



---

# PERFORMANCE REPORT

---

---

23 DE MAYO DE 2022

E2.07

<https://github.com/gonmarfer2/Acme-Toolkits>

---

Jaime Borrego Conde: [jaiborcon@alum.us.es](mailto:jaiborcon@alum.us.es)

Antonio Campos Gil: [antcamgil@alum.us.es](mailto:antcamgil@alum.us.es)

Ana Conde Marrón: [anaconmar@alum.us.es](mailto:anaconmar@alum.us.es)

Gonzalo Martínez Fernández: [gonmarfer2@alum.us.es](mailto:gonmarfer2@alum.us.es)

Jaime Moscoso Bernal: [jaimosber@alum.us.es](mailto:jaimosber@alum.us.es)

Enrique Muñoz Pérez: [enrmunper@alum.us.es](mailto:enrmunper@alum.us.es)

Tabla de contenido

Resumen ejecutivo .....2

Tabla de revisión .....3

Introducción .....4

Primer Análisis.....5

Segundo Análisis .....3

Hipótesis de contraste .....2

Conclusiones .....3

Bibliografía .....4

## Resumen ejecutivo

El informe de rendimiento es un documento que recopila métricas relacionadas con este atributo de calidad, en este caso, para el proyecto *Acme Toolkits*. Estas métricas incluyen dos análisis, realizados en ordenadores diferentes, sobre el intervalo de confianza de tiempo real (*wall time*) que abarcan las peticiones, así como un contraste de hipótesis que clarifica qué ordenador es más eficiente dentro de dicho intervalo.

## Tabla de revisión

<i>Número de revisión</i>	<i>Fecha</i>	<i>Descripción</i>
1	23/05/2022	Desarrollo del documento

## Introducción

Los requisitos de rendimiento son requisitos no funcionales que establece el cliente en relación con el funcionamiento de la aplicación, normalmente referidos al tiempo que tarda en ejecutar una petición, en mostrar vistas o en responder a las interacciones del usuario.

En el caso de *Acme-Toolkits*, deben realizarse dos análisis para el tiempo real medio de las peticiones del sistema, cada uno en un ordenador diferente, de manera que se clarifique cuál es el ordenador más eficiente dentro del intervalo de confianza del 95%. Para ello, se presenta el siguiente informe en el que se detalla el procedimiento seguido para obtener los resultados.

En primer lugar, se muestra el apartado referido a las pruebas de rendimiento del primer ordenador, es decir, la documentación del análisis estadístico llevado a cabo para reconocer el tiempo que tarda el sistema en servir las peticiones, así como el porqué de ese tiempo.

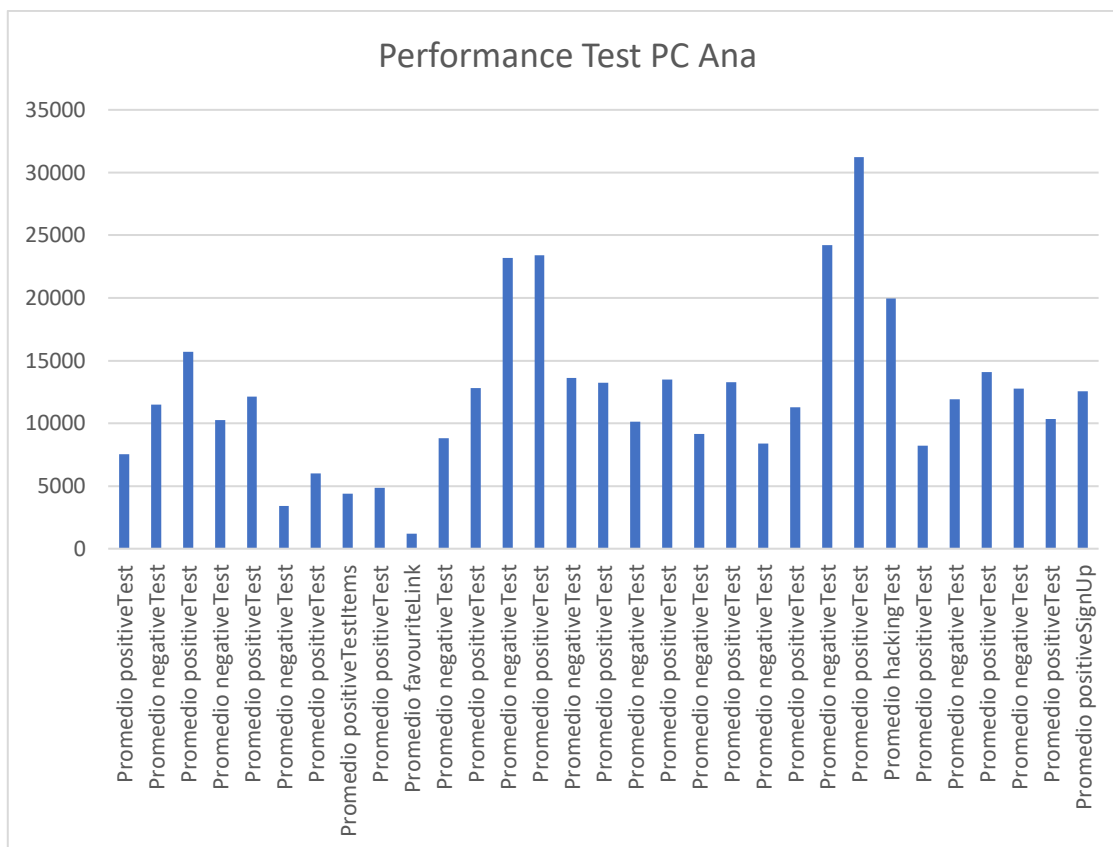
Posteriormente, se incluye el apartado del segundo análisis, que se emplea para realizar una hipótesis de contraste con el primer ordenador.

