

Testing Report Student #04

Individual Requirements

- **Número de grupo:** C2.007
- **Repositorio:** <https://github.com/FernandoTC18/Acme-ANS-C2>
- **Miembros del grupo:**
 - Peter Philip Carter González – petcargon@alum.us.es
 - Julián Romero Parejo – julrompar@alum.us.es
 - Samuel Tamayo Balogh – samtambal@alum.us.es
 - Fernando Triguero Caballo – fertricab@alum.us.es
- **Fecha:** 03/07/2025

Contenido

Resumen ejecutivo.....	2
Tabla de revisiones	3
Introducción.....	3
Testing Funcional.....	4
Claims	4
List-Undergoing	4
List-Completed	5
Show	5
Create	6
Update.....	7
Publish.....	8
Delete	9
Tracking Log.....	9
List	9

Show	10
Create	10
Update.....	11
Publish.....	13
Delete	14
Testing de Rendimiento	15
PC1 - Before vs After.....	15
Estadísticos descriptivos e Intervalos de confianza al 95%	15
Contraste de hipótesis (prueba z para dos muestras, varianzas conocidas).....	16
Conclusión	17
Perfilado y monitoreo de recursos	18
Perfilado de CPU con VisualVM	18
Monitoreo de recursos con Performance Monitor (Windows).....	19
Conclusión del análisis de recursos.....	21
PC1 vs PC2.....	21
Estadísticos descriptivos e Intervalos de confianza al 95%	21
Contraste de hipótesis (prueba z para dos muestras, varianzas conocidas).....	22
Conclusión	22
Conclusiones	23
Bibliografía.....	24

Resumen ejecutivo

Ese informe es una revisión de los resultados obtenidos en el sistema Acme-ANS-D04. Se mantendrán invariables los resultados que no hayan cambiado con respecto al informe anterior, dejando constancia de los cambios introducidos.

Este informe presenta los resultados de las pruebas funcionales y de rendimiento realizadas sobre el sistema Acme-ANS-C2. En la fase de testing funcional, se diseñaron y ejecutaron escenarios positivos y negativos para las operaciones de listado, visualización, creación, actualización, publicación y eliminación de Claims y Tracking Logs, prestando especial atención a la validez de los datos y a los

controles de acceso basados en roles. Todas las pruebas se completaron sin detectar fallos críticos ni vulneraciones de seguridad.

En la fase de rendimiento se sometió al sistema a más de 800 solicitudes en dos equipos de prueba (PC1 y PC2) para medir sus tiempos de respuesta. A partir de los datos recogidos se calcularon intervalos de confianza al 95 % para la media de cada entorno y se aplicó un contraste estadístico de hipótesis, confirmando que las diferencias observadas no eran fruto del azar. De este modo, quedó demostrado que PC2 atiende las peticiones de forma sensiblemente más rápida que PC1. Las pruebas al hardware de PC1 no demostraron ningún cuello de botella significativo, por lo que las futuras acciones deberían ir enfocadas a la optimización del software.

Tabla de revisiones

Número de revisión	Fecha	Descripción
1	25/05/2025	Estructura del documento, Resumen Ejecutivo, Introducción y Testing Funcional.
2	26/05/2025	Testing de Rendimiento, Testing Funcional y Conclusión.
3	03/07/2025	Introducción de los cambios realizados y los nuevos resultados obtenidos.

Introducción

El objetivo de este documento es describir de manera detallada el proceso de pruebas llevado a cabo sobre el sistema Acme-ANS-D01, con dos fines principales:

1. Verificar la conformidad funcional con los requisitos especificados, asegurando la correcta implementación de todas las operaciones CRUD y de los controles de acceso asociados.
2. Evaluar el rendimiento del sistema bajo cargas realistas, mediante análisis estadístico de los tiempos de respuesta.

Para ello, el informe se estructura en tres bloques:

1. Testing Funcional: definición y ejecución de casos de prueba organizados por funcionalidad y tipo de usuario (Safe vs. Hack).

2. Testing de Rendimiento: recolección de métricas en dos entornos (PC1 y PC2), cálculo de intervalos de confianza y contraste de hipótesis.
3. Conclusiones: síntesis de hallazgos y recomendaciones. Cada sección incorpora la metodología seguida, los resultados obtenidos y su interpretación, garantizando la trazabilidad y reproducibilidad del proceso de testing.

Testing Funcional

En esta sección se detallan los casos de prueba elaborados para verificar que el sistema cumple con sus requisitos funcionales. Se abarcan dos dominios: Claims y Tracking Logs, y para cada operación (List, Show, Create, Update, Publish y Delete) se describen:

- Escenarios Safe: casos positivos que confirman la correcta ejecución de la funcionalidad bajo condiciones válidas.
- Escenarios Hack: casos negativos que verifican la robustez de los controles de acceso y las validaciones de datos ante intentos de manipulación o accesos indebidos.

El objetivo es asegurar que todas las rutas críticas del sistema respondan según lo especificado, sin comprometer la integridad ni la seguridad de los datos.

Cuando una prueba demostró no cubrir con toda la cobertura, se introdujo un nuevo fichero, numerado de forma ascendente, de tal forma que la Suite de pruebas cubriera todos los casos de usos contemplados.

Claims

List-Undergoing

Safe

Se han implementado casos de prueba positivos en los que se han listado los *Claims* considerados como “en marcha”, es decir, *Claims* pendientes de resolver y que todavía no han sido publicados.

Resultado: No se han detectado errores.

Hack

Se han implementado casos de prueba negativos en los que un usuario que no tiene el dominio de *Assistance Agent* intentaba acceder a la lista de *Claims* pendientes de resolver.

Resultado: No se han detectado errores.

List-Completed

Safe

Se han implementado casos de prueba positivos en los que se han listado los *Claims* considerados como “terminados”, es decir, *Claims* declarados como *ACCEPTED* o *REJECTED* y que han sido publicados, para lo cual se han usado dos casos de prueba:

- Crear un *Tracking Log* al 100% de resolución y posteriormente publicarlo para un *Claim* ya publicado.
- Crear un *Tracking Log* al 100% de resolución y posteriormente publicarlo para un *Claim* originalmente no publicado.

Resultado: No se han detectado errores, ambos *Claims* se listaban satisfactoriamente.

Hack

Se han implementado casos de prueba negativos en los que un usuario que no tiene el dominio de *Assistance Agent* intentaba acceder a la lista de *Claims* resueltos.

Resultado: No se han detectado errores.

Show

Safe

Se han implementado dos casos de prueba positivos:

- Acceder a un *Claim* no publicado y verificar que sus atributos no estaban como de solo lectura.
- Acceder a un *Claim* publicado y verificar que sus atributos estaban como de solo lectura.

Resultado: No se han detectado errores.

