



---

# INFORME DE TESTING INDIVIDUAL

---

Grupo 1-C1.020 | <https://github.com/PDJ6975/Acme-ANS-D04-25.5.0>



Nombre	Correo Corporativo
Antonio Rodríguez Calderón	antrodcal@alum.us.es
Adrián Ramírez Gil	adrramgil@alum.us.es
Jianwu Hu	jiahu3@alum.us.es
Pablo Castrillón Mora	pabcasmor1@alum.us.es
Pablo Olivencia Moreno	pabolimor@alum.us.es

# Tabla de Contenido

<b>1. TESTING FUNCIONAL</b>	<b>2</b>
1.1 Introducción	2
1.2 Metodología	2
1.3 Casos de Prueba por característica	4
1.4 Calidad de la cobertura	10
1.4.1. Claim	10
1.4.2 Tracking Log	11
1.5 Conclusiones	12
<b>2. Testing de Rendimiento</b>	<b>13</b>
2.1 Entorno y protocolo	13
2.2 Recogida de datos	13
2.3 Estadísticas Descriptivas	14
2.4 Gráficos de eficiencia por característica	15
2.5 hipótesis y conclusión	17
<b>3. Tabla de Revisión</b>	<b>17</b>

# 1. TESTING FUNCIONAL

## 1.1 Introducción

En esta sección del documento se pretende describir un listado con los casos de prueba realizados, agrupados por características, para las entidades Claim y TrackingLog.

Para cada caso de prueba, se va a proporcionar una descripción y cuál es su eficiencia para detectar errores.

## 1.2 Metodología

Para probar las entidades obligatorias del estudiante 4, se han aplicado pruebas “.safe” y “.hack”.

Para las primeras se ha probado que las entidades se listen, muestren y permitan un CRUD válido y funcional, permitiendo todo tipo de valores válidos y rechazando aquellos que no cumplen con la validación.

Para las segundas, se han realizado pruebas de tipo GET/POST hacking para comprobar el correcto funcionamiento del sistema frente a acciones ilegales.

A continuación, se muestra una tabla con la cobertura total alcanzada:

Paquete	Cobertura
<b>Assistance-agent.claim</b>	99.3 %
<b>Assistance-agent.trackingLog</b>	98.1 %

Por otro lado, se muestra una tabla de la cobertura de cada uno de los servicios individualmente de los anteriores paquetes:

Servicio	Cobertura
<b>AssistanceAgentCalimUpdateService</b>	99.1 %
<b>AssistanceAgentCalimCreateService</b>	99.4 %
<b>AssistanceAgentCalimDeleteService</b>	98.6 %
<b>AssistanceAgentCalimPublishService</b>	98.6 %
<b>AssistanceAgentCalimListOngoingService</b>	100 %
<b>AssistanceAgentCalimListService</b>	100 %
<b>AssistanceAgentCalimShowService</b>	100 %

Servicio	Cobertura
<b>AssistanceAgentTrackingLogPublishService</b>	95,3 %
<b>AssistanceAgentTrackingLogCreateService</b>	97,7 %
<b>AssistanceAgentTrackingLogDeleteService</b>	98,4 %
<b>AssistanceAgentTrackingLogListService</b>	100 %
<b>AssistanceAgentTrackingLogShowService</b>	100 %
<b>AssistanceAgentTrackingLogUpdateService</b>	100 %

Para los controladores de ambas entidades se alcanza un 100% de la cobertura, lo que significa que se prueban todas las líneas de código contenidas en ellos (es decir, todos los servicios que ofrecen).

### 1.3 Casos de Prueba por característica

En primer lugar, se muestra los casos de prueba por característica para los servicios relacionados con claim y a continuación se mostrarán los de Tracking log. En ésta, se describe el caso de prueba, el tipo de esta, una descripción resumida de los que se ha probado y la eficacia de la prueba.