Finalmente, se ofrecen las conclusiones extraídas del informe que aportan el punto de cierre del documento.

## Primer Análisis

A la hora de evaluar el rendimiento de las solicitudes a la aplicación, la primera vez que se hizo el análisis fue en un equipo con un procesador Intel(R) Core(TM) i5-8265U CPU, 8 GB RAM y Windows 10 Home. Se pretendía hacer un primer análisis, tras este mejorar lo que permitiese llegar al nivel del 95% de confianza y volver a hacer un análisis. Pero, tras obtener los siguientes resultados se decidió que no era necesario refactorizar más la aplicación pues ya se encontraba dentro de los intervalos estipulados de rendimiento.

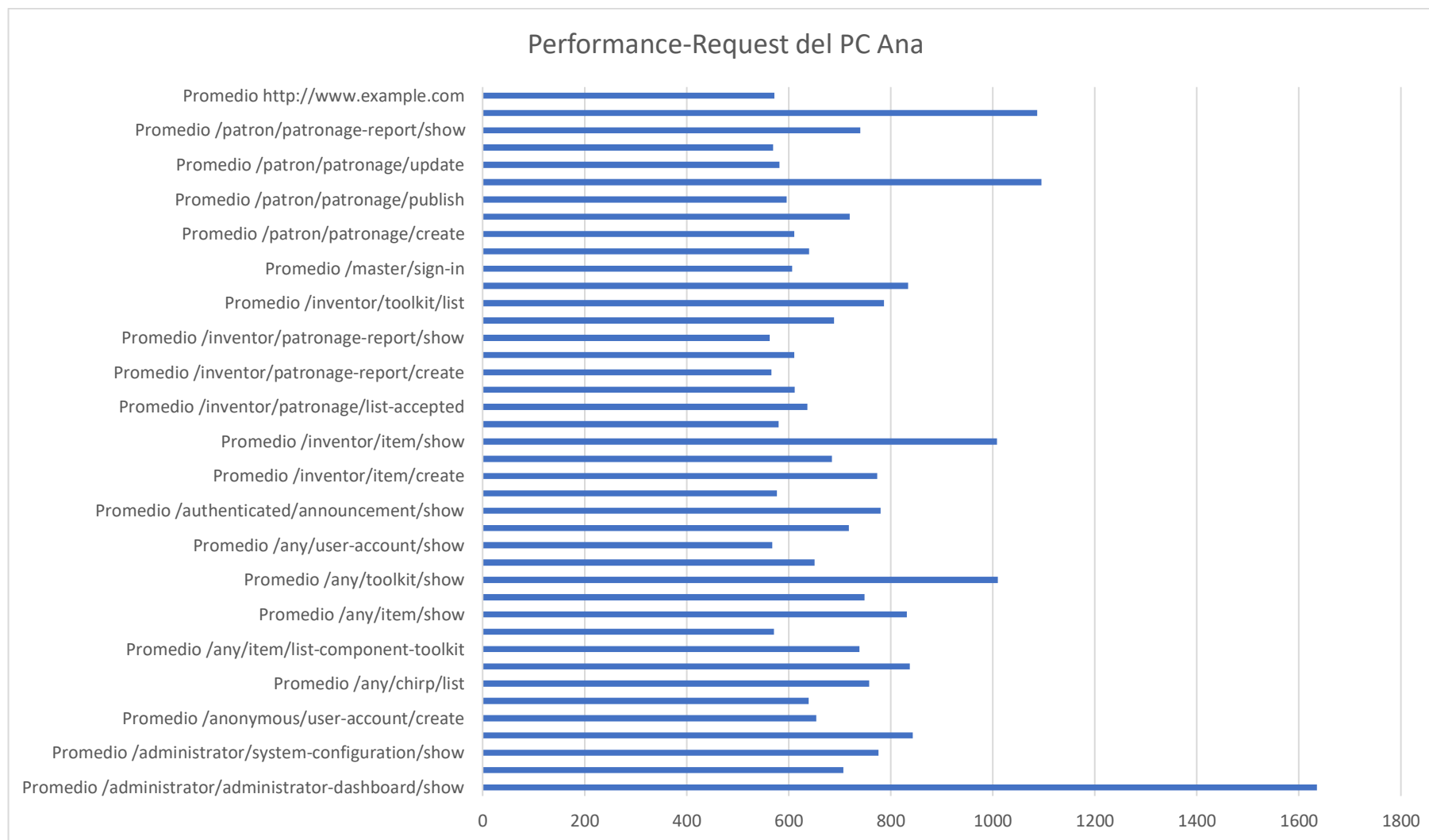
Estos fueron los datos arrojados por el primer análisis:



