

DP2 2021-2022

Progress Report

Acme Toolkits

Repositorio:

<https://github.com/RingoML/Acme-Toolkits>

Miembros:

- Caro Albarrán, Francisco Andrés (fracaralb@alum.us.es)
- Gallego Huerta, Alberto(albgalhue@alum.us.es)
- Martín Luque, José Manuel (josmarluq@alum.us.es)
- Reyes Madrid, Francisco (frareymad@alum.us.es)
- Sillero Manchón, Jorge (jorsilman@alum.us.es)

GRUPO E3.06

21 de abril de 2022

Índice

Repositorio:.....	1
Miembros:.....	1
GRUPO E3.06	1
Resumen ejecutivo	2
Tabla de revisiones	2
Introducción.....	3
Recopilación de datos.....	4
Análisis de datos	4
Conclusión.....	9

Resumen ejecutivo

El informe de rendimiento a realizar es un documento que agrupa y analiza las métricas correspondientes a este atributo con respecto a nuestro proyecto Acme-Toolkits. Esto es, dos análisis realizados en distintos ordenadores sobre el intervalo de confianza y un contraste de hipótesis que compara los resultados obtenidos en ambos dispositivos

Tabla de revisiones

FECHA	VERSIÓN	DESCRIPCIÓN
2022-04-25	1	Versión inicial

Introducción

Los requisitos estudiados en este informe son requisitos no funcionales, más concretamente centrados en el tiempo que tarda un ordenador en ejecutar una petición o mostrar una vista.

En nuestro caso, se llevará a cabo el análisis del tiempo real medio usado por cada uno de los dos ordenadores para llevar a cabo las tareas mencionadas anteriormente. Se debe declarar cuál de los dispositivos es más eficiente a un nivel de confianza del 95%.

Primero se explicarán y se realizarán todas las pruebas correspondientes al primer ordenador. Más tarde, seguiremos el mismo procedimiento con el segundo, para por último realizar una hipótesis de contraste de los resultados obtenidos para ambos dispositivos