Casos de Prueba	Tipo	Descripción	Eficacia
<b>List.safe</b>	Renderizado	Se prueba que el listado se muestra correctamente tanto para los claims finalizados como aquellos que están en curso	No se detectó ninguna incidencia ya que el servicio esta correctamente implementado
<b>List.hack</b>	Hack	Se prueba que: <ul style="list-style-type: none"><li>- Miembros con otro rol no pueden ver el listado de los claims</li><li>- Miembros con el mismo rol solo pueden ver el listado de sus claims</li></ul>	No se detectó ninguna incidencia ya que el servicio esta correctamente implementado tanto en el “authorised” como en la consulta.
<b>Show.safe</b>	Renderizado	Se prueba que un claim se muestre correctamente sin ningún error	No se detectó ninguna incidencia ya que el servicio esta correctamente implementado
<b>Show.hack</b>	Hack	Se prueba que: <ul style="list-style-type: none"><li>- Miembros con otro rol no pueden ver el cualquier calim</li><li>- Miembros con el mismo rol solo pueden ver sus claims</li></ul>	No se detectó ninguna incidencia ya que se realizaba la validación pertinente en el “authorised”

<b>Create.safe</b>	Casos positivos y negativos del formulario	Se prueba que todos los campos del formulario estén correctamente validados, rechazando los valores no admitidos por el modelo y aceptando aquellos que si lo son.	No se detectó ninguna incidencia ya que todos los campos están correctamente validados y se realizan las comprobaciones necesarias en el "validate"
<b>Create.hack</b>	Hack(url y valores)	Se prueba que: <ul style="list-style-type: none"> <li>- Miembros con otro rol no pueden crear o acceder al formulario de creación.</li> <li>- No permite asignar valores ilegales a los atributos de selección.</li> </ul>	No se detectó ninguna incidencia ya que el servicio implementa correctamente el enumerado y se realiza la validación pertinente en el "authorised"
<b>Update.safe</b>	Casos positivos y negativos del formulario	Se prueba que todos los campos del formulario estén correctamente validados, rechazando los valores no admitidos por el modelo y aceptando aquellos que si lo son.	No se detectó ninguna incidencia ya que todos los campos están correctamente validados y se realizan las comprobaciones necesarias en el "validate"
<b>Update.hack</b>	Hack(url y valores)	Se prueba que: <ul style="list-style-type: none"> <li>- Miembros con otro rol no pueden actualizar o acceder al formulario de actualización.</li> <li>- Miembros con el mismo rol no pueden actualizar o acceder al formulario de actualización de claims de otro miembro.</li> </ul>	No se detectó ninguna incidencia ya que el servicio implementa correctamente el enumerado y se realiza la validación pertinente en el "authorised"

		<ul style="list-style-type: none"> <li>- Claims ya publicadas no pueden ser actualizadas</li> <li>- No permite asignar valores ilegales a los atributos de selección.</li> </ul>	
<b>Publish.safe</b>	Casos positivos y negativos del formulario	Se prueba que todos los campos del formulario estén correctamente validados, rechazando los valores no admitidos por el modelo y aceptando aquellos que si lo son.	No se detectó ninguna incidencia ya que todos los campos están correctamente validados y se realizan las comprobaciones necesarias en el "validate"
<b>Publish.hack</b>	Hack(url y valores)	<p>Se prueba que:</p> <ul style="list-style-type: none"> <li>- Miembros con otro rol no pueden publicar cualquier claim.</li> <li>- Miembros con el mismo rol no pueden publicar claims de otro miembro.</li> </ul>	No se detectó ninguna incidencia ya que el servicio implementa correctamente el enumerado y se realiza la validación pertinente en el "authorised"
<b>Delete.safe</b>	Casos positivos y negativos del formulario	<p>Se prueba que:</p> <ul style="list-style-type: none"> <li>- Un claim se borre correctamente, incluso al meter valores diferentes de los mostrados, bien positivos o negativos.</li> <li>- Al borrar el claim se borran los tracking logs asociados</li> </ul>	No se detectó ninguna incidencia ya que todos los campos están correctamente validados y se realizan las comprobaciones necesarias en el "validate"
<b>Delete.hack</b>	Hack	Se prueba que:	No se detectó ninguna incidencia ya que el servicio implementa