Gráfica 1. Performance-Test PC Ana

Sin embargo, quizá por un mal nombramiento de los test, la tabla no arroja valores concretos: las etiquetas del eje X se repiten continuamente. El rendimiento obtenido en este análisis, no obstante, es correcto ya que se adapta a los requisitos del cliente. Por tanto, la mayoría de los test tardarán aproximadamente entre 1 y 32 segundos.

A continuación, se muestran los resultados ofrecidos en el análisis de rendimiento de las peticiones, recogidos en un diagrama de barras.



Gráfica 2. Performance-Request del PC Ana

La mayoría de las peticiones se mantienen estables exceptuando algunos casos como los paneles (*DashBoard*), que necesitan un mayor tiempo de ejecución al necesitar realizar cálculos sobre varias entidades. También se ha encontrado un pico de tiempo en *el update de Patronage*.

Tras la ejecución de las pruebas, se ha alcanzado un 68,4% de cobertura. Esto es debido a que en los requisitos del cliente no se especificaba un testeo completo de la aplicación, sino que cada integrante del grupo de *testing* hiciese un test exhaustivo de al menos una funcionalidad. Al haber funcionalidades no cubiertas por test, los resultados que ofrece el análisis son muy pobres, y actúan en detrimento de la media general de cobertura.

> Acme-Framework-22.5.2	56,0 %	12.887	10.137	23.024
Acme-Toolkits	68,4 %	10.278	4.738	15.016
src/main/java	61,2 %	7.424	4.708	12.132
acme.features.inventor.toolkit	18,3 %	265	1.181	1.446
acme.features.inventor.quantity	26,8 %	199	544	743
acme.features.inventor.item	76,2 %	1.355	423	1.778
acme.features.authenticated.inventor	7,6 %	32	388	420
acme.features.authenticated.patron	7,6 %	32	388	420
acme.features.patron.patronage	79,3 %	1.384	362	1.746
acme.features.inventor.patronages	41,5 %	255	359	614
acme.features.authenticated.moneyExchange	30,4 %	100	229	329
acme.features.any.toolkit	61,5 %	323	202	525
acme.features.inventor.patronageReports	81,9 %	453	100	553
acme.features.any.item	80,0 %	353	88	441
acme.features.administrator.configuration	76,7 %	250	76	326
acme.features.any.chirp	80,1 %	305	76	381
acme.features.administrator.announcement	79,3 %	207	54	261
acme.features.authenticated.announcement	76,5 %	143	44	187
acme.features.patron.patronageReport	80,9 %	174	41	215
acme.features.any.userAccount	80,5 %	132	32	164
acme.roles	63,4 %	52	30	82
acme.features.administrator.dashboard	95,5 %	468	22	490
acme.features.authenticated.configuration	74,1 %	63	22	85
acme.features.patron.dashboard	91,1 %	224	22	246
acme.components	67,7 %	21	10	31
acme.forms	94,9 %	168	9	177
acme.entities.patronageReport	93,3 %	42	3	45
acme.entities.quantity	87,5 %	21	3	24
acme.datatypes	100,0 %	3	0	3
acme.entities.announcement	100,0 %	38	0	38
acme.entities.chirp	100,0 %	38	0	38
acme.entities.configuration	100,0 %	45	0	45
acme.entities.item	100,0 %	90	0	90
acme.entities.patronage	100,0 %	114	0	114
acme.entities.toolkits	100,0 %	52	0	52
acme.features.spam	100,0 %	23	0	23
src/test/java	99,0 %	2.854	30	2.884
acme.testing	84,1 %	159	30	189
acme.testing.administrator.administratorDas	100,0 %	16	0	16
acme.testing.administrator.announcement	100,0 %	112	0	112
acme.testing.administrator.systemConfigurat	100,0 %	149	0	149
acme.testing.any.chirp	100,0 %	140	0	140
acme.testing.any.item	100,0 %	89	0	89
acme.testing.any.toolkit	100,0 %	83	0	83
acme.testing.any.userAccount	100,0 %	45	0	45
acme.testing.authenticated.announcements	100,0 %	53	0	53
acme.testing.authenticated.configuration	100,0 %	28	0	28
acme.testing.inventor.item	100,0 %	786	0	786
acme.testing.inventor.patronage	100,0 %	99	0	99
acme.testing.inventor.patronageReport	100,0 %	163	0	163
acme.testing.inventor.toolkit	100,0 %	82	0	82
acme.testing.patron.patronage	100,0 %	774	0	774
acme.testing.patron.patronageReport	100,0 %	60	0	60
acme.testing.patron.patronDashboard	100,0 %	16	0	16

