ACME AirNav Solutions



TESTING REPORT – Student 4

Grupo: C1.050

Miembros: Cristina Fernández Chica ([criferchi@alum.us.es](mailto:criferchi@alum.us.es)), Ángel Amo Sánchez ([angamosan@alum.us.es](mailto:angamosan@alum.us.es)), Candela Jazmín Gutiérrez González ([cangutgon@alum.us.es](mailto:cangutgon@alum.us.es)), Marta Aguilar Morcillo ([maragumor@alum.us.es](mailto:maragumor@alum.us.es)) y Luis Emmanuel Chávez Malavé ([luichamal@alum.us.es](mailto:luichamal@alum.us.es))

Repositorio: <https://github.com/Cristinafernandezchica/Acme-ANS>

Planning dashboard: <https://github.com/users/Cristinafernandezchica/projects/1/views/1>

Sevilla 26 Mayo, 2025

**TABLA DE CONTENIDOS**

[**Resumen Ejecutivo 3**](#_uqzfrhqx4u4)

[**Tabla de Revisiones 3**](#_vnwg21y46q5z)

[**Introducción 4**](#_d71yrsf36elu)

[**Contenido 4**](#_h6693km0wbu)

[Pruebas funcionales 4](#_olrzoglvb38f)

Recubrimiento del código 8

Análisis de desempeño 8

**Gráficas 10**

**Comparación de rendimiento con otro ordenador 12**

[**Conclusión**](#_qla8o5xpmqnh) **13**

[**Bibliografía**](#_o5dkbw1z0jts) **13**

#### **Resumen Ejecutivo**

Este documento ofrece una descripción detallada de los diferentes procedimientos aplicados en el contexto del testing formal del proyecto Acme-ANS. Abarca desde la creación de una cantidad adecuada de datos de prueba hasta la verificación minuciosa del cumplimiento de los requisitos funcionales. Además, se explican las herramientas estadísticas empleadas para analizar los tiempos de respuesta entre distintas peticiones, evaluar el rendimiento comparativo entre dos equipos y observar las mejoras en un mismo equipo tras la incorporación de índices en la base de datos.

En definitiva, se ha adoptado una metodología rigurosa para identificar y corregir los errores detectados durante las pruebas, con el objetivo de garantizar un producto final de alta calidad que cumpla con las expectativas del cliente.

#### **Tabla de Revisiones**

|  |  |  |  |
| --- | --- | --- | --- |
| **Número de revisión** | **Fecha** | **Descripción de revisión** | **Autor** |
| 1.0 | 26/05/2025 | Realización de todos los apartados del reporte | Marta Aguilar Morcillo |
|  |  |  |  |

#### **Introducción**

Este informe tiene como propósito detallar el proceso seguido para llevar a cabo el testing funcional y el análisis de rendimiento en el marco del proyecto. En particular, se centra en la verificación de los requisitos funcionales 8 y 9 correspondientes al Student Nº4, siendo estos las operaciones sobre registros claims y tracking logs.

La estructura del documento responde a los lineamientos establecidos en el anexo compartido a través de Enseñanza Virtual. Comienza con una portada que incluye la información del autor, seguida de una tabla de control de versiones donde se documentan las actualizaciones del informe, indicando el número de versión y la fecha de cada modificación. A continuación, se presenta un resumen ejecutivo que proporciona una visión general del contenido, acompañado de una introducción en la que se anticipan los temas tratados y se explica cómo se ha organizado el texto.

El cuerpo principal del informe se divide en dos secciones fundamentales, cada una dedicada a una fase específica del testing. La primera sección aborda las pruebas funcionales, mostrando los distintos casos de prueba agrupados por funcionalidad. De cada uno se ofrece una breve explicación junto con una evaluación de su capacidad para detectar fallos. La segunda sección está dedicada a las pruebas de rendimiento, donde se analizan los tiempos de respuesta mediante representaciones gráficas y se calcula un intervalo de confianza del 95% para dichos tiempos.

Esta parte también incluye una comparación entre dos escenarios —antes y después de aplicar índices a la base de datos— mediante un contraste de hipótesis (Z-test) que permite determinar si la mejora observada es estadísticamente significativa. Además del análisis de las pruebas en dos ordenadores diferentes y ver la diferencia de rendimientos.

