

PERFORMANCE REPORT

Grupo E7.02



Álvaro Úbeda Ruiz (alvuberui@alum.us.es)

Mario Pérez Coronel (marpercor8@alum.us.es)

Carlos Garrido Rodríguez (cargarrod12@alum.us.es)

Ramón Rodríguez Bejarano (ramrodbej@alum.us.es)

Mario Rodríguez García (marpercor8@alum.us.es)

Juan Carlos Gómez Rodríguez (juagomram4@alum.us.es)

Repositorio: https://github.com/alvuberui/Acme-Toolkits

Tablero: https://github.com/users/alvuberui/projects/4

23 DE MAYO DE 2022

Contenido

1.	. 1	Resumen ejecutivo	2
		Tabla de revisión	
3.	. 1	Introducción	2
4.	. (Contenido	2
	4.1	L Análisis Computador 1	2
		2 Análisis Computador 2	
		Conclusión	
6.	. [Bibliografía	6

1. Resumen ejecutivo

El presente documento nos dará la posibilidad de tener un informe de rendimiento el cual consiste en realizar dos análisis con un intervalo de confianza de un 95% para el tiempo promedio de pared. Cada análisis debe ser ejecutado en distintas computadoras para comparar ambos análisis, dejando claro que ordenador es más eficiente con este nivel de confianza.

2. Tabla de revisión

Num. Revisión	Fecha	Descripción
1	23/05/2022	Creación del documento.
2	23/05/2022	Finalización del documento para entregar.

3. Introducción

Como bien ya se ha explicado en el <u>apartado uno "Resumen ejecutivo"</u> este documento tiene como objetivo crear un informe sobre el rendimiento para el tiempo promedio de pared tomado por el sistema. Para ello vamos a ejecutar las pruebas creadas por los miembros de este grupo para las tareas que se le han asignado y desarrollado. Una vez ejecutado estas pruebas, se generarán una serie de documentos Excel con los que tendremos que trabajar. A partir de estos documentos Excel vamos a generar unas gráficas que nos dirá el promedio en tiempo que ha tardado en ejecutarse estas pruebas. Al realizar estas pruebas en dos computadoras diferentes y generar documentos Excel diferentes podremos comparar estos resultados de tiempo y poder observar así la diferencia de ambos análisis dependiendo del computador que ha ejecutado las pruebas.

4. Contenido

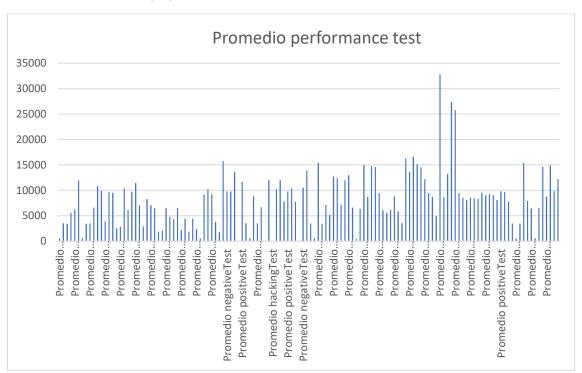
4.1 Análisis Computador

El primer análisis se ha analizado con un lenovo y-700. En este caso cuenta con 12 GB de Ram DDR4, un I7 6700U, y un SSD de 512Gb-

En la siguiente gráfica podemos ver el tiempo promedio de las request que se han realizado a la hora de la ejecución de las pruebas en el computador 1.



Y a continuación el tiempo promedio en los test.



Siguiendo con los tiempos promedios de los request. En la siguiente imagen, podemos apreciar las estadísticas básicas de los datos de rendimiento al finalizar. En este caso, los malos olores aparecido eran simple nombres de variables, por lo que no afectaba al rendimiento de la aplicación, por tanto, no hemos tenido un tiempo de ejecución después de este. Por consiguiente, no hemos creado la tabla de z-Test para comprobar si la refactorización mejoro, o empeoro el funcionamiento.