Imagen 1. Nivel de cobertura del proyecto



PC Ana	
Media	671,8938832
Error típico	8,750129376
Mediana	578
Moda	563
Desviación estándar	303,8702116
Varianza de la muestra	92337,1055
Curtosis	260,4886219
Coeficiente de asimetría	12,11463298
Rango	7494
Mínimo	342
Máximo	7836
Suma	810304,0231
Cuenta	1206
Nivel de confianza(95,0%)	17,16718175
Intervalo de Confianza	
	654,7267014      689,0610649

Tabla 1. Estadística descriptiva PC Ana

Como puede observarse en la tabla anterior, el nivel de confianza ha sido de 17,16718175 ms y el intervalo de confianza oscila entre 654,726 y 689,061 que está por debajo de los 1000 milisegundos requeridos. Por lo tanto, el resultado ha sido positivo y aceptable.

Así mismo, se ha realizado el *profiling* tanto del proyecto como del ordenador para encontrar posibles cuellos de botella. En la siguiente página se muestran los resultados obtenidos tras los análisis.

CPU samples Thread CPU time			
Results:   View:     Collected data:  Snapshot			
Name		Total Time	Total Time (CPU)
main		754.835 ms (100%)	525.154 ms (100%)
Reference Handler		754.835 ms (100%)	0,0 ms (-%)
Finalizer		754.835 ms (100%)	7,84 ms (100%)
mysql-cj-abandoned-connection-cleanup		754.835 ms (100%)	281 ms (100%)
Exec Default Executor		754.835 ms (100%)	0,0 ms (-%)
Exec Stream Pumper		754.835 ms (100%)	754.835 ms (100%)
Exec Stream Pumper		754.835 ms (100%)	754.835 ms (100%)
OkHttpClient.ConnectionPool		754.835 ms (100%)	0,0 ms (-%)
RMI TCP Accept-0		754.835 ms (100%)	0,0 ms (-%)
RMI TCP Connection(1)-192.168.56.1		754.835 ms (100%)	754.835 ms (100%)
RMI Scheduler(0)		754.835 ms (100%)	0,0 ms (-%)
JMX server connection timeout 47		754.835 ms (100%)	0,0 ms (-%)
RMI TCP Connection(2)-192.168.56.1		754.835 ms (100%)	754.835 ms (100%)
Okio Watchdog		754.742 ms (100%)	0,0 ms (-%)
Catalina-utility-1		751.533 ms (100%)	0,0 ms (-%)
Catalina-utility-2		751.533 ms (100%)	343 ms (100%)

Name		Self Time (CPU)	Total Time (CPU)
java.lang.Thread.run ()		0,0 ms (0%)	3.247.784 ms (4,9%)
java.net.SocketInputStream.read ()		0,0 ms (0%)	2.019.392 ms (3%)
java.net.SocketInputStream.socketRead ()		0,0 ms (0%)	2.019.392 ms (3%)
java.net.SocketInputStream.socketRead0[native] ()		2.019.392 ms (44,8%)	2.019.392 ms (3%)
java.util.concurrent.ThreadPoolExecutor\$Worker.run ()		0,0 ms (0%)	1.733.962 ms (2,6%)
java.util.concurrent.ThreadPoolExecutor.runWorker ()		0,0 ms (0%)	1.733.962 ms (2,6%)
java.security.AccessController.doPrivileged[native] ()		2.582 ms (0,1%)	1.630.978 ms (2,4%)
java.io.FilterInputStream.read ()		0,0 ms (0%)	1.519.939 ms (2,3%)
sun.rmi.transport.tcp.TCPTransport\$ConnectionHandler.run ()		0,0 ms (0%)	1.509.670 ms (2,3%)
sun.rmi.transport.tcp.TCPTransport\$ConnectionHandler.lambda\$run\$0 ()		0,0 ms (0%)	1.509.670 ms (2,3%)
sun.rmi.transport.tcp.TCPTransport\$ConnectionHandler.lambda\$run\$0 ()		0,0 ms (0%)	1.509.670 ms (2,3%)
sun.rmi.transport.tcp.TCPTransport\$ConnectionHandler.run0 ()		0,0 ms (0%)	1.509.670 ms (2,3%)
sun.rmi.transport.tcp.TCPTransport.handleMessages ()		0,0 ms (0%)	1.509.670 ms (2,3%)
java.io.FileInputStream.read ()		0,0 ms (0%)	1.507.413 ms (2,3%)
java.io.FileInputStream.readBytes[native] ()		1.507.413 ms (33,4%)	1.507.413 ms (2,3%)
java.io.BufferedInputStream.read ()		0,0 ms (0%)	1.506.377 ms (2,3%)
java.io.BufferedInputStream.fill ()		0,0 ms (0%)	1.506.292 ms (2,3%)
sun.misc.reflect.Method.invoke ()		0,0 ms (0%)	1.370.064 ms (2,1%)