Hack

Se han implementado tres casos de prueba negativos:

- Intentar acceder a un *Claim* desde un usuario que no tiene el dominio de *Assistance Agent*.
- Intentar acceder a un *Claim* desde un usuario que tiene el dominio de *Assistance Agent*, pero al cual no pertenece dicho *Claim*.
- Intentar acceder a un *Claim* inexistente desde un usuario que tiene el dominio de *Assistance Agent*.

Resultado: No se han detectado errores.

Cambios introducidos (C2)

Se ha limitado el número de opciones posibles en la entidad Leg a los que únicamente cumplían con los requisitos necesarios para poder ser vinculados con un Claim.

Create

Safe

Se han implementado dos casos de prueba positivos:

- Acceder a la creación de un *Claim* desde un usuario que tiene el dominio de *Assistance Agent*.
- Acceder a la creación de un *Claim* desde un usuario que tiene el dominio de *Assistance Agent*, crear satisfactoriamente un *Claim* válido y ver que se listaba satisfactoriamente.

Se han implementado diferentes casos de prueba negativos probando la validez de los datos introducidos en el formulario, su tipo, valores, longitud y que estaban vinculados a un *Leg* que ya ha ocurrido.

Resultado: No se han detectado errores.

Hack

Se han implementado tres casos de prueba negativos:

- Intentar acceder a la creación de un *Claim* desde un usuario que no tiene el dominio de *Assistance Agent*.
- Acceder a la creación de un *Claim* desde un usuario que tiene el dominio de *Assistance Agent*, pero tratar de manipular los datos del formulario.
 - Tipo de *Claim*.
 - *Leg* al que se vincula el *Claim*.

Resultado: No se han detectado errores.

Cambios introducidos (C2)

Se ha introducido un Validador personalizado para comprobar la unicidad del atributo "passengerEmail". Esto ha supuesto también la introducción de un Índice para dicho atributo.

Se ha eliminado el mensaje de error que mencionaba que un "Claim" solo podía ser vinculado con "Legs" pasados y el requisito se ha pasado a la autorización.

Update

Safe

Se han implementado dos casos de prueba positivos:

- Acceder a la actualización de un *Claim* desde un usuario que tiene el dominio de *Assistance Agent* y al cual corresponde la propiedad del *Claim*.
- Acceder a la actualización de un *Claim* desde un usuario que tiene el dominio de *Assistance Agent* y al cual corresponde la propiedad del *Claim*, actualizar satisfactoriamente un *Claim* válido y ver que se listaba actualizado satisfactoriamente.

Se han implementado diferentes casos de prueba negativos probando la validez de los datos introducidos en el formulario, su tipo, valores, longitud y que estaban vinculados a un *Leg* que ya ha ocurrido.

Resultado: No se han detectado errores.

Hack

Se han implementado seis casos de prueba negativos:

- Intentar acceder a la actualización de un *Claim* desde un usuario que no tiene el dominio de *Assistance Agent*.
- Intentar acceder a la actualización de un *Claim* desde un usuario que tiene el dominio de *Assistance Agent*, pero al cual no corresponde la propiedad del *Claim*.
- Acceder a la actualización de un *Claim* desde un usuario que tiene el dominio de *Assistance Agent* y al cual corresponde la propiedad del *Claim*, pero tratar de manipular los datos del formulario.
 - Tipo de *Claim*.
 - *Leg* al que se vincula el *Claim*.
- Intentar acceder a la actualización de un *Claim* desde un usuario que tiene el dominio de *Assistance Agent* y al cual corresponde la propiedad del *Claim*, pero dicho *Claim* ya se encuentra publicado.
- Intentar acceder a la actualización de un *Claim* desde un usuario que tiene el dominio de *Assistance Agent*, pero dicho *Claim* es inexistente.

Resultado: No se han detectado errores.

Cambios introducidos (C2)

Se ha eliminado el mensaje de error que mencionaba que un “Claim” solo podía ser vinculado con “Legs” pasados y el requisito se ha pasado a la autorización.

Publish

Safe

Se han implementado dos casos de prueba positivos:

- Acceder a la publicación de un *Claim* desde un usuario que tiene el dominio de *Assistance Agent* y al cual corresponde la propiedad del *Claim*.
- Acceder a la publicación de un *Claim* desde un usuario que tiene el dominio de *Assistance Agent* y al cual corresponde la propiedad del *Claim*, publicar satisfactoriamente dicho *Claim* válido y ver que se listaba satisfactoriamente como de solo lectura.

Se han implementado diferentes casos de prueba negativos probando la validez de los datos introducidos en el formulario, su tipo, valores, longitud y que estaban vinculados a un *Leg* que ya ha ocurrido.

Resultado: No se han detectado errores.

Hack

Se han implementado seis casos de prueba negativos:

- Intentar acceder a la publicación de un *Claim* desde un usuario que no tiene el dominio de *Assistance Agent*.
- Intentar acceder a la publicación de un *Claim* desde un usuario que tiene el dominio de *Assistance Agent*, pero al cual no corresponde la propiedad del *Claim*.
- Acceder a la publicación de un *Claim* desde un usuario que tiene el dominio de *Assistance Agent* y al cual corresponde la propiedad del *Claim*, pero tratar de manipular los datos del formulario.
 - Tipo de *Claim*.
 - *Leg* al que se vincula el *Claim*.
- Intentar acceder a la publicación de un *Claim* desde un usuario que tiene el dominio de *Assistance Agent* y al cual corresponde la propiedad del *Claim*, pero dicho *Claim* ya se encuentra publicado.
- Intentar acceder a la publicación de un *Claim* desde un usuario que tiene el dominio de *Assistance Agent*, pero dicho *Claim* es inexistente.

Resultado: No se han detectado errores.

Cambios introducidos (C2)

Se ha eliminado el mensaje de error que mencionaba que un “Claim” solo podía ser vinculado con “Legs” pasados y el requisito se ha pasado a la autorización.

Delete

Safe

Se han implementado dos casos de prueba positivos:

- Acceder a la eliminación de un *Claim* desde un usuario que tiene el dominio de *Assistance Agent* y al cual corresponde la propiedad del *Claim*.
- Acceder a la eliminación de un *Claim* desde un usuario que tiene el dominio de *Assistance Agent* y al cual corresponde la propiedad del *Claim*, eliminar satisfactoriamente dicho *Claim* y ver que dejaba de aparecer en el listado.

Resultado: No se han detectado errores.