		<ul style="list-style-type: none"> <li>- Miembros con otro rol no pueden borrar ningún claim</li> <li>- Miembros con el mismo rol no pueden borrar claims de otro miembro.</li> <li>- Claims ya publicadas no pueden ser borrados</li> </ul>	correctamente el enumerado y se realiza la validación pertinente en el “authorised”
--	--	--	---

Tabla de los casos de prueba por característica para los servicios de tracking log:

Casos de Prueba	Tipo	Descripción	Eficacia
<b>List.safe</b>	Renderizado	Se prueba que el listado se muestra correctamente los tracking logs asignados a cada claim	No se detectó ninguna incidencia ya que el servicio esta correctamente implementado
<b>List.hack</b>	Hack	Se prueba que: <ul style="list-style-type: none"> <li>- Miembros con otro rol no pueden ver cualquier listado de los tracking logs</li> <li>- Miembros con el mismo rol solo pueden ver el listado de los tracking logs asociados a sus claims</li> </ul>	No se detectó ninguna incidencia ya que el servicio esta correctamente implementado tanto en el “authorised” como en la consulta.
<b>Show.safe</b>	Renderizado	Se prueba que un tracking log se muestre correctamente sin ningún error	No se detectó ninguna incidencia ya que el servicio esta correctamente implementado
<b>Show.hack</b>	Hack	Se prueba que:	No se detectó ninguna incidencia ya que se realizaba la validación pertinente en el “authorised”



		<ul style="list-style-type: none"> <li>- Miembros con otro rol no pueden ver el cualquier tracking log</li> <li>- Miembros con el mismo rol solo pueden ver un tracking logs asociados a sus claims</li> </ul>	
<b>Create.safe</b>	Casos positivos y negativos del formulario	Se prueba que todos los campos del formulario estén correctamente validados, rechazando los valores no admitidos por el modelo y aceptando aquellos que si lo son.	No se detectó ninguna incidencia ya que todos los campos están correctamente validados y se realizan las comprobaciones necesarias en el "validate"
<b>Create.hack</b>	Hack(url y valores)	<p>Se prueba que:</p> <ul style="list-style-type: none"> <li>- Miembros con otro rol no pueden crear o acceder al formulario de creación.</li> <li>- Miembros con el mismo rol no pueden crear ni acceder al formulario de creación para tracking logs que no pertenezcan a sus claims.</li> <li>- No permite asignar valores ilegales a los atributos de selección.</li> </ul>	No se detectó ninguna incidencia ya que el servicio implementa correctamente el enumerado y se realiza la validación pertinente en el "authorised"
<b>Update.safe</b>	Casos positivos y negativos del formulario	Se prueba que todos los campos del formulario estén correctamente validados, rechazando los valores no admitidos por el	No se detectó ninguna incidencia ya que todos los campos están correctamente validados y se realizan las comprobaciones necesarias en el "validate"

		modelo y aceptando aquellos que si lo son.	
<b>Update.hack</b>	Hack(url y valores)	<p>Se prueba que:</p> <ul style="list-style-type: none"> <li>- Miembros con otro rol no pueden actualizar o acceder al formulario de actualización de los tracking logs.</li> <li>- Miembros con el mismo rol no pueden actualizar o acceder al formulario de actualización de tracking logs de un claim que no les pertenezca.</li> <li>- Tracking logs ya publicadas no pueden ser actualizadas</li> <li>- No permite asignar valores ilegales a los atributos de selección.</li> <li>- Los atributos de solo lectura no pueden ser modificados</li> </ul>	No se detectó ninguna incidencia ya que el servicio implementa correctamente el enumerado y se realiza la validaciones pertinentes”
<b>Publish.safe</b>	Casos positivos y negativos del formulario	Se prueba que todos los campos del formulario estén correctamente validados, rechazando los valores no admitidos por el modelo y aceptando aquellos que si lo son.	No se detectó ninguna incidencia ya que todos los campos están correctamente validados y se realizan las comprobaciones necesarias en el “validate”
<b>Publish.hack</b>	Hack(url y valores)	<p>Se prueba que:</p> <ul style="list-style-type: none"> <li>- Miembros con otro rol no pueden publicar cualquier tracking log.</li> </ul>	No se detectó ninguna incidencia ya que el servicio implementa correctamente el enumerado y se realiza la validación