Imagen 2. Project Profiling Ana

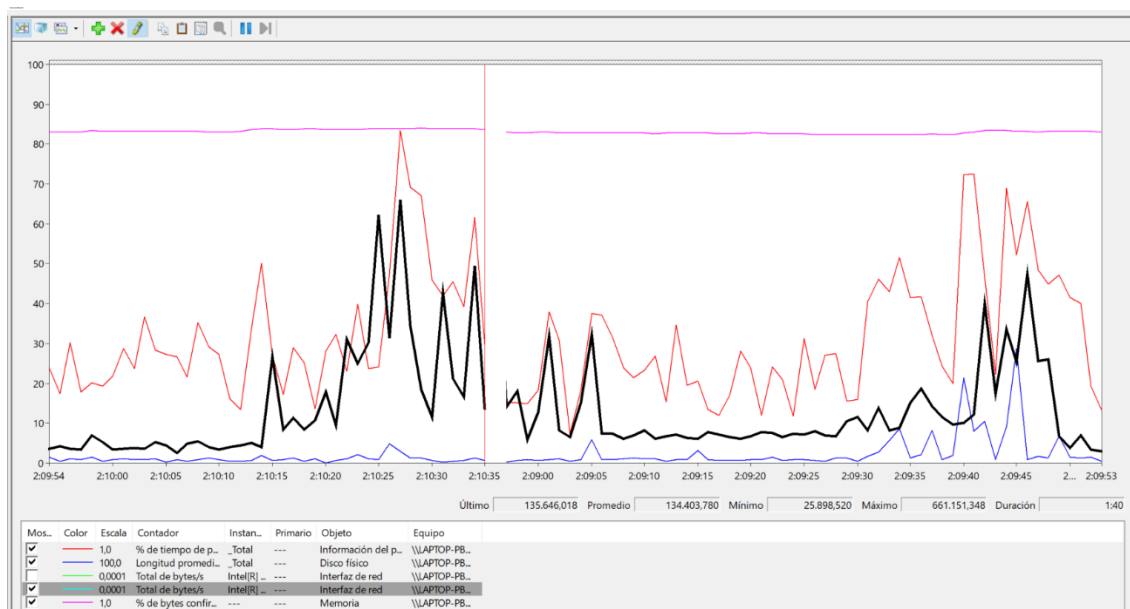
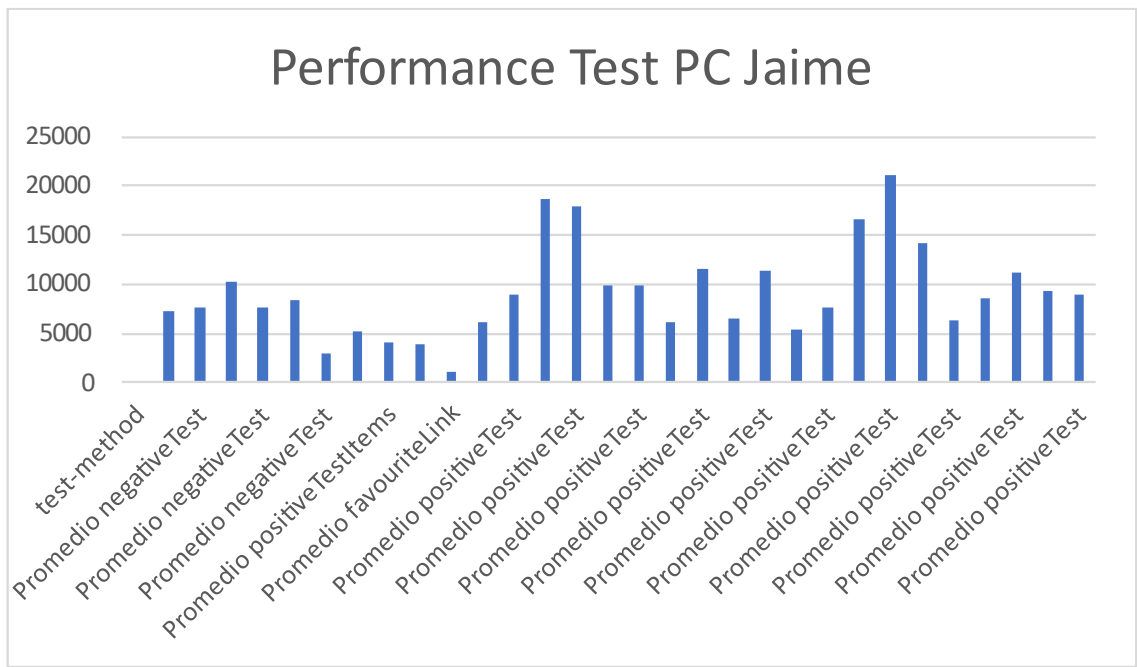