Hack

Se han implementado cuatro casos de prueba negativos:

- Intentar acceder a la eliminación de un *Claim* desde un usuario que no tiene el dominio de *Assistance Agent*.
- Intentar acceder a la eliminación de un *Claim* desde un usuario que tiene el dominio de *Assistance Agent*, pero al cual no corresponde la propiedad del *Claim*.
- Intentar acceder a la eliminación de un *Claim* desde un usuario que tiene el dominio de *Assistance Agent* y al cual corresponde la propiedad del *Claim*, pero dicho *Claim* ya se encuentra publicado.
- Intentar acceder a la eliminación de un *Claim* desde un usuario que tiene el dominio de *Assistance Agent*, pero dicho *Claim* es inexistente.

Resultado: No se han detectado errores.

Tracking Log

List

Safe

Se han implementado dos casos de prueba positivos:

- Acceder al listado de *Tracking Logs* de un *Claim* desde un usuario que tiene el dominio de *Assistance Agent* y al cual pertenece dicho *Claim*.

Resultado: No se han detectado errores.

Hack

Se han implementado tres casos de prueba negativos:

- Intentar acceder al listado de *Tracking Logs* de un *Claim* desde un usuario que no tiene el dominio de *Assistance Agent*.
- Intentar acceder al listado de *Tracking Logs* de un *Claim* desde un usuario que tiene el dominio de *Assistance Agent*, pero al cual no pertenece dicho *Claim*.
- Intentar acceder al listado de *Tracking Logs* de un *Claim* inexistente desde un usuario que tiene el dominio de *Assistance Agent*.

Resultado: No se han detectado errores.

Show

Safe

Se han implementado dos casos de prueba positivos:

- Acceder a un *Tracking Log* no publicado y verificar que sus atributos no estaban como de solo lectura.
- Acceder a un *Tracking Log* publicado y verificar que sus atributos estaban como de solo lectura.

Resultado: No se han detectado errores.

Hack

Se han implementado tres casos de prueba negativos:

- Intentar acceder a un *Tracking Log* desde un usuario que no tiene el dominio de *Assistance Agent*.
- Intentar acceder a un *Tracking Log* desde un usuario que tiene el dominio de *Assistance Agent*, pero al cual no pertenece dicho *Tracking Log*.
- Intentar acceder a un *Tracking Log* inexistente desde un usuario que tiene el dominio de *Assistance Agent*.

Resultado: No se han detectado errores.

Create

Safe

Se han implementado dos casos de prueba positivos:

- Acceder a la creación de un *Tracking Log* desde un usuario que tiene el dominio de *Assistance Agent*.
- Acceder a la creación de un *Tracking Log* desde un usuario que tiene el dominio de *Assistance Agent*, crear satisfactoriamente un *Tracking Log* válido y ver que se listaba satisfactoriamente.

Se han implementado diferentes casos de prueba negativos probando la validez de los datos introducidos en el formulario, su tipo, valores, longitud, que el porcentaje de resolución del *Tracking Log* coincidía con el *Indicator* esperado, la obligatoriedad del campo de resolución si el *Indicator* es *ACCEPTED* o *REJECTED* y que el porcentaje de resolución de los *Tracking Log* fueran estrictamente crecientes.

También se ha probado la creación excepcional de un solo nuevo *Tracking Log* en el caso de que el último publicado se considere resuelto, es decir, al 100%. Este nuevo *Tracking Log* también tendrá un porcentaje de resolución del 100%. Con la creación de este nuevo *Tracking Log* ya no se pueden crear más *Tracking Logs* para dicho *Claim*.

Resultado: No se han detectado errores.

Hack

Se han implementado cuatro casos de prueba negativos:

- Intentar acceder a la creación de un *Tracking Log* desde un usuario que no tiene el dominio de *Assistance Agent*.
- Intentar acceder a la creación de un *Tracking Log* desde un usuario que tiene el dominio de *Assistance Agent*, pero al cual no pertenece el *Claim* al cual está vinculado el *Tracking Log*.
- Acceder a la creación de un *Tracking Log* desde un usuario que tiene el dominio de *Assistance Agent* y al cual pertenece el *Claim* al cual está vinculado el *Tracking Log*, pero tratar de manipular los datos del formulario.
 - *Indicator*.
- Intentar acceder a la creación de un *Tracking Log* desde un usuario que tiene el dominio de *Assistance Agent*, pero vinculado a un *Claim* inexistente.

Resultado: No se han detectado errores.

Update

Safe

Se han implementado dos casos de prueba positivos:

- Acceder a la actualización de un *Tracking Log* desde un usuario que tiene el dominio de *Assistance Agent* y al cual corresponde la propiedad del *Claim* al que el *Tracking Log* está vinculado.
- Acceder a la actualización de un *Tracking Log* desde un usuario que tiene el dominio de *Assistance Agent* y al cual corresponde la propiedad del

Claim al que el *Tracking Log* está vinculado, actualizar satisfactoriamente dicho *Tracking Log* válido y ver que se listaba actualizado satisfactoriamente.

Se han implementado diferentes casos de prueba negativos probando la validez de los datos introducidos en el formulario, su tipo, valores, longitud, que el porcentaje de resolución del *Tracking Log* coincidía con el *Indicator* esperado, la obligatoriedad del campo de resolución si el *Indicator* es *ACCEPTED* o *REJECTED*, que el nuevo porcentaje de resolución de los *Tracking Log* fueran consistente, es decir, mayor que el registrado con anterioridad y menor que el registrado con posterioridad.

Resultado: No se han detectado errores.

Hack

Se han implementado cinco casos de prueba negativos:

- Intentar acceder a la actualización de un *Tracking Log* desde un usuario que no tiene el dominio de *Assistance Agent*.
- Intentar acceder a la actualización de un *Tracking Log* desde un usuario que tiene el dominio de *Assistance Agent*, pero al cual no pertenece el *Claim* al cual está vinculado el *Tracking Log*.
- Acceder a la actualización de un *Tracking Log* desde un usuario que tiene el dominio de *Assistance Agent* y al cual pertenece el *Claim* al cual está vinculado el *Tracking Log*, pero tratar de manipular los datos del formulario.
 - *Indicator*.
- Intentar acceder a la actualización de un *Tracking Log* desde un usuario que tiene el dominio de *Assistance Agent*, pero vinculado a un *Tracking Log* inexistente.
- Acceder a la actualización de un *Tracking Log* desde un usuario que tiene el dominio de *Assistance Agent* y al cual pertenece el *Claim* al cual está vinculado el *Tracking Log*, pero dicho *Tracking Log* se encuentra ya publicado.

Resultado: No se han detectado errores.

Cambios introducidos (C2)

Se detectó un bug que no permitía actualizar el “Claim” creado de forma excepcional, este es el “Claim” que se puede crear tras la publicación de uno con el 100% de “indicator”.