		<ul style="list-style-type: none"> <li>- Miembros con el mismo rol no pueden publicar tracking logs de otro miembro.</li> </ul>	pertinente en el “authorised”
<b>Delete.safe</b>	Casos positivos y negativos del formulario	Se prueba que un tracking log se borre correctamente, incluso al meter valores diferentes de los mostrados, bien positivos o negativos.	No se detectó ninguna incidencia ya que todos los campos están correctamente validados y se realizan las comprobaciones necesarias en el “validate”
<b>Delete.hack</b>	Hack	<p>Se prueba que:</p> <ul style="list-style-type: none"> <li>- Miembros con otro rol no pueden borrar ningún tracking log</li> <li>- Miembros con el mismo rol no pueden borrar tracking log de otro miembro.</li> <li>- Tracking logs ya publicadas no pueden ser borrados</li> </ul>	No se detectó ninguna incidencia ya que el servicio implementa correctamente el enumerado y se realiza la validación pertinente en el “authorised”

## 1.4 Calidad de la cobertura

Ahora procederemos a analizar la calidad de la cobertura del código de los servicios tanto de Claim como de Tracking log:

### 1.4.1. Claim

La cobertura obtenida para el paquete assistance-agent.claim es del **99,3%**, lo da señal de una implementación muy robusta de los casos de prueba. Esta cifra sugiere que la gran mayoría de los caminos de ejecución han sido ejecutados, garantizando un nivel elevado de confiabilidad.

En particular, los servicios de **AssistanceAgentClaimListOngoingService**, **AssistanceAgentClaimListService** y **AssistanceAgentClaimShowService** alcanzan una cobertura del **100%**, asegurando así la validación completa de sus funcionalidades.

Respecto a los servicios de **crear, actualizar, eliminar y publicar**, los porcentajes de cobertura se sitúan entre el **98,6%** y el **99,4%**, destacando **AssistanceAgentClaimCreateService** con **99,4%**. Estos niveles de cobertura una muy baja presencia de rutas no cubiertas o condicionales no evaluadas.

Como ocurre en otros módulos, se han implementado validaciones adicionales en métodos como `authorise` y `validate`, lo cual, puede limitar la cobertura funcional obtenida a través de pruebas desde la interfaz. Dichas validaciones, al estar orientadas a reforzar la seguridad, podrían requerir técnicas específicas para lograr una cobertura completa.

### 1.4.2 Tracking Log

La cobertura obtenida para el paquete `assistance-agent.trackingLog` es del **98,1%**, lo que refleja una implementación muy sólida y bien verificada de los casos de prueba. Esta cifra indica que la mayoría de los caminos de ejecución han sido evaluados, asegurando un comportamiento confiable del sistema.

En particular, los servicios de **AssistanceAgentTrackingLogListService**, **AssistanceAgentTrackingLogShowService** y **AssistanceAgentTrackingLogUpdateService** alcanzan una cobertura del **100%**, validando completamente sus funcionalidades.