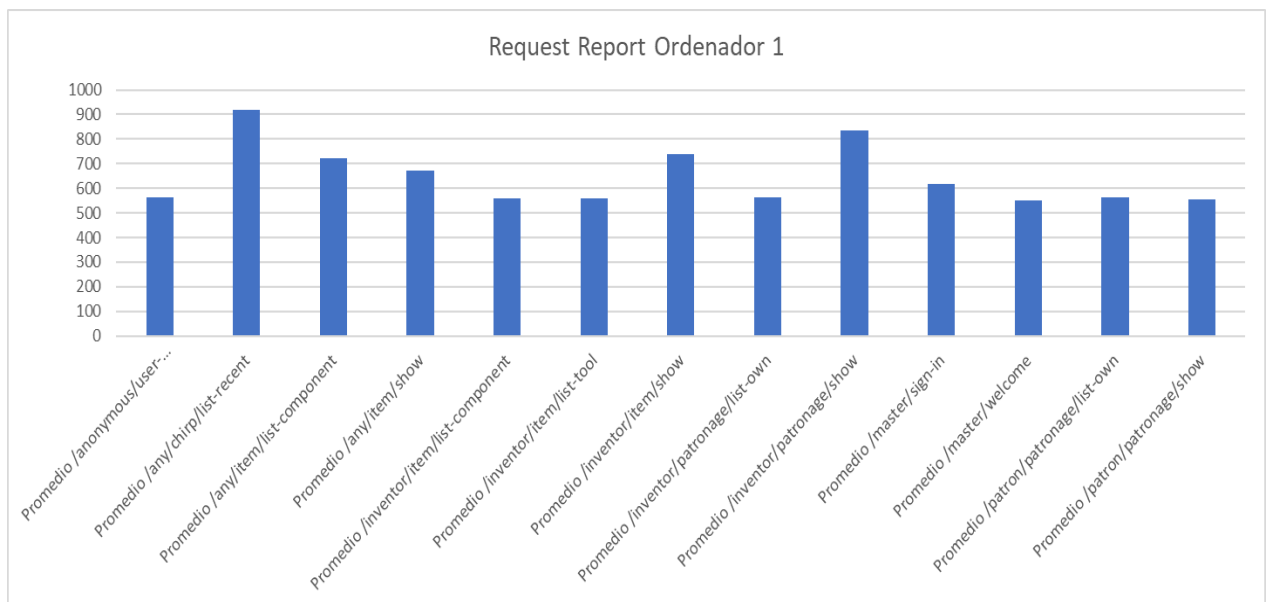
Recopilación de datos

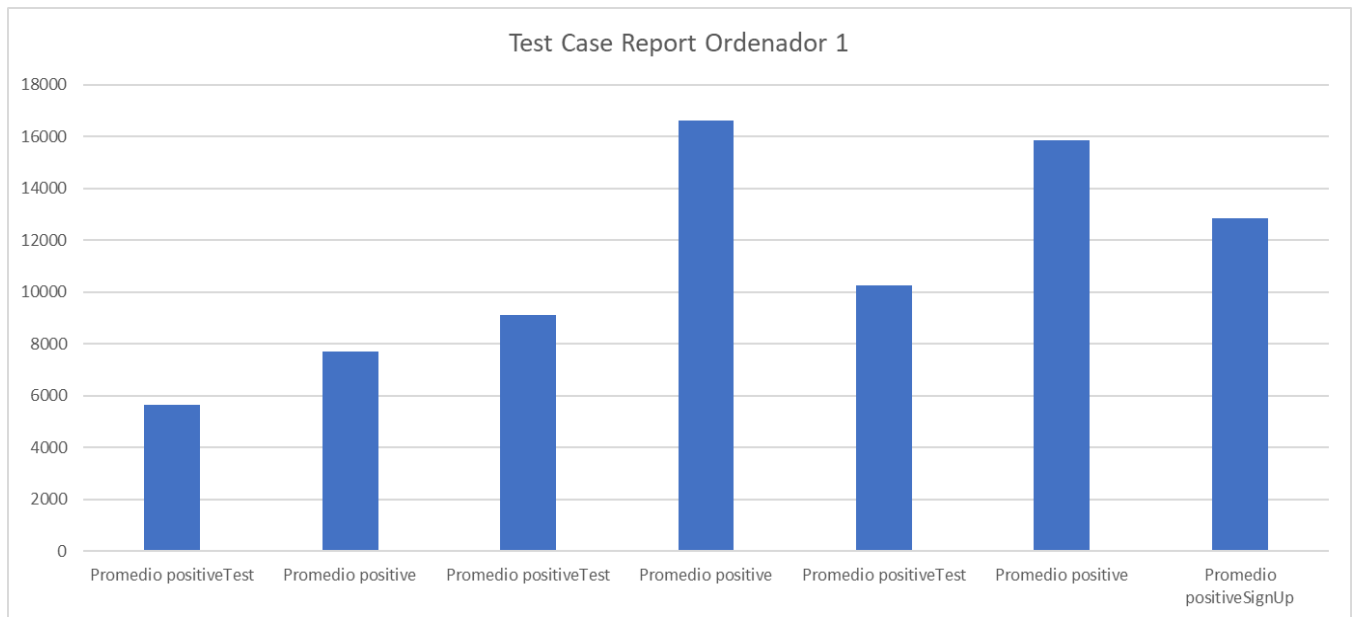
Los datos usados para medir dicho rendimiento gracias a que al ejecutar los tests de Acme-Toolkits, el framework genera automáticamente unos archivos con extensión .csv, los cuáles abrimos con Microsoft Excel. Una vez aquí, podemos aplicarle funciones a los datos para que nos ofrezcan las métricas

Análisis de datos

Ordenador 1

Dicho dispositivo cuenta con un procesador I7-8750H, una RAM de 8 GB y Windows 11 como sistema operativo. Tras la ejecución de los tests, se ha generado la siguiente gráfica de tiempos promedio.