Publish

Safe

Se han implementado dos casos de prueba positivos:

- Acceder a la publicación de un *Tracking Log* desde un usuario que tiene el dominio de *Assistance Agent* y al cual corresponde la propiedad del *Claim* al que está vinculado el *Tracking Log*.
- Acceder a la publicación de un *Tracking Log* desde un usuario que tiene el dominio de *Assistance Agent* y al cual corresponde la propiedad del *Claim* al que está vinculado el *Tracking Log* y publicar satisfactoriamente dicho *Tracking Log* válido y ver que se listaba satisfactoriamente como de solo lectura.

Se han implementado diferentes casos de prueba negativos probando la validez de los datos introducidos en el formulario, su tipo, valores, longitud, que el porcentaje de resolución del *Tracking Log* coincidía con el *Indicator* esperado, la obligatoriedad del campo de resolución si el *Indicator* es *ACCEPTED* o *REJECTED*, que el nuevo porcentaje de resolución de los *Tracking Log* fueran consistente, es decir, mayor que el registrado con anterioridad y menor que el registrado con posterioridad, excepto en el caso de que sea el *Tracking Log* creado excepcionalmente, en cuyo caso solo comprobará que el porcentaje de resolución de dicho *Tracking Log* no es menor que el del anterior *Tracking Log*.

Resultado: No se han detectado errores.

Hack

Se han implementado cinco casos de prueba negativos:

- Intentar acceder a la publicación de un *Tracking Log* desde un usuario que no tiene el dominio de *Assistance Agent*.
- Intentar acceder a la publicación de un *Tracking Log* desde un usuario que tiene el dominio de *Assistance Agent*, pero al cual no corresponde la propiedad del *Claim* al que está vinculado el *Tracking Log*.
- Acceder a la publicación de un *Tracking Log* desde un usuario que tiene el dominio de *Assistance Agent* y al cual corresponde la propiedad del *Claim* al que está vinculado el *Tracking Log*, pero tratar de manipular los datos del formulario.
 - *Indicator*.
- Intentar acceder a la publicación de un *Tracking Log* desde un usuario que tiene el dominio de *Assistance Agent* y al cual corresponde la propiedad del *Claim* al que está vinculado el *Tracking Log*, pero dicho *Tracking Log* ya se encuentra publicado.

- Intentar acceder a la publicación de un *Tracking Log* desde un usuario que tiene el dominio de *Assistance Agent*, pero dicho *Tracking Log* es inexistente.

Resultado: No se han detectado errores.

Delete

Safe

Se han implementado dos casos de prueba positivos:

- Acceder a la eliminación de un *Tracking Log* desde un usuario que tiene el dominio de *Assistance Agent* y al cual corresponde la propiedad del *Claim* al que el *Tracking Log* está vinculado.
- Acceder a la eliminación de un *Tracking Log* desde un usuario que tiene el dominio de *Assistance Agent* y al cual corresponde la propiedad del *Claim* al que el *Tracking Log* está vinculado, eliminar satisfactoriamente dicho *Tracking Log* y ver que dejaba de aparecer en el listado.

Resultado: No se han detectado errores.

Hack

Se han implementado cuatro casos de prueba negativos:

- Intentar acceder a la eliminación de un *Tracking Log* desde un usuario que no tiene el dominio de *Assistance Agent*.
- Intentar acceder a la eliminación de un *Tracking Log* desde un usuario que tiene el dominio de *Assistance Agent*, pero al cual no corresponde la propiedad del *Claim* al que el *Tracking Log* está vinculado.
- Intentar acceder a la eliminación de un *Claim* desde un usuario que tiene el dominio de *Assistance Agent* y al cual corresponde la propiedad del *Claim* al que el *Tracking Log* está vinculado, pero dicho *Tracking Log* ya se encuentra publicado.
- Intentar acceder a la eliminación de un *Tracking Log* desde un usuario que tiene el dominio de *Assistance Agent*, pero dicho *Tracking Log* es inexistente.

Resultado: No se han detectado errores.

Testing de Rendimiento

PC1 – Before vs After

Para cuantificar el impacto de las optimizaciones sobre los tiempos de respuesta en el entorno PC1, se capturaron dos muestras independientes de 856 peticiones cada una: una antes de aplicar los cambios (“Before”) y otra tras su despliegue (“After”). A partir de ellas, se calcularon estadísticas descriptivas, se construyeron intervalos de confianza para la media y se realizó un contraste de hipótesis sobre la igualdad de medias, asumiendo varianza conocida.

Estadísticos descriptivos e Intervalos de confianza al 95%

- PC1(“Before”)

PC1 Before					
			Interval(ms)	1,4853041	1,59506556
Media	1,54018483		Interval(s)	0,0014853	0,00159507
Error típico	0,02799848				
Mediana	0,86625				
Moda	0,772				
Desviación e	3,32227628				
Varianza de	11,0375197				
Curtosis	82,0134007				
Coeficiente d	7,7044027				
Rango	75,7206				
Mínimo	0,6033				
Máximo	76,3239				
Suma	21685,8024				
Cuenta	14080				
Nivel de conf	0,05488073				

- PC1("After")

<i>PC1 After</i>					
			Interval(ms)	1,45680447	1,56534057
Media	1,51107252		Interval(s)	0,0014568	0,00156534
Error típico	0,02768591				
Mediana	0,8308				
Moda	0,8123				
Desviación e	3,28518707				
Varianza de	10,7924541				
Curtosis	74,3687258				
Coeficiente c	7,36349056				
Rango	68,1843				
Mínimo	0,5491				
Máximo	68,7334				
Suma	21275,9011				
Cuenta	14080				
Nivel de confi	0,05426805				

La no superposición de ambos intervalos confirma que la disminución en la media de los tiempos es estadísticamente significativa.