En cuanto a los servicios de **crear, eliminar y publicar**, los niveles de cobertura oscilan entre el **95,3%** y el **98,4%**, destacando **AssistanceAgentTrackingLogDeleteService** con **98,4%**

Por otro lado, **AssistanceAgentTrackingLogPublishService** presenta una cobertura del **95,3%**, lo que podría indicar la existencia de algunas rutas no cubiertas dentro del flujo de ejecución. Esto debido a lo que ya hemos comentado anteriormente y teniendo en cuenta además que ésta posee muchas validaciones a nivel de lógica de negocio que en algunos casos es difícil de probar en su totalidad.

## 1.5 Conclusiones

En definitiva, se han desarrollado un conjunto de pruebas que cubre la gran mayoría de líneas de código de las entidades del estudiante 4. Aunque haya métodos de validación que se pueden optimizar, se ha cumplido el objetivo de obtener un sistema robusto y seguro ante la gran mayoría de escenarios que se dan en el mundo real.

## 2. Testing de Rendimiento

### 2.1 Entorno y protocolo

Para el desarrollo de esta sección se va a emplear un equipo con las siguientes características:

Modelo	Ram	CPU	GPU	Disco	SO
<b>HP Pavilion 15-cs0008ns</b>	16 GB	Intel Core i7-8550u 1.80Hz	Nvidia MX150 2GB	1.82 TB	Windows 11

Para este estudio se va a emplear únicamente los casos de prueba que he ejecutado individualmente como estudiante 4, empleando la versión 25.5.0 del proyecto y del framework. En el primer lanzamiento, no se emplearán índices en mis entidades, para ver si el rendimiento mejora una vez se implementen.

### 2.2 Recogida de datos

Para la obtención de datos, tanto antes como después de implementar índices, se han seguido los pasos explicados en la teoría, analizando la información del “.trace” generado por la aplicación y filtrándola para obtener un fichero limpio del que poder obtener información útil como gráficos.

## 2.3 Estadísticas Descriptivas

Una vez filtradas las peticiones realizadas en los casos de prueba eliminando las irrelevantes, podemos afirmar que, para el estudio, se utiliza un total de 679 filas de datos. En ambos casos se presenta una media y desviación similares, de los que se pueden derivar, además, los siguientes datos:

Datos obtenidos de la primera traza no mejorada:

<i>Estadísticas Descriptiva</i>	
Media	11,53168868
Error típico	0,65900389
Mediana	6,99995
Moda	1,612
Desviación estándar	17,18471898
Varianza de la muestra	295,3145663
Curtosis	32,41293993
Coefficiente de asimetría	4,426255295
Rango	190,878
Mínimo	1,3468
Máximo	192,2248
Suma	7841,5483
Cuenta	680
Nivel de confianza(95,0%)	1,293930339

Interval (ms)	10,23775834	12,82561902
Interval (s)	0,010237758	0,012825619

Datos obtenidos de la segunda traza mejorada:

<i>Estadísticas Descriptiva</i>	
Media	12,52918691
Error típico	0,690156165
Mediana	7,2496
Moda	2,0263
Desviación estándar	17,99707092
Varianza de la muestra	323,8945617
Curtosis	18,39371513
Coefficiente de asimetría	3,475605701
Rango	167,1379
Mínimo	1,3548
Máximo	168,4927
Suma	8519,8471
Cuenta	680
Nivel de confianza(95,0%)	1,355096706

Interval (ms)	11,17409021	13,88428362
Interval (s)	0,01117409	0,013884284

Como podemos observar, las traza no mejorada y mejorada no introduce cambios significativos en cuanto a los datos del rendimiento general. Es más, incluso se puede apreciar una pequeña disminución del rendimiento aunque esto puede no ser un problema y ser coherente con respecto a como funciona los motores de base de datos que se verá a continuación.

Podemos decir que ambas trazas cumplen con creces el requisito del rendimiento ya que los límites superiores tanto 12,8 ms y 13,88 ms son ambos considerablemente menores que 1 segundo (1000ms) nuestro requisito base.