#### **Contenido**

##### **Pruebas funcionales**

Para cada servicio se han realizado tanto pruebas positivas (.safe) como de hacking (.hack) de la siguiente forma:

**/assistanceAgent/claim/…**

| **Tipo** | **Acción** | **Descripción** | **Bugs Detectados** |
| --- | --- | --- | --- |
| List-mine.safe | list | Se listaron todos los reclamos, los claims, de los clientes. Se probaron listados sin claims, listados con un solo claim y listados con varios. | No se detectaron bugs. |
| Show- mine.safe | show | Se mostraron cada uno de los reclamos, los claims, de cada cliente individualmente tanto publicados como no publicados, con registros de seguimiento (tracking logs) y sin ellos. | No se detectaron bugs. |
| Create- mine.safe | create | Se probó cada variación posible de los campos del formulario de creación de reclamos (claims), formularios vacíos y validaciones. | No se detectaron bugs. |
| Update- mine.safe | update | Se verificaron variaciones de los campos del formulario de actualización de las reclamos (claims), el formulario vacío y posibles validaciones. | No se detectaron bugs. |
| Publish- mine.safe | publish | Se verificaron variaciones de los campos del formulario de publicación de las reclamos (claims), el formulario vacío y posibles validaciones. | No se detectaron bugs. |
| Delete-mine.safe | delete | Se probó borrar un reclamo (un claim) no publicado. | No se detectaron bugs. |
| List-mine.hack | list | Se intentó acceder al listado de reclamos (claims) sin rol o con un rol incorrecto. | No se detectaron bugs. |
| Show- mine.hack | show | Se intentó ver reclamos (claims) con un rol correcto pero usuario erróneo o sin rol, ver reclamos con un id incorrecto o inexistente o ver reclamos sin especificar ningún id. | No se detectaron bugs. |
| Create- mine.hack | create | Se intentó acceder al formulario de creación de reclamos (claims) desde roles incorrectos o sin roles; o crear un reclamo con un tipo de reclamo (un claim type) erróneo, una pata (una leg) errónea, o crear un reclamo especificando un parámetro id. | He detectado y solucionado el no poder acceder a la URL de create con id, actuando como update. |
| Update- mine.hack | update | Se intentó acceder al formulario de actualización de reclamos (claims) usando el id de un reclamo publicado, con un id inexistente o sin id. Además se intentó acceder al formulario de actualización de un reclamo concreto con el rol de otro cliente con el mismo rol pero distinto usuario, desde roles incorrectos o sin roles. También se intentó actualizar campos de solo lectura, actualizar un reclamo con un tipo de reclamo (claim type). .incorrecto o con una pata (una leg) errónea, es decir, una pata que no se encuentra entre las opciones. | He detectado y solucionado el no poder acceder a la URL de update con id nulo. |
| Publish- mine.hack | publish | Se intentó acceder al formulario de publicación de reclamos (claims) usando el id de un reclamo publicado, con un id inexistente o sin id. Además se intentó acceder al formulario de publicación de un reclamo concreto con el rol de otro cliente con el mismo rol pero distinto usuario, desde roles incorrectos o sin roles. También se intentó actualizar campos de solo lectura al publicar, publicar un reclamo con un tipo de reclamo (claim type) incorrecto o con una pata (una leg) errónea, es decir, una pata que no se encuentra entre las opciones. | He detectado y solucionado el no poder acceder a la URL de publish con id nulo. |
| Delete-mine.hack | delete | Se intentó acceder al formulario de borrado de reclamos (claims) usando el id de un reclamo publicado, con un id inexistente o sin id. Además se intentó acceder al formulario de borrado de un reclamo concreto con el rol de otro cliente con el mismo rol pero distinto usuario, desde roles incorrectos o sin roles. | No se detectaron bugs. |