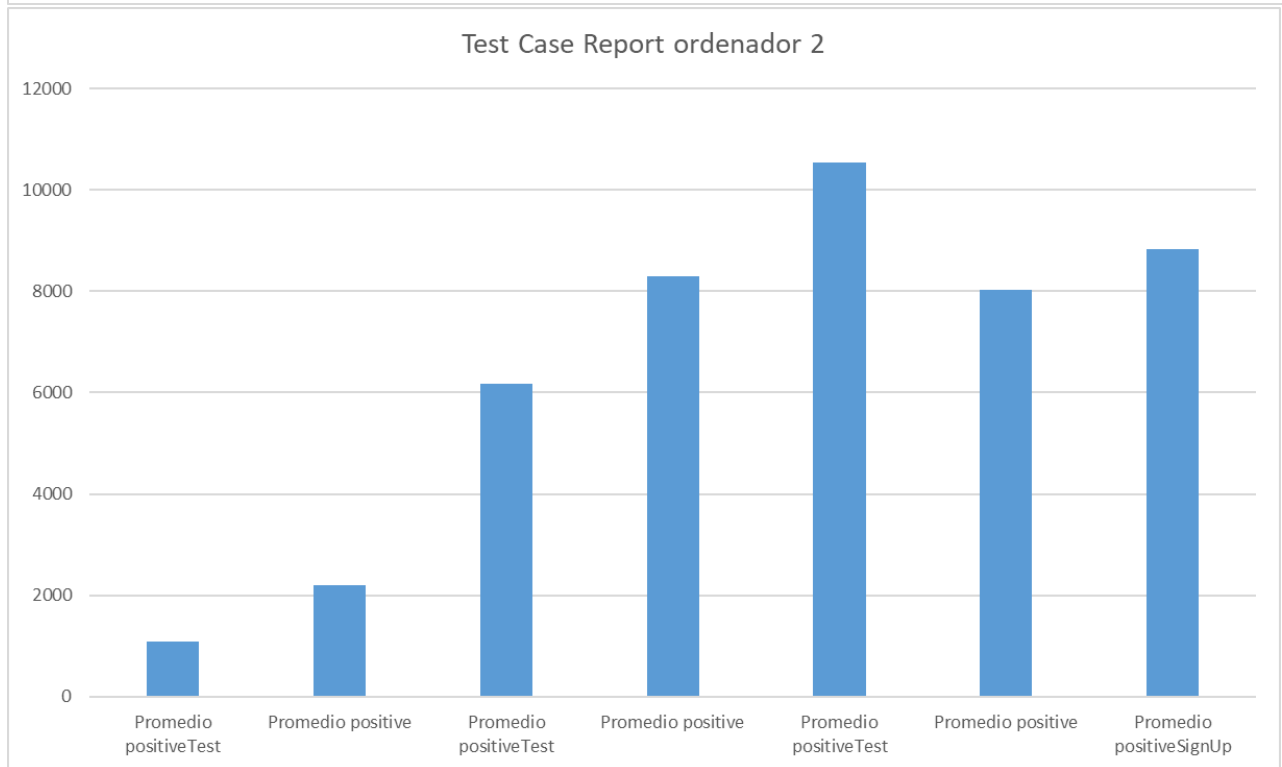
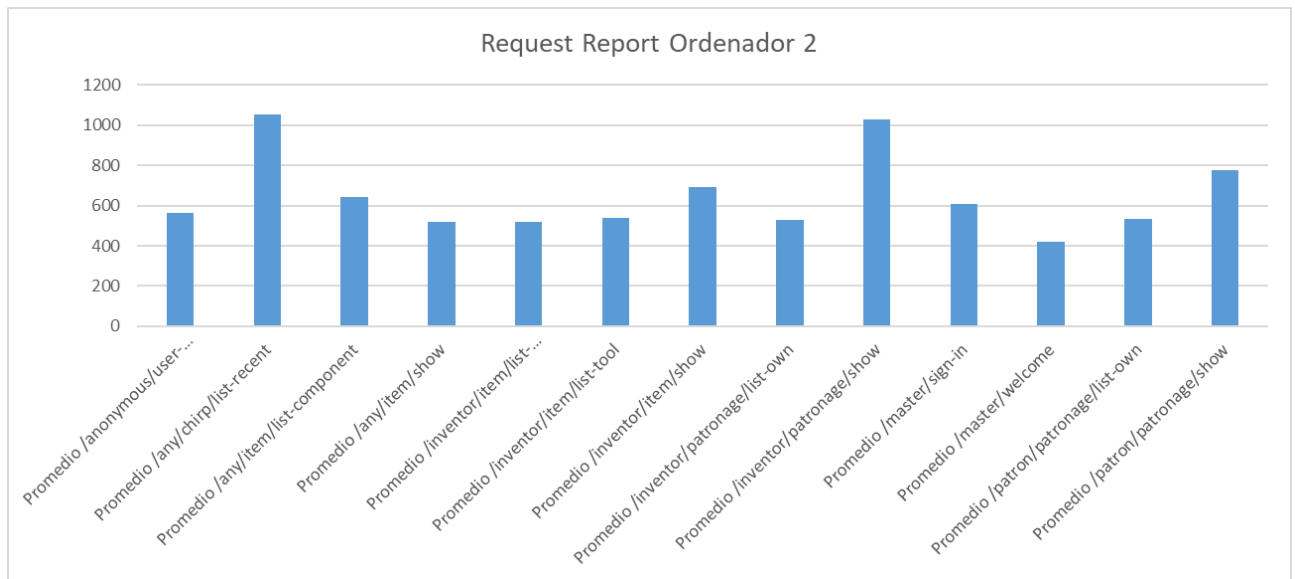