Imagen 3. Computer Profiling Ana

El consumo de recursos del sistema ha sido no ha sido pequeño. No obstante, a pesar de observar la presencia de picos de consumo, se considera aceptable, pues estos han sido casos puntuales y no alargados en el tiempo.

## Segundo Análisis

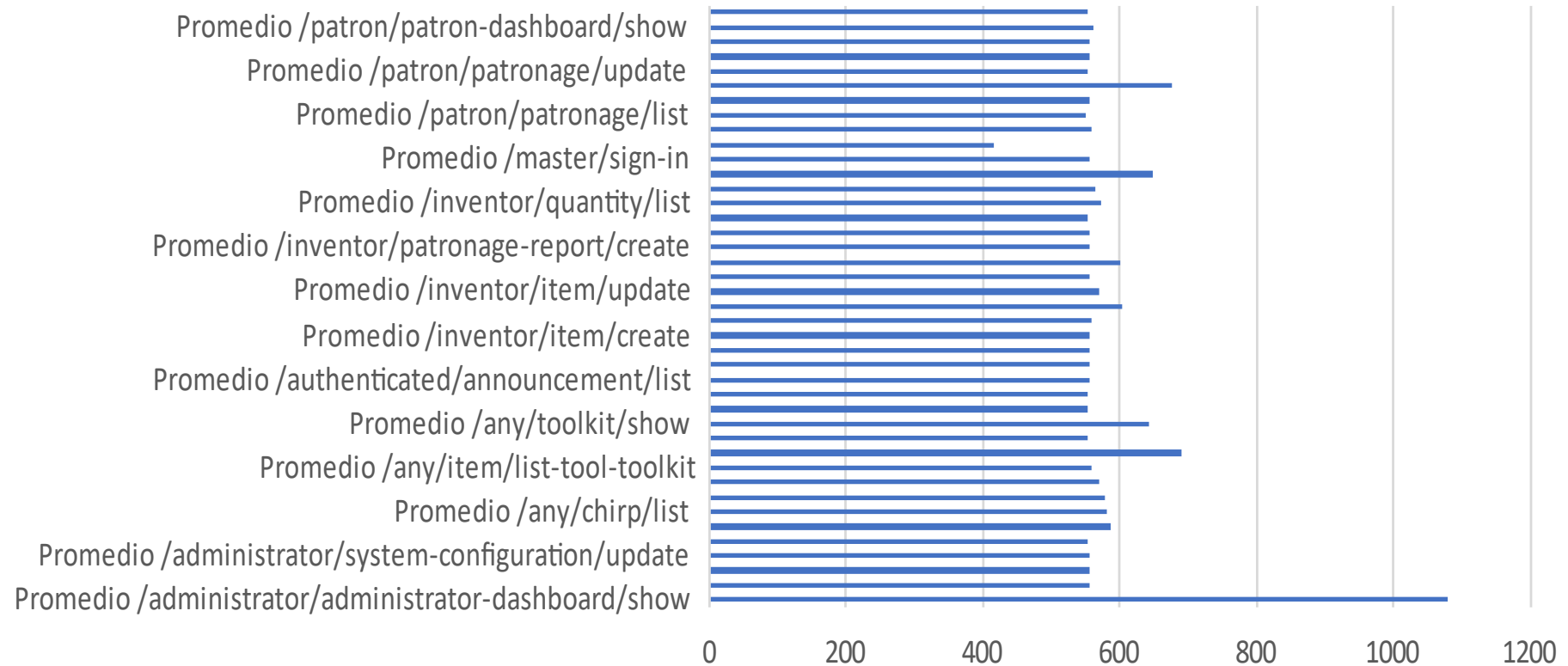
El segundo análisis se realizó sobre un equipo con un procesador AMD Ryzen 7 4800HS, 16 GB RAM y Windows 11 Home. A continuación, se muestran los resultados obtenidos:



Gráfica 3. Performance Test PC Jaime

Como se puede observar, este ordenador era más potente y su capacidad se ve reflejada en la ejecución de los test. El tiempo total de ejecución se encontró entre 1 y 21 segundos, mejorando así los tiempos de los test del primer análisis.