**/assistanceAgent /trackingLog /…**

| **Tipo** | **Acción** | **Descripción** | **Bugs Detectados** |
| --- | --- | --- | --- |
| List-mine.safe | list | Se listaron todos los registros de seguimiento, los tracking logs, de todos los clientes. Se probaron listados sin registros de seguimiento, listados con un solo registro de seguimiento y listados con varios. | No se detectaron bugs. |
| Show- mine.safe | show | Se mostraron cada uno de los registros de seguimiento, los tracking logs, de cada cliente individualmente tanto publicados como no publicados. | No se detectaron bugs. |
| Create- mine.safe | create | Se probó cada variación posible de los campos del formulario de creación de registros de seguimiento (tracking logs), formularios vacíos y todas las posibles validaciones. | No se detectaron bugs. |
| Update- mine.safe | update | Se verificaron variaciones de los campos del formulario de actualización de los registros de seguimiento (tracking logs), el formulario vacío y todas las posibles validaciones. | No se detectaron bugs. |
| Publish- mine.safe | publish | Se verificaron variaciones de los campos del formulario de publicación de los registros de seguimiento (tracking logs), el formulario vacío y todas las posibles validaciones. | No se detectaron bugs. |
| Delete-mine.safe | delete | Se probó borrar un registro de seguimiento no publicado (un tracking log). | No se detectaron bugs. |
| List-mine.hack | list | Se intentó acceder con un rol correcto pero usuario erróneo al listado de registros de seguimiento (de tracking logs) de un reclamo (un claim) que no pertenece a dicho usuario, también se intentó acceder una lista de registros de seguimiento (de tracking logs) sin rol, usando un claim\_id inexistente, nulo o el claim\_id de un reclamo que no esté publicado. | No se detectaron bugs. |
| Show- mine.hack | show | Se intentó ver registros de seguimiento (tracking logs) con un rol correcto pero usuario erróneo o sin rol o ver registros de seguimiento con un id nulo o inexistente. | No se detectaron bugs. |
| Create- mine.hack | create | Se intentó acceder al formulario de creación de registros de seguimiento (tracking logs) desde roles incorrectos o sin roles; o crear un registro de seguimiento con un estado (un status) erróneo, o especificando un parámetro id. Además también se ha intentado acceder al formulario de creación de registros de seguimiento con un claim\_id de un claim no publicado, un claim\_id nulo o un claim\_id inexistente. | He detectado y solucionado el no poder acceder a la URL de create con id, actuando como update. |
| Update- mine.hack | update | Se intentó acceder al formulario de actualización de registros de seguimiento (de tracking logs) desde roles incorrectos o sin roles o con rol correcto pero usando un usuario que no es dueño del reclamo al que pertenece el registro de seguimiento. También se ha intentado cambiar campos de solo lectura o actualizar un registro de seguimiento con un estado (status) incorrecto. A su vez, se ha intentado acceder al formulario de actualización de un registro de seguimiento ya publicado, usando un id nulo, usando un id inexistente, o sin especificar el registro de seguimiento a actualizar. | He detectado y solucionado el no poder acceder a la URL de update con id. |
| Publish- mine.hack | publish | Se intentó acceder al formulario de publicación de registros de seguimiento (de tracking logs) desde roles incorrectos o sin roles o con rol correcto pero usando un usuario que no es dueño del reclamo al que pertenece el registro de seguimiento. También se ha intentado cambiar campos de solo lectura o publicar un registro de seguimiento con un estado (status) incorrecto. A su vez, se ha intentado acceder al formulario de publicación de un registro de seguimiento ya publicado, usando un id nulo, usando un id inexistente, o sin especificar el registro de seguimiento a publicar. | He detectado y solucionado el no poder acceder a la URL de publish con id. |
| Delete-mine.hack | delete | Se intentó acceder al formulario de borrado de registros de seguimiento (de tracking logs) desde roles incorrectos o sin roles o con rol correcto pero usando un usuario que no es dueño del reclamo al que pertenece el registro de seguimiento. A su vez, se ha intentado acceder al formulario de borrado de un registro de seguimiento ya publicado, usando un id nulo, usando un id inexistente, o sin especificar el registro de seguimiento a borrar. | No se detectaron bugs. |

##### **Recubrimiento del código**

Las pruebas son bastante fiables al contar con un porcentaje de cobertura superior al 85%, pues tenemos que tener en cuenta las instrucciones que no se realizarán o únicamente en casos extremos (como en el ShowService de Claim que se añade en el unbind como se mandaría un select vacío en el caso de que no hubiera legs posibles, pero realmente ese caso nunca se puede dar porque no se permite crear claims si no hay legs posibles. También es el caso de los unbind en los DeleteServies)

Gráfico, Gráfico de barras

El contenido generado por IA puede ser incorrecto.

Gráfico, Gráfico de barras

El contenido generado por IA puede ser incorrecto.

#### **Análisis de desempeño**

En primer lugar, analizamos las estadísticas sobre el rendimiento de la aplicación antes y después de los índices, así como sus gráficas de tiempo. Para poder comparar estos tiempos, se ha realizado el Z-test y lo compararemos con las estadísticas descriptivas para poder llegar a una conclusión y comprobar si los cambios tienen significancia estadística.

|  |  |  |
| --- | --- | --- |
| Prueba z para medias de dos muestras | | |
| |  |  |  | | --- | --- | --- | |  |  |  | |  | *Before* | *After* | | Media | 14,4580055 | 18,1899032 | | Varianza (conocida) | 302,24322 | 470,238031 | | Observaciones | 842 | 842 | | Diferencia hipotética de las medias | 0 |  | | z | -3,896205 |  | | P(Z<=z) una cola | 4,8856E-05 |  | | Valor crítico de z (una cola) | 1,64485363 |  | | Valor crítico de z (dos colas) | 9,7712E-05 |  | | Valor crítico de z (dos colas) | 1,95996398 |  | |  |  |

Como se puede observar, el p-valor obtenido en la prueba Z es 0.0000977. Considerando un nivel de confianza del 95% (α = 0.05), y dado que el p-valor es mucho menor que α, el resultado es estadísticamente significativo. Según la teoría estadística, esto permite comparar los promedios de los tiempos antes y después de los cambios realizados.