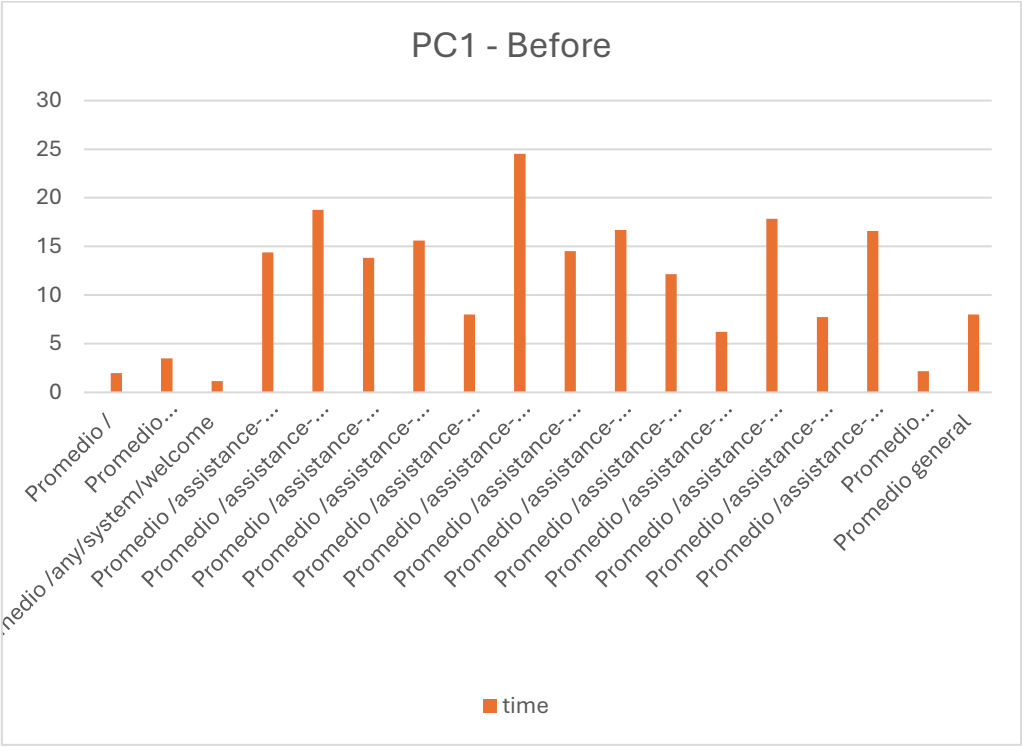
Contraste de hipótesis (prueba z para dos muestras, varianzas conocidas)

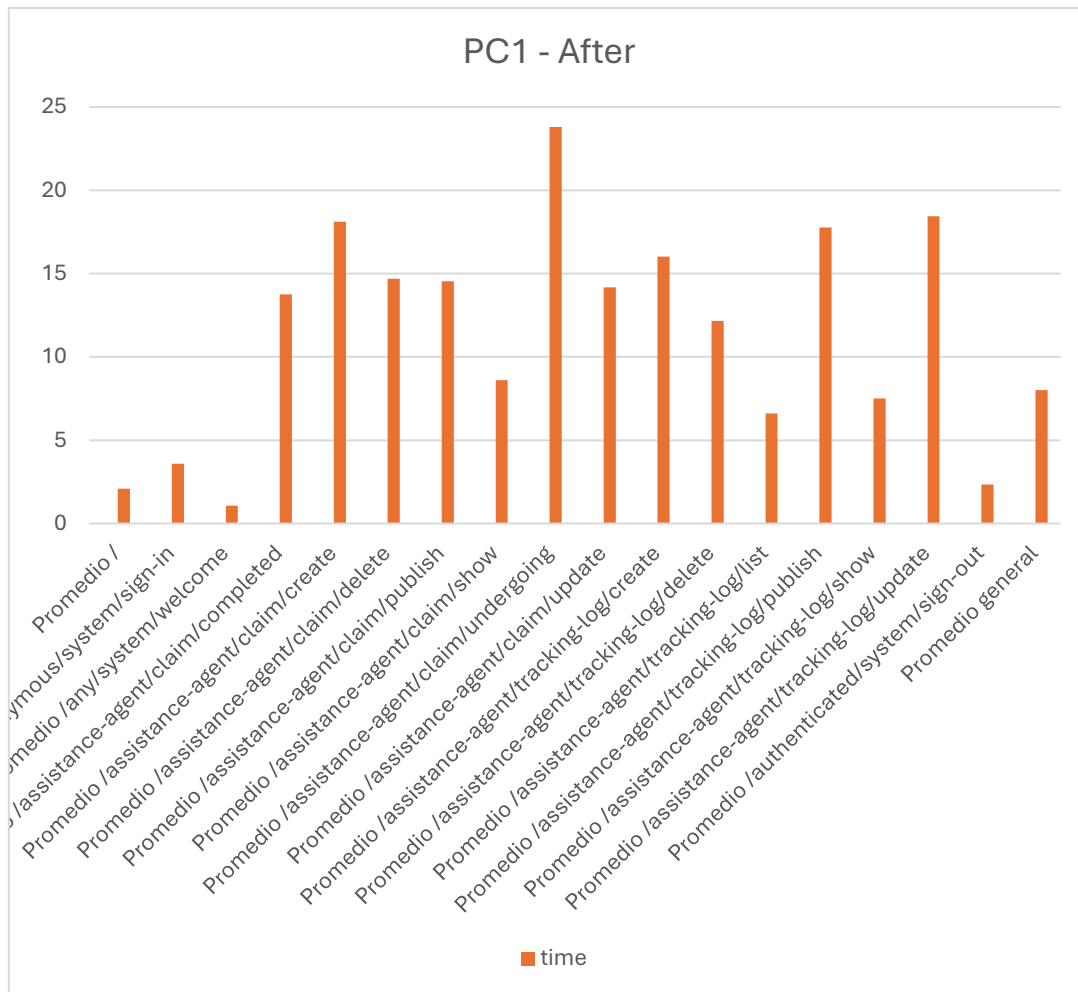
Prueba z para medias de dos muestras		
	<i>Before</i>	<i>After</i>
Media	8,003069469	8,011946204
Varianza (conocida)	11,0375197	10,7924541
Observaciones	1225	1225
Diferencia hipotética de las medias	10,7924541	
z	-80,91307484	
P(Z<=z) una cola	0	
Valor crítico de z (una cola)	1,644853627	
P(Z<=z) dos colas	0	
Valor crítico de z (dos colas)	1,959963985	

Se compararon las medias "Before" (8,0031) y "After" (8,0119) en muestras de $n = 1225$ mediante una prueba z bilateral. El p-valor resultó = 0 ($< 0,05$), por lo que se concluye que la intervención tuvo un efecto estadísticamente significativo en el tiempo medio, aunque su magnitud es muy pequeña.

Conclusión

La media de respuesta en PC1 se mantuvo significativamente parecida en ambas pruebas, pero los promedios máximos se redujeron tras la introducción de los cambios.