## Performance Request PC Jaime



Gráfica 4. Performance Request PC Jaime

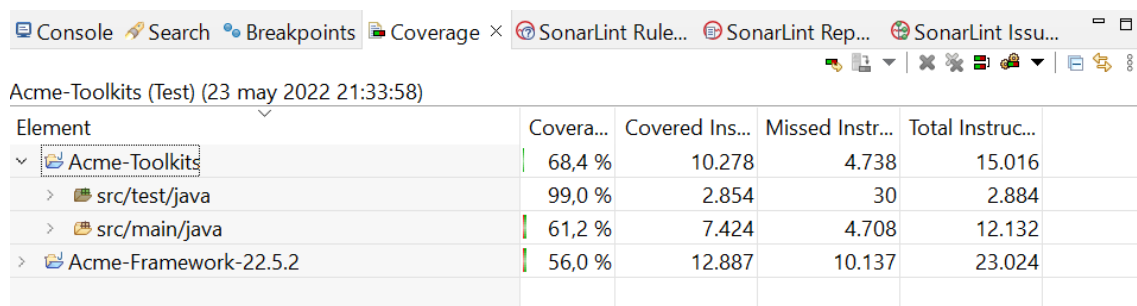
La mayoría de las peticiones se mantienen estables, de manera similar al primer análisis, exceptuando algunos casos como el *DashBoard* de administrador que necesita un mayor tiempo de ejecución debido a los cálculos necesarios. Sorprendentemente, en este PC el *DashBoard* de patrocinador obtiene un valor similar al promedio. En la clase *Patron Patronage Update* se vuelven a obtener valores muy altos, por lo que se descarta que sea debido a la ejecución en paralelo.

PC Jaime	
Media	494,833814
Error típico	5,171749201
Mediana	554
Moda	554
Desviación estándar	179,6019758
Varianza de la muestra	32256,8697
Curtosis	156,1508138
Coefficiente de asimetría	7,434079859
Rango	4041
Mínimo	188
Máximo	4229
Suma	596769,5797
Cuenta	1206
Nivel de confianza(95,0%)	10,1466338
Intervalo de Confianza	
	484,6871802      504,9804478

Tabla 2. Estadística Descriptiva PC Jaime

En esta ocasión, el nivel de confianza es de 10,15 y el intervalo de confianza oscila entre 484,68 y 504,98 milisegundos, por lo que los resultados obtenidos son mejores y, por tanto, positivos.

La cobertura del proyecto no ha variado con respecto al primer análisis porque este no ha cambiado entre uno y otro. Por tanto, a modo de resumen, se mostrará la siguiente tabla de nivel de cobertura.



Element	Covera...	Covered Ins...	Missed Instr...	Total Instruc...
Acme-Toolkits	68,4 %	10.278	4.738	15.016
src/test/java	99,0 %	2.854	30	2.884
src/main/java	61,2 %	7.424	4.708	12.132
Acme-Framework-22.5.2	56,0 %	12.887	10.137	23.024

Imagen 4. Nivel de Cobertura

También se ha vuelto a realizar el *profiling* del proyecto y del ordenador para encontrar cuellos de botella. Los resultados obtenidos de este proceso son los siguientes:

CPU samples				Thread CPU time			
Results:		View:		Collected data:		Thread Dump	
Name	Total Time	Total Time (CPU)		Name	Self Time (CPU)	Total Time (CPU)	
Forwarding findElement on session 71	640.998 ms (100%)	474.659 ms (100%)		java.lang.Thread.run ()	0,0 ms (0%)	2.658.241 ms (4,8%)	
Reference Handler	640.998 ms (100%)	0,0 ms (-%)		java.net.SocketInputStream.read ()	0,0 ms (0%)	1.751.112 ms (3,1%)	
Finalizer	640.998 ms (100%)	0,0 ms (-%)		java.net.SocketInputStream.socketRead ()	0,0 ms (0%)	1.751.112 ms (3,1%)	
mysql-cj-abandoned-connection-cle	640.998 ms (100%)	0,0 ms (-%)		java.net.SocketInputStream.socketRead0[n	1.750.900 ms (46,5%)	1.751.112 ms (3,1%)	
Exec Default Executor	640.998 ms (100%)	0,0 ms (-%)		java.util.concurrent.ThreadPoolExecutor\$Work	0,0 ms (0%)	1.375.139 ms (2,5%)	
Exec Stream Pumper	640.998 ms (100%)	640.998 ms (100%)		java.util.concurrent.ThreadPoolExecutor.runW	0,0 ms (0%)	1.375.139 ms (2,5%)	
Exec Stream Pumper	640.998 ms (100%)	640.998 ms (100%)		java.security.AccessController.doPrivileged[	2.106 ms (0,1%)	1.331.464 ms (2,4%)	
				java.io.FilterInputStream.read ()	0,0 ms (0%)	1.291.498 ms (2,3%)	