Como podemos observar en dicha gráfica, todos los tests tardan en ejecutarse entre 6 y 16 segundos

<i>time</i>		
Media	582,515228	
Error típico	23,60178042	
Mediana	561	
Moda	562	
Desviación estándar	332,944085	
Varianza de la muestra	110851,7637	
Curtosis	120,3587584	
Coeficiente de asimetría	10,04345933	
Rango	4469	
Mínimo	235	
Máximo	4704	
Suma	115920,5304	
Cuenta	199	
Nivel de confianza(95,0%)	46,54312349	
Nivel de confianza	535,9721045	629,058352

En esta tabla podemos observar que hemos obtenido: el nivel de confianza ha sido del 46.54 milisegundos mientras que el intervalo de confianza oscila entre 535.97 y 629.05. Ambos son menores que los 1000 ms requeridos. Se obtiene por tanto un resultado positivo

En la siguiente imagen se muestra el profiling del proyecto obtenido en el dispositivo 1 tras ejecutar todos los tests



En este segundo dispositivo, observamos que los tests tardan en ejecutarse entre 1 y 11 segundos

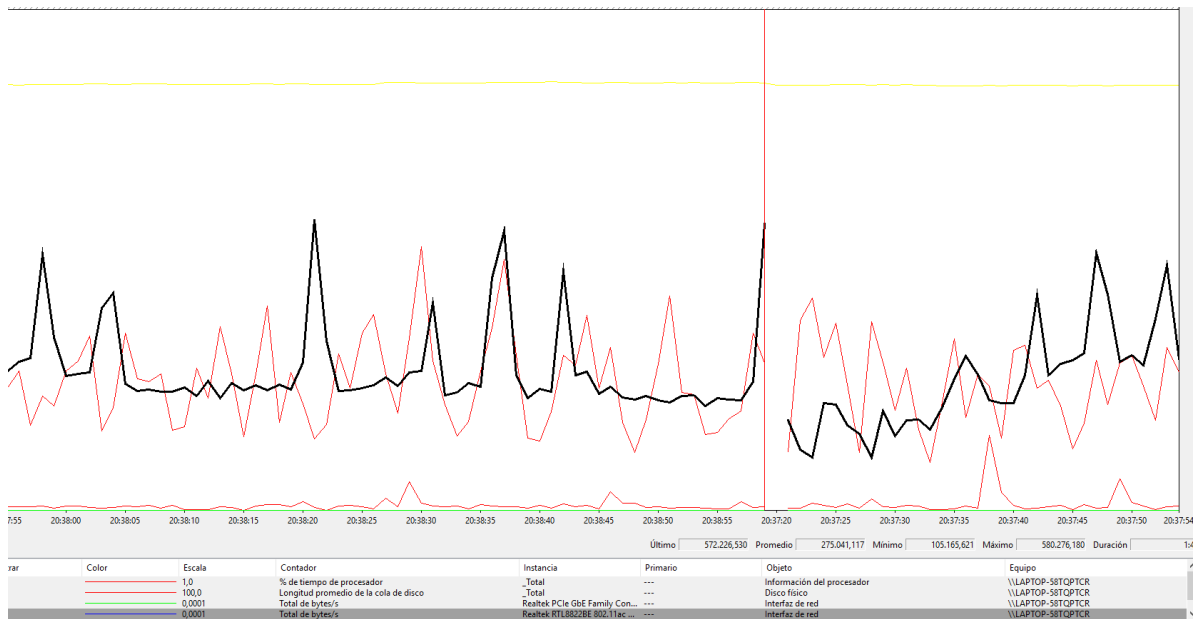
time		
Media	498,004955	
Error típico	34,4229594	
Mediana	520	
Moda	520	
Desviación estándar	485,595599	
Varianza de la muestra	235803,086	
Curtosis	106,579151	
Coeficiente de asimetría	9,24794901	
Rango	6118	
Mínimo	211	
Máximo	6329	
Suma	99102,986	
Cuenta	199	
Nivel de confianza(95,0%)	67,8826775	
Nivel de confianza	430,122277	565,887632

En esta tabla, observamos que el nivel de confianza obtenido es de 67.88 ms mientras que el intervalo de confianza oscila entre 430.12 y 565.88 ambos menores que los 1000 ms requeridos. Se ha obtenido por tanto un resultado positivo

Se ha realizado también el profiling del proyecto en busca de posibles cuellos de botella