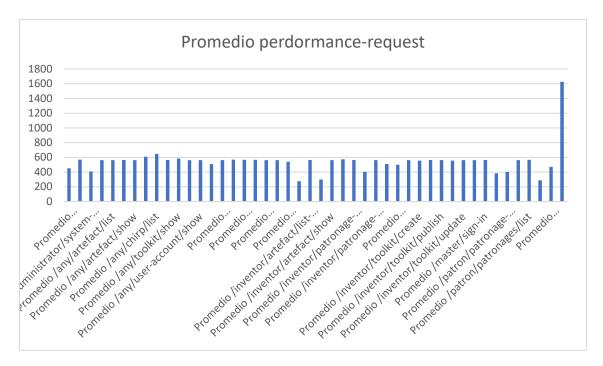
time		
Media	527,7378368	
Error típico	2,528219758	
Mediana	561	
Moda	563	
Desviación estándar	126,081891	
Varianza de la muestra	15896,64323	
Curtosis	528,8529009	
Coeficiente de asimetría	15,49592688	
Rango	4535	
Mínimo	253	
Máximo	4788	
Suma	1312484	
Cuenta	2487	
Nivel de confianza (95,0%)	4,957633386	
Intervalo de nivel de confianza	522,7802034	532,6954701

El promedio de peticiones, su rango de tiempos oscila entre 0,52 segundos a 0,53 segundos con un nivel de confianza del 95%. Haciendo que el rendimiento satisficiera el requerimiento de rendimiento

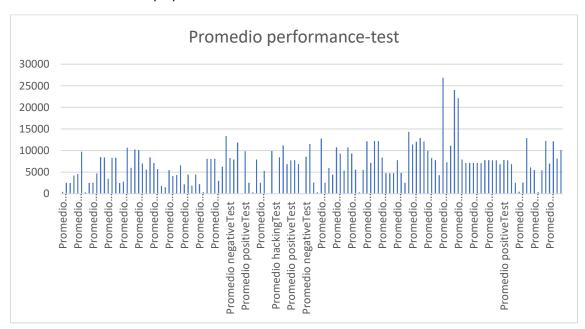
4.2 Análisis Computador 2

El segundo análisis se ha realizado con un asus GL553VD. En este caso cuenta con 8 GB de memoria Ram DDR4, un I7 7700HQ y un HDD 5200RPM.

En la siguiente gráfica podemos ver el tiempo promedio de las request que se han realizado a la hora de la ejecución de las pruebas en el computador 2.



Y a continuación el tiempo promedio en los test.



Siguiendo con los tiempos promedios de los request. En la siguiente imagen, podemos apreciar las estadísticas básicas de los datos de rendimiento al finalizar. En este caso, los malos olores aparecido eran simple nombres de variables, por lo que no afectaba al rendimiento de la aplicación, por tanto, no hemos tenido un tiempo de ejecución después de este. Por consiguiente, no hemos creado la tabla de z-Test para comprobar si la refactorización mejoro, o empeoro el funcionamiento.

time		
Media	456,4354644	
Error típico	3,487833707	
Mediana	560	
Moda	561	
Desviación estándar	173,937676	
Varianza de la muestra	30254,31512	
Curtosis	135,2088897	
Coeficiente de asimetría	5,908461131	
Rango	4387	
Mínimo	222	
Máximo	4609	
Suma	1135155	
Cuenta	2487	
Nivel de confianza(95,0%)	6,839358316	
Intervalo de confianza	449,5961061	463,274823

El promedio de peticiones, su rango de tiempos oscila entre 0,44 segundos a 0,46 segundos con un nivel de confianza del 95%. Haciendo que el rendimientos satisficiera el requerimiento de rendimiento

5. Conclusión

Como conclusión, podemos destacar que dependiendo del computador y del código podemos ver la eficiencia. Esto podría influir a la hora de trabajar en un gran proyecto debido a que podríamos tardar más tiempo en el proyecto debido a la ejecución de pruebas como las mostradas anteriormente. No solo podría ser en casos de ejecutar pruebas, sino a la hora de ejecutar el proyecto. Esto podría ralentizar la producción dando lugar a una pérdida económica al no poder trabajar en otros proyectos.

También que, al crear código sin malos olores desde el primer momento, hemos conseguido mantener o incluso mejorar el rendimiento en el sprint anterior. Haciendo que tengamos menos perdidas de tiempo en arreglar estos y una aplicación con respuestas mas rápidas, facilitando su testeo y uso.

6. Bibliografía

Intencionadamente en blanco.