Imagen 5. Project Profiling PC2

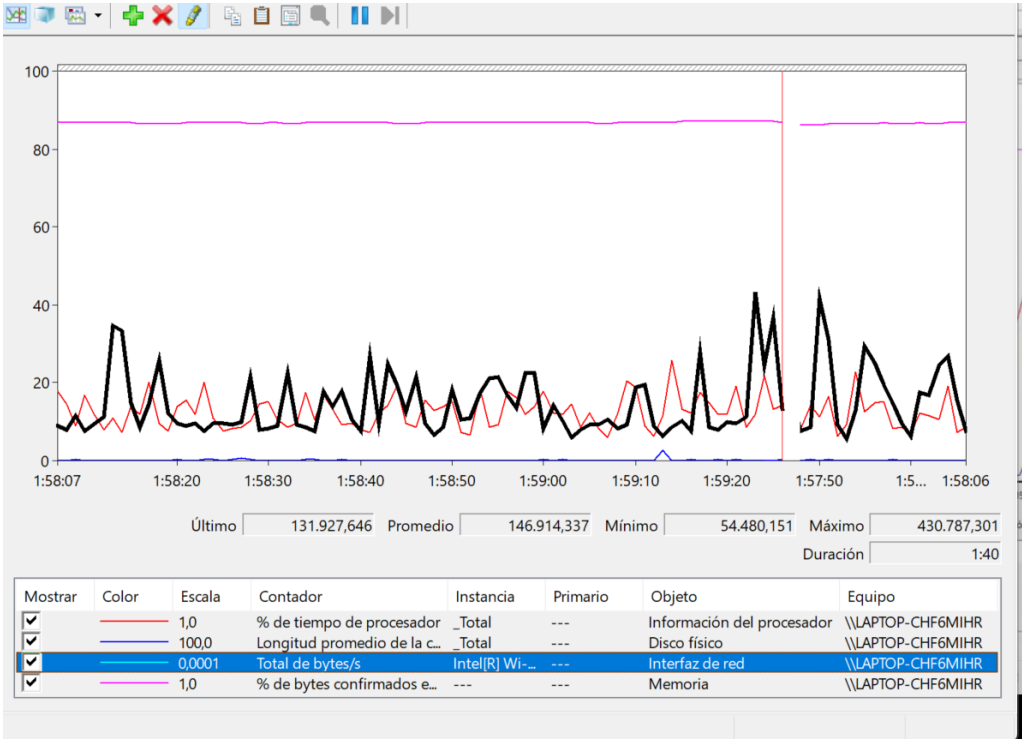


Imagen 6. Computer Profiling PC2

Cabe destacar la homogeneidad en la utilización de recursos, ya que no ha habido ningún aumento prolongado en el consumo de recursos del sistema.

## Hipótesis de contraste

A continuación, se introduce la comparación de los resultados obtenidos entre los dos PC a través de un Z-Test.

Prueba z para medias de dos muestras		
	PC Ana	PC Jaime
Media	671,8938832	494,833814
Varianza (conocida)	92337,1055	32256,8697
Observaciones	1206	1206
Diferencia hipotética de las medias		0
z	17,4199042	
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	

Tabla 3. Z-Test

Se puede observar que, pese a que los resultados obtenidos son satisfactorios para ambos ordenadores, uno es mejor que otro. Se sospecha que PC Jaime es más potente, por tanto, para cerciorarse de que es mejor, se realiza un Z-Test. El resultado de P(Z) es menor que alfa (0.05).

## Conclusiones

El rendimiento probablemente sea de los atributos de calidad más importantes de un proyecto, pues si un sistema presenta grandes tiempos de espera u opera de manera muy lenta perjudica enormemente la experiencia del cliente con el mismo, así como el nombre de la empresa que crea la aplicación.

Por ello, se debe trabajar con sumo cuidado que el rendimiento del sistema sea óptimo. Este documento ha servido como apoyo para entender el estudio de esta característica inherente del sistema, pues gracias a todos los análisis, poco a poco se ha ido desgranando el funcionamiento interno del proyecto, mostrando todos los comportamientos que provocaban retardos en la ejecución, bien por un cálculo excesivo de datos o por una cantidad de bucles inusitada.

Otro de los aspectos que ha ayudado a comprender el funcionamiento del desarrollo del proyecto ha sido poder comparar los resultados entre dispositivos, lo que ha otorgado el conocimiento sobre la diferencia que podía existir al probar la aplicación en ordenadores de distintas gamas, marcas o componentes. De esta manera, si el dispositivo de gama baja sufría mucho en la ejecución de las pruebas, se debían implementar los cambios necesarios para aligerar el proceso y mejorar el rendimiento.

Este documento ha sido, por tanto, una grata experiencia que ha aportado enseñanzas valiosas a todos los miembros del grupo y que ha acercado al grupo al desarrollo de un proyecto profesional.



## Bibliografía

Intencionalmente en blanco.