CPU samples Thread CPU time			
Results: View: Collected data: Thread Dump			
Name	Total Time	Total Time (CPU)	
main	259,500 ms (100%)	184,980 ms (100%)	
ReaderThread	259,500 ms (100%)	259,500 ms (100%)	
Catalina-utility-1	259,500 ms (100%)	0,0 ms (-%)	
Catalina-utility-2	259,500 ms (100%)	0,0 ms (-%)	
container-0	259,500 ms (100%)	0,0 ms (-%)	
mysql-cj-abandoned-connection-cleanup	259,500 ms (100%)	0,0 ms (-%)	
http-nio-8081-BlockPoller	254,299 ms (100%)	0,0 ms (-%)	
http-nio-8081-exec-1	254,299 ms (100%)	10,351 ms (100%)	
http-nio-8081-exec-2	254,299 ms (100%)	12,976 ms (100%)	
http-nio-8081-exec-3	254,299 ms (100%)	7,627 ms (100%)	
http-nio-8081-exec-4	254,299 ms (100%)	8,882 ms (100%)	
http-nio-8081-exec-5	254,299 ms (100%)	10,274 ms (100%)	
http-nio-8081-exec-6	254,299 ms (100%)	5,895 ms (100%)	
http-nio-8081-exec-7	254,299 ms (100%)	9,335 ms (100%)	
http-nio-8081-exec-8	254,299 ms (100%)	8,118 ms (100%)	
http-nio-8081-exec-9	254,299 ms (100%)	5,032 ms (100%)	
http-nio-8081-exec-10	254,299 ms (100%)	6,844 ms (100%)	
Name	Self Time (CPU)	Total Time (CPU)	
org.apache.commons.exec.StreamPumper.run()	0,0 ms (0%)	506,413 ms (2,4%)	
org.eclipse.jdt.internal.junit.runner.RemoteTestRunner\$ReaderThread.run()	0,0 ms (0%)	259,500 ms (1,2%)	
org.eclipse.jdt.internal.junit.runner.RemoteTestRunner.main()	0,0 ms (0%)	184,980 ms (0,9%)	
org.eclipse.jdt.internal.junit.runner.RemoteTestRunner.run()	0,0 ms (0%)	184,980 ms (0,9%)	
org.eclipse.jdt.internal.junit.runner.RemoteTestRunner.runTests()	0,0 ms (0%)	184,980 ms (0,9%)	
org.eclipse.jdt.internal.junit.runner.TestExecution.run()	0,0 ms (0%)	184,980 ms (0,9%)	
org.eclipse.jdt.internal.junit5.runner.JUnit5TestReference.run()	0,0 ms (0%)	184,980 ms (0,9%)	
org.junit.platform.launcher.core.DefaultLauncher.execute()	0,0 ms (0%)	184,980 ms (0,9%)	
org.junit.platform.launcher.core.EngineExecutionOrchestrator.execute()	0,0 ms (0%)	184,980 ms (0,9%)	
org.junit.platform.launcher.core.EngineExecutionOrchestrator.withInterceptedStreams()	0,0 ms (0%)	184,980 ms (0,9%)	
org.junit.platform.launcher.core.EngineExecutionOrchestrator.lambda\$execute\$0()	0,0 ms (0%)	184,980 ms (0,9%)	
org.junit.platform.engine.support.hierarchical.HierarchicalTestEngine.execute()	0,0 ms (0%)	184,980 ms (0,9%)	
org.junit.platform.engine.support.hierarchical.HierarchicalTestExecutor.execute()	0,0 ms (0%)	184,980 ms (0,9%)	
org.junit.platform.engine.support.hierarchical.SameThreadHierarchicalTestExecutorService.submit()	0,0 ms (0%)	184,980 ms (0,9%)	
org.junit.platform.engine.support.hierarchical.NodeTestTask.execute()	0,0 ms (0%)	184,980 ms (0,9%)	
org.junit.platform.engine.support.hierarchical.NodeTestTask.executeRecursively()	0,0 ms (0%)	184,980 ms (0,9%)	
org.junit.platform.engine.support.hierarchical.ThrowableCollector.execute()	0,0 ms (0%)	184,980 ms (0,9%)	

Por último, se ha analizado el consumo de recursos del ordenador mientras se ejecuta el proyecto, habiendo obtenido unos resultados muy favorables. El consumo se mantiene medio durante toda la ejecución sin presencia de ningún pico



Hipótesis de contraste

Observamos con la ejecución del test-z que ambos ordenadores han obtenido resultados satisfactorios

Prueba z para medias de dos muestras		
	time 1	time 2
Media	566,3243243	568,3459459
Varianza (conocida)	1108517637	235803086
Observaciones	185	185
Diferencia hipotética de las medias	0	
z	-0,000749953	
P(Z<=z) una cola	0,499700812	
Valor crítico de z (una cola)	1,644853627	
Valor crítico de z (dos colas)	0,999401624	
Valor crítico de z (dos colas)	1,959963985	

Conclusión

Como conclusión, podemos sacar que el rendimiento es uno de los atributos más importantes a tener en cuenta a la hora de realizar un proyecto, ya que si la ejecución de tests o la muestra de vistas necesita grandes tiempos de espera, esto empeora la calidad del producto que estamos creando

Este informe nos ha servido para comprender todos los factores que pueden afectar al rendimiento de un proyecto software de distintas maneras, como el consumo de recursos provocado por dicho proyecto, los tiempos de espera para la ejecución de los tests o las diferencias entre dispositivos que hemos obtenido según sus características de fábrica

Hay que tener por tanto mucho cuidado e intentar mantener siempre el rendimiento lo más alto posible