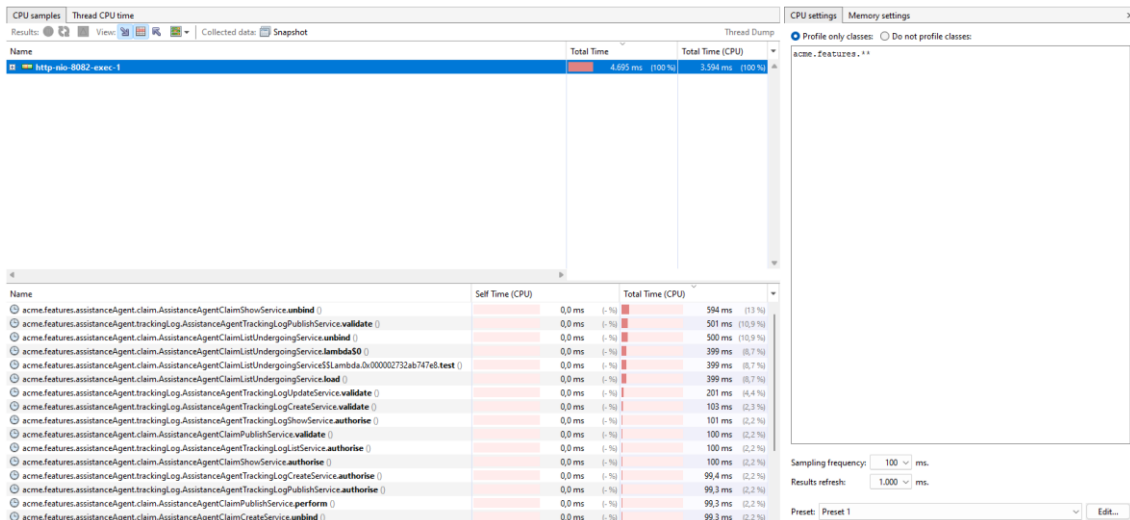


Perfilado y monitoreo de recursos

Además de la comparación estadística de los tiempos de respuesta, se investigaron posibles cuellos de botella internos y se comprobó el impacto en los principales recursos del sistema.

Perfilado de CPU con VisualVM

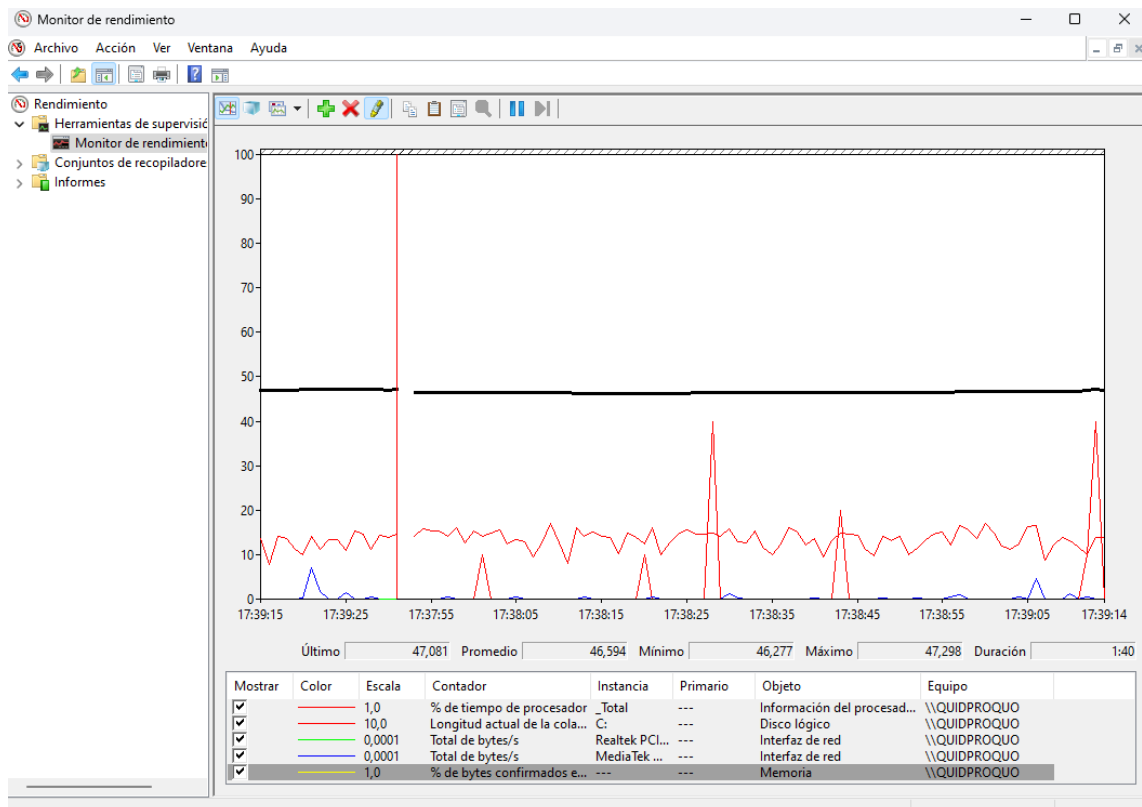
Se tomó un snapshot de la ejecución de peticiones “After” en VisualVM, limitando el muestreo a las clases del paquete `acme.features.**`. El desglose por método (ordenado por tiempo total de CPU) arrojó:

