, el tiempo que ha tardado en ejecutarse cada prueba es:



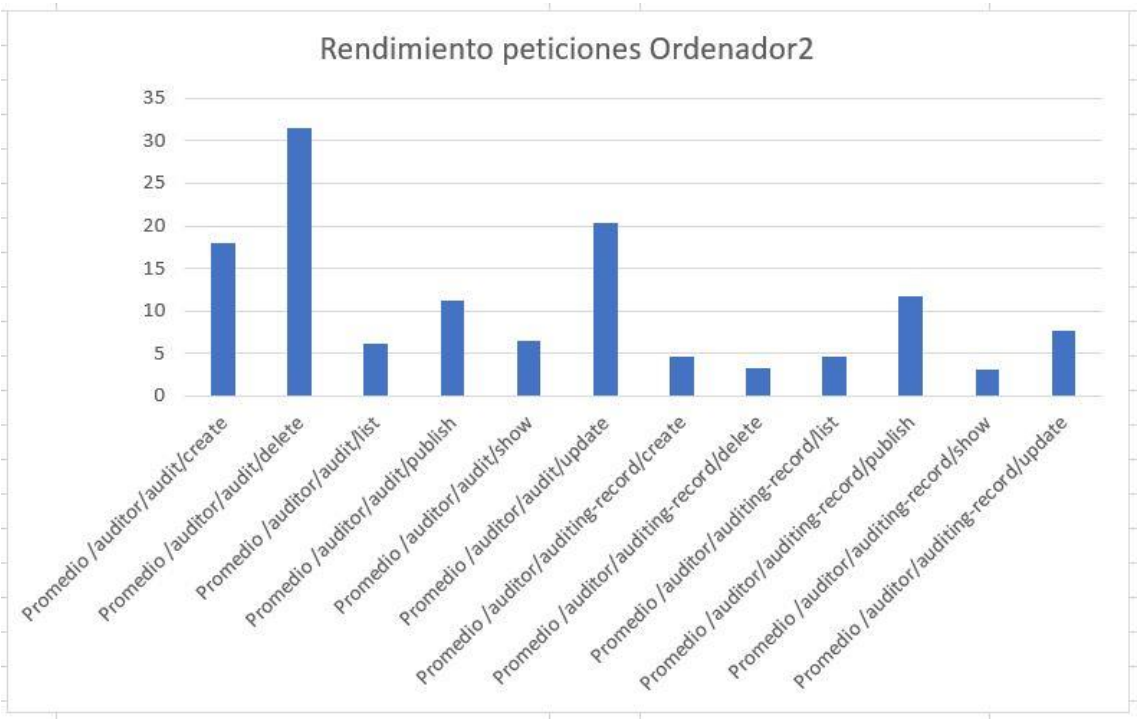
En cuanto a la velocidad con la que se han servido las distintas peticiones en este ordenador es la siguiente:



En otra máquina cuyos componentes principales son, un procesador AMD Ryzen 7 5800H de 3.2GHz y 16 Gigas de RAM, el tiempo que ha tardado en ejecutarse cada prueba es:



En cuanto a la velocidad con la que se han servido las distintas peticiones en este ordenador es la siguiente:



Luego, he comparado los valores de los 2 ordenadores usados para realizar los tests, mostrando un Z-test. En esta gráfica nos fijaremos en el valor crítico de z (dos colas)

Prueba z para medias de dos muestras		
	Before	*After*
Media	11,34571702	6,9949973
Varianza (conocida)	50,7958428	59,0463646
Observaciones	1368	1382
Diferencia hipotética de las medias	0	
z	15,39590603	
P(Z<=z) una cola	0	
Valor crítico de z (una cola)	1,644853627	
Valor crítico de z (dos colas)	0	
Valor crítico de z (dos colas)	1,959963985	

Observándola detenidamente, podemos ver que P(Z<=z) una cola es 0, por lo tanto, las peticiones en el mejor de los casos se sirven en el mismo momento. Además, a través del valor crítico de z (dos colas) podemos denotar que en el peor de los casos el ordenador con menor capacidad tardará 1,96 milisegundos más que el ordenador más rápido. Para saber cuál de las 2 máquina es más veloz a la hora de ejecutar los tests nos fijamos en el valor de z, que al ser positivo nos indica que la máquina adecuada es la segunda.

Una vez comentado esto deberíamos hacer los cálculos, a partir del segundo ordenador, para poder definir un rango de confianza en el cual podamos asegurar a nuestro cliente que de media nuestra aplicación servirá todas las peticiones en un tiempo dentro de ese rango, para calcular este rango hemos utilizado la siguiente tabla de Excel:

Columna1					
			Interval(ms)	7,41190497	6,60123498
Media	7,00656997		Interval(s)	0,0074119	0,00660123
Error típico	0,20662625				
Mediana	5				
Moda	3				
Desviación es	7,68416323				
Varianza de la	59,0463646				
Curtosis	31,0319999				
Coeficiente d	4,55873795				
Rango	99				
Mínimo	0				
Máximo	99				
Suma	9690,08628				
Cuenta	1383				
Nivel de confi	0,405335				

Dicho esto, podemos asegurar a nuestro cliente que de media todas las peticiones realizadas al sistema se servirán en un intervalo de entre 0,0074119 y 0,00660123 segundos.

Conclusiones.

La efectividad final que han tenido las pruebas de nuestra aplicación, gracias a las pruebas hemos podido detectar algún fallo, como por ejemplo en los update del auditing record, en los cuáles no se actualizaban el campo Mark.

Además, como hemos demostrado a lo largo de este documento, podemos garantizar la buena ejecución de todos los tests, en un tiempo estipulado como inferior al límite crítico establecido por parte del equipo de trabajo.

Bibliografía.

Intencionalmente vacío.