Por tanto, una de las conclusiones que podemos sacar es que este equipo ejecuta los casos de prueba de una forma bastante rápida. Así, aunque implementemos pequeñas mejoras como los índices, verdaderamente no se va a notar en el tiempo empleado, puesto que de por sí el tiempo base es muy pequeño. O incluso como en este caso puede aumentar el tiempo general.

## 2.4 Gráficos de eficiencia por característica

Uno de los aspectos que más nos interesa del rendimiento, es comprobar cuál es la petición más ineficiente (MIR), y ver si los índices suponen una mejora real en el rendimiento de la misma.

Grafico de eficiencia de la traza sin mejorar:

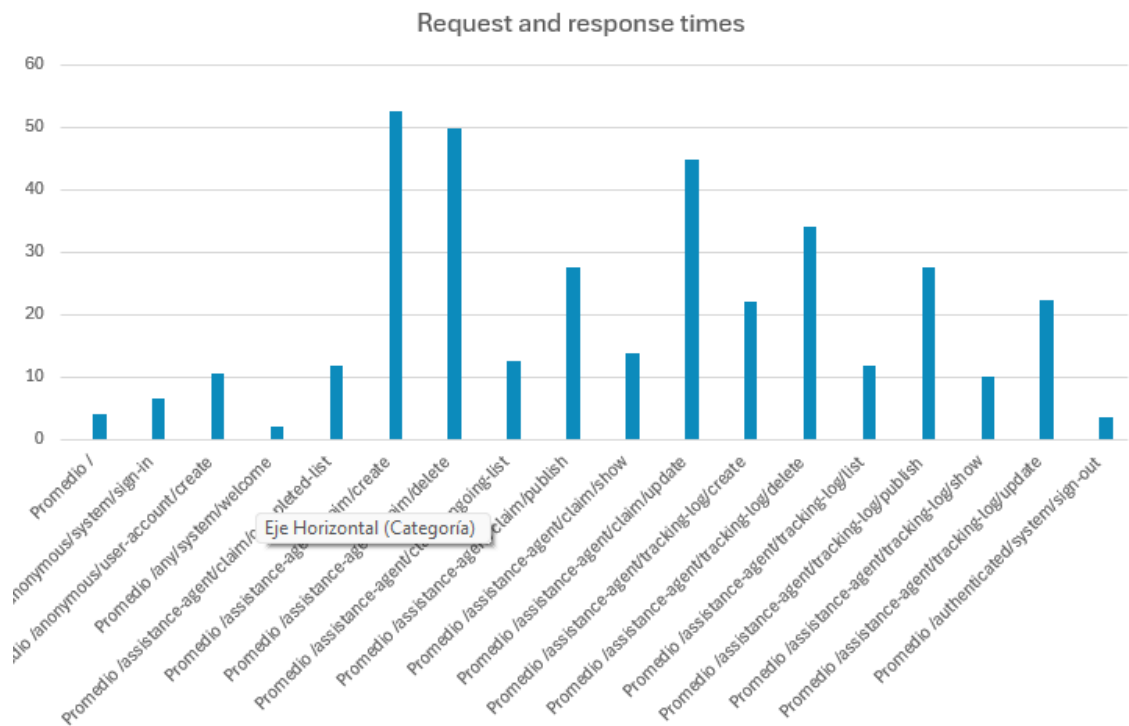
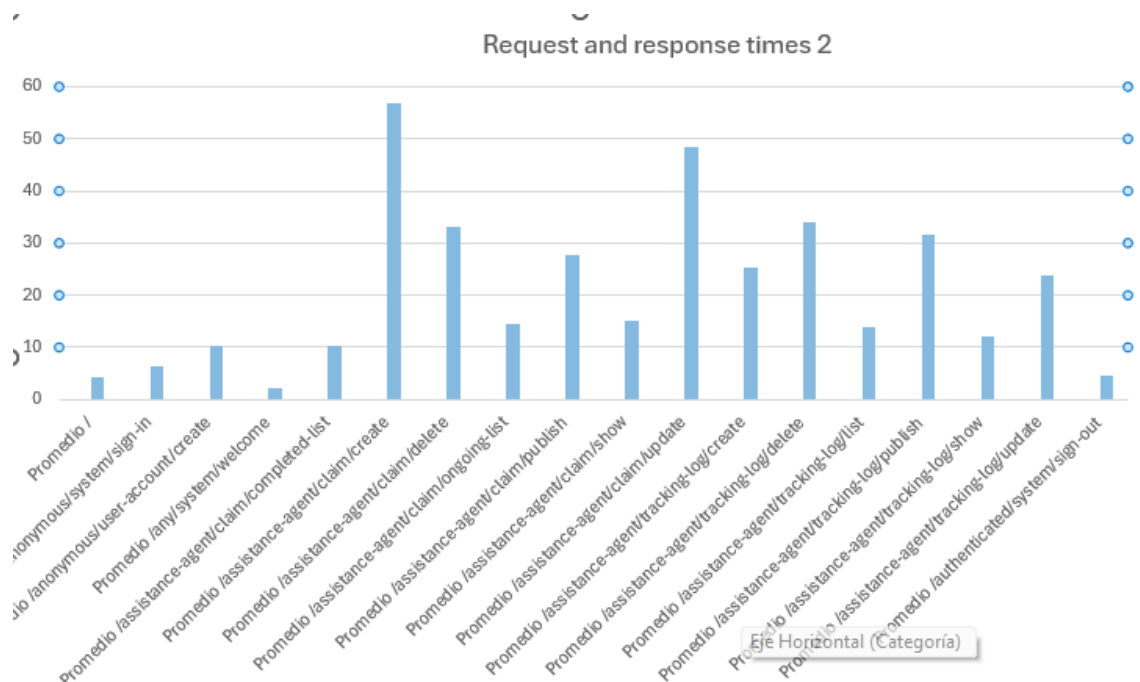