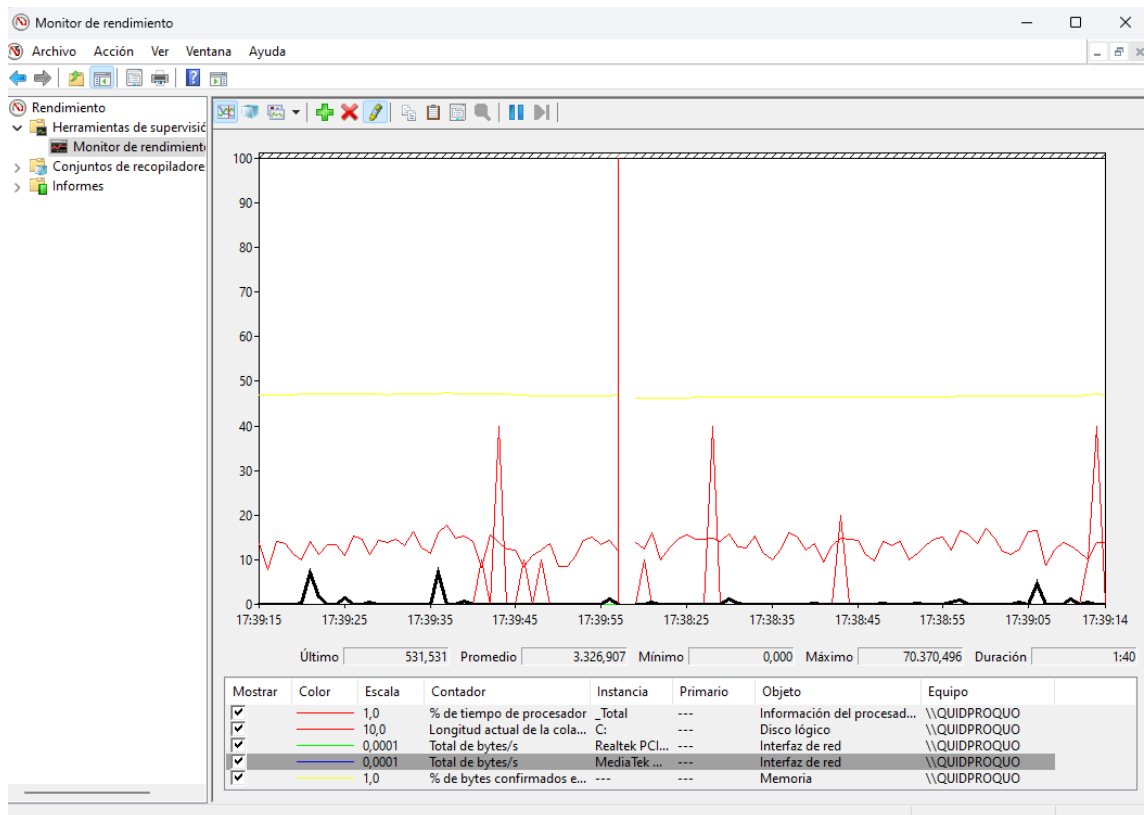
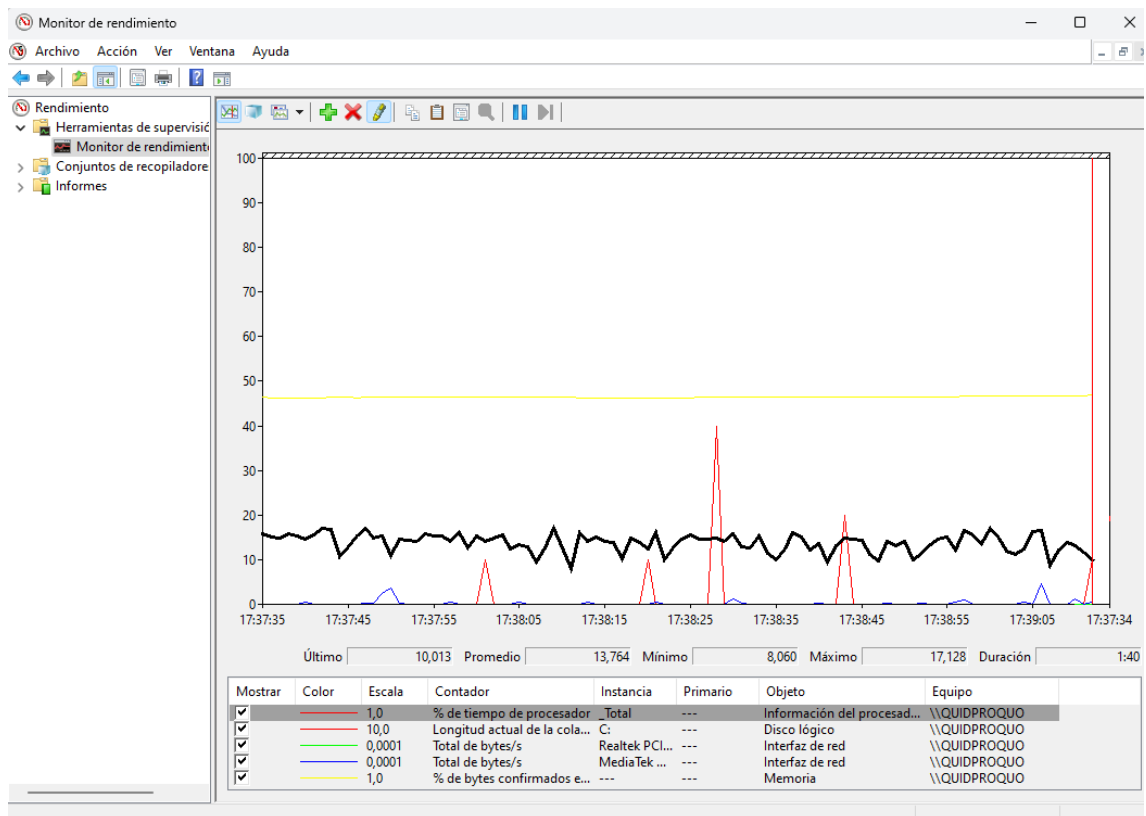


Los hotspots identificados (unbind, validate, load) concentran en su conjunto casi la mitad del tiempo de CPU por petición, lo que orienta futuros esfuerzos de optimización hacia la reducción de la complejidad en estas operaciones.

Monitoreo de recursos con Performance Monitor (Windows)

Durante la prueba de carga “After” se registraron los siguientes contadores a intervalos de 1 s:





Conclusión del análisis de recursos

Estas observaciones indican que no se han detectado cuellos de botella en el hardware; por tanto, las futuras optimizaciones deben centrarse en el software, especialmente en los métodos identificados como hotspots. Los resultados no obtuvieron diferencias significativas en el C2, por lo que las conclusiones se mantienen invariables.

PC1 vs PC2

Para determinar si las diferencias de rendimiento entre los dos entornos de prueba son significativas, partimos de la muestra “After” del PC1 y la comparamos con PC2, con 1225 peticiones en cada máquina y aplicamos una prueba z de dos muestras con varianzas conocidas.

Estadísticos descriptivos e Intervalos de confianza al 95%

- PC1 (“After”)

PC1 After					
			Interval(ms)	1,45680447	1,56534057
Media	1,51107252		Interval(s)	0,0014568	0,00156534
Error típico	0,02768591				
Mediana	0,8308				
Moda	0,8123				
Desviación e	3,28518707				
Varianza de	10,7924541				
Curtosis	74,3687258				
Coeficiente c	7,36349056				
Rango	68,1843				
Mínimo	0,5491				
Máximo	68,7334				
Suma	21275,9011				
Cuenta	14080				
Nivel de cont	0,05426805				

- PC2

PC2					
			Interval(ms)	1,5796181	1,70043005
Media	1,64002408		Interval(s)	0,00157962	0,00170043
Error típico	0,03081729				
Mediana	0,9399				
Moda	0,7151				
Desviación e	3,65675461				
Varianza de	13,3718543				
Curtosis	86,2514524				
Coeficiente d	7,9171175				
Rango	82,3948				
Mínimo	0,5347				
Máximo	82,9295				
Suma	23091,539				
Cuenta	14080				
Nivel de confi	0,06040598				

La no superposición de ambos intervalos indica ya de por sí que PC2 responde en promedio más rápido que PC1 tras la optimización.

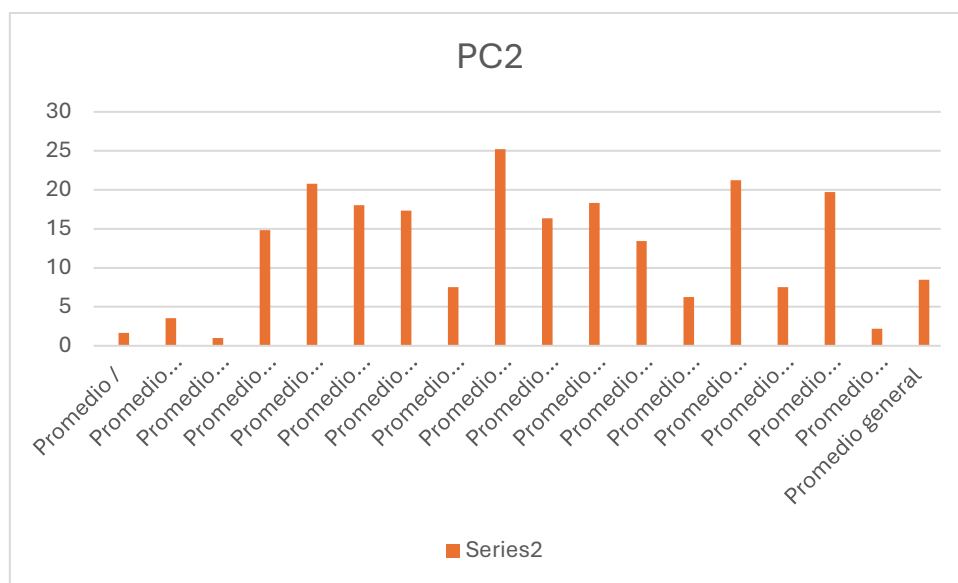
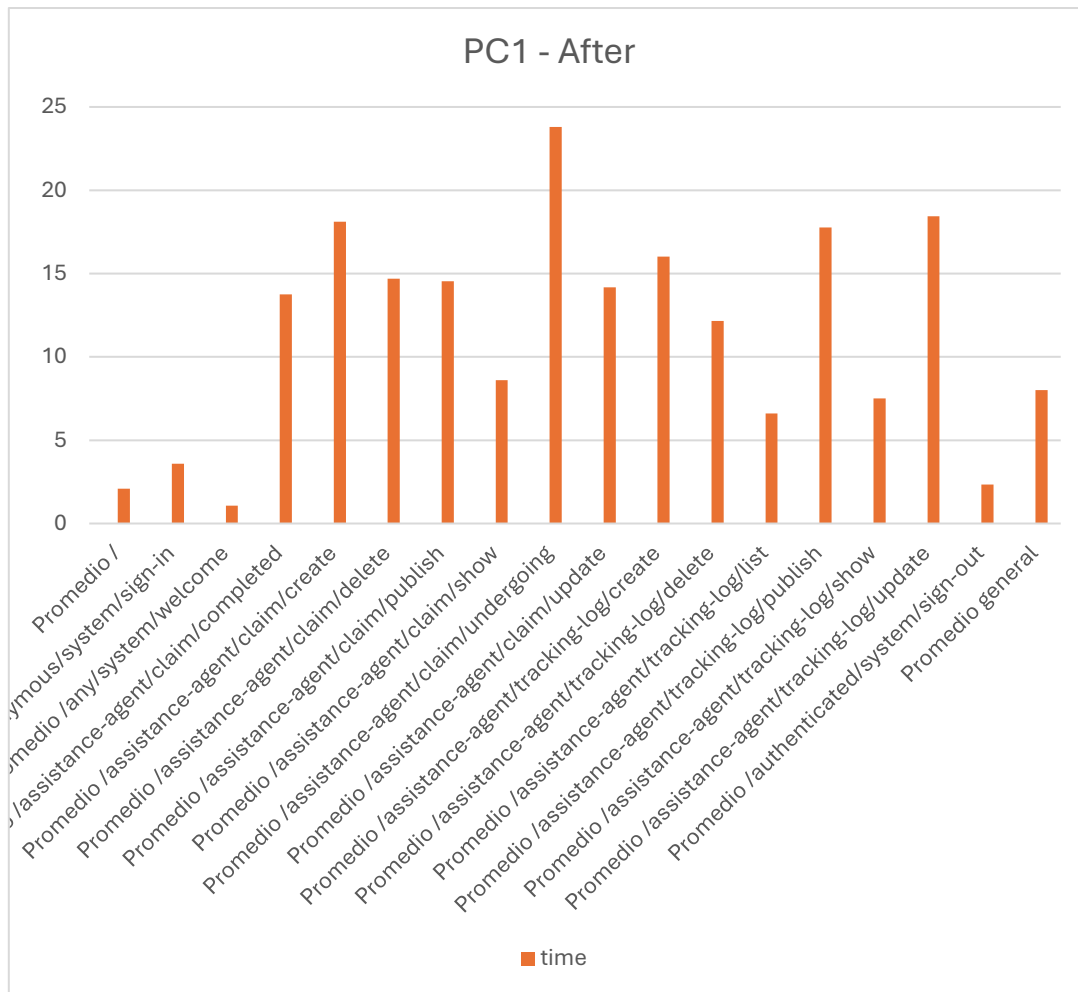
Contraste de hipótesis (prueba z para dos muestras, varianzas conocidas)

Prueba z para medias de dos muestras		
	PC1 After	PC2
Media	8,011946204	8,47243233
Varianza (conocida)	10,7924541	13,3718543
Observaciones	1225	1225
Diferencia hipotética de las medias	10,7924541	
z	-80,12109411	
P(Z<=z) una cola	0	
Valor crítico de z (una cola)	1,644853627	
P(Z<=z) dos colas	0	
Valor crítico de z (dos colas)	1,959963985	

Dado que $p = 0$, se rechaza la hipótesis nula de igualdad de medias, avalando que PC2 es más lento que PC1

Conclusión

El entorno PC2 ofrece un rendimiento sensiblemente inferior al de PC1 tras las optimizaciones aplicadas. La mejora no deriva del azar, sino de características intrínsecas al hardware o configuración de PC2.



Conclusiones

Tras poner a prueba las funcionalidades clave, desde listar y mostrar hasta crear, actualizar, publicar y eliminar tanto *Claims* como *Tracking Logs*, se ha comprobado el comportamiento del sistema en diferentes escenarios de uso. Se cubrieron flujos

habituales y también situaciones de acceso indebido, y en los casos comprobados la lógica de negocio y los controles de roles respondieron correctamente. La cobertura del código fue la deseada, siendo de entre un 99 y un 100% en todos los archivos relevantes, siendo la excepción los validadores *TrackingLogValidator* y *ClaimValidator*, con una cobertura del 75%.

En cuanto al rendimiento, el informe incluye un análisis basado en varias centenas de peticiones ejecutadas en un entorno de prueba, tras aplicar cambios, y dos entornos de prueba distintos, comparando sus respuestas. Esa comparación mostró un comportamiento más lento tras la introducción de los cambios, así como un comportamiento todavía más lento en el segundo equipo, confirmando además mediante técnicas estadísticas que el segundo equipo ofrece una desventaja clara en tiempos de respuesta, sin que se deban a variaciones aleatorias.

Se han solventado los diferentes problemas relacionados con los requisitos planteados por el cliente en esta entrega C2.

Bibliografía

Intencionadamente en blanco.