Grafico de eficiencia de la traza sin mejorar:



Podemos observar que hay pequeñas variaciones en los tiempos promedios entre las trazas mejoras y las no mejoradas, siendo estas primeras un ínfimamente mayores.

Sin embargo, si nos fijamos detenidamente, vemos que hay un rendimiento en específico que mejora en gran medida pasando de ser uno de los más ineficientes a estar en un rendimiento promedio.

Aunque el tiempo medio total ha aumentado ligeramente, el objetivo principal de la optimización **mejorar las consultas más ineficientes (MIRs)** se ha cumplido con éxito. Estas mejoras específicas son mucho más importantes para la experiencia del usuario y la escalabilidad que una simple reducción del promedio general.

Así, si bien no era necesario analizar el MIR porque el límite superior de los intervalos de confianza estaba muy por debajo del requisito impuesto, nos ha ayudado a perfilar un poco más las diferencias entre ambas trazas y a entender un poco mejor dónde el proyecto tiene más carga.

## 2.5 hipótesis y conclusión

Para confirmar si las diferencias observadas son estadísticamente significativas, se ha realizado una prueba Z para dos muestras independientes con varianzas conocidas:

	<i>Before</i>	<i>After</i>
Media	11,34968851	12,36668866
Varianza (conocida)	295,3145663	323,8945617
Observaciones	679	679
Diferencia hipotética de las medias	0	
z	-1,06496964	
P(Z<=z) una cola	0,143444834	
Valor crítico de z (una cola)	1,644853627	
Valor crítico de z (dos colas)	0,286889668	
Valor crítico de z (dos colas)	1,959963985	

El valor p es claramente mayor que  $\alpha$ , por lo que no podemos rechazar la hipótesis nula. En consecuencia, no hay evidencia estadística para afirmar que los índices han mejorado el rendimiento de forma significativa.

## 3. Tabla de Revisión

Versión	Fecha	Descripción de los cambios
1.0	26/05/2025	Creación inicial del documento