En este caso, como el promedio posterior a los cambios es mayor, se puede concluir que el rendimiento ha empeorado. No obstante, este empeoramiento no se debe necesariamente a una degradación del código, sino a una diferencia en la forma en que se obtuvieron los datos. El archivo tester.trace utilizado para los datos "antes de los cambios" fue generado únicamente con las clases y pruebas que afectan a mi *student* en mi rama. En cambio, el archivo tester.trace correspondiente a los datos "después de los cambios" fue generado desde la rama *main*, que incluye el código y las pruebas de todos los *students* del equipo.

A continuación, se puede observar que la diferencia entre los datos antes y después de los cambios no es especialmente significativa.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| *Before* | | | |  | | |  | *After* | | | | | |  | | |
|  | |  | |  | | |  |  | | |  | | |  | | |
| Media | | 23,4129437 | |  | | |  | Media | | 29,4799386 | | |  | | |
| Error típico | | 0,81863444 | |  | | |  | Error típico | | 1,02110547 | | |  | | |
| Mediana | | 20,071899 | |  | | |  | Mediana | | 25,169 | | |  | | |
| Moda | | #N/D | |  | | |  | Moda | | #N/D | | |  | | |
| Desviación estándar | | 17,3851437 | |  | | |  | Desviación estándar | | 21,6849725 | | |  | | |
| Varianza de la muestra | | 302,24322 | |  | | |  | Varianza de la muestra | | 470,238031 | | |  | | |
| Curtosis | | 17,8781866 | |  | | |  | Curtosis | | 14,5893557 | | |  | | |
| Coeficiente de asimetría | | 3,12015195 | |  | | |  | Coeficiente de asimetría | | 2,83491774 | | |  | | |
| Rango | | 169,5927 | |  | | |  | Rango | | 199,0844 | | |  | | |
| Mínimo | | 3,6426 | |  | | |  | Mínimo | | 4,6364 | | |  | | |
| Máximo | | 173,2353 | |  | | |  | Máximo | | 203,7208 | | |  | | |
| Suma | | 10559,2376 | |  | | |  | Suma | | 13295,4523 | | |  | | |
| Cuenta | | 451 | |  | | |  | Cuenta | | 451 | | |  | | |
| Nivel de confianza(95,0%) | | 1,60882106 | |  | | |  | Nivel de confianza(95,0%) | | 2,00672719 | | |  | | |
|  | |  | |  | | |  |  | |  | | |  | | |
| Interval (ms) | 21,8041226 | | 25,0217647 | |  | Interval (ms) | | | 27,4732114 | | | 31,4866658 | | |
| Interval (s) | 0,02180412 | | 0,02502176 | |  | Interval (s) | | | 0,02747321 | | | 0,03148667 | | |

Como se puede apreciar, aunque el tiempo de resolución de peticiones ha aumentado, sigue siendo considerablemente bajo, con un valor de 31,49 ms.

#### **Gráficas**

Tras el análisis podremos observar cómo han quedado finalmente las gráficas junto con un análisis de los MIR que no han sido posible de reducir:

A continuación se muestran los resultados del perfilador de VisualVM:  
Texto

El contenido generado por IA puede ser incorrecto.

Interfaz de usuario gráfica, Texto, Aplicación

El contenido generado por IA puede ser incorrecto.

Tal y como se muestra en la gráfica, las solicitudes que mayor tiempo de procesamiento requieren son aquellas gestionadas por la clase CreateService de *Claim*. Este comportamiento se debe a que dicha clase realiza múltiples llamadas a la base de datos, principalmente para recuperar las *legs* asociadas a cada *claim*. Estas consultas se llevan a cabo tanto en el método unbind, con el objetivo de mostrarlas en un *select* dentro del formulario, como en el método authorise, para garantizar que no se asocie una *leg* incorrecta a un \*claim`.

Además, existen otras llamadas a la base de datos, como las que se realizan durante el bind, donde se toma el leg\_id enviado desde el formulario para recuperar la instancia correspondiente de *leg*. Todo este conjunto de operaciones incrementa notablemente el tiempo total de ejecución de estas solicitudes.

Por otro lado, los resultados obtenidos mediante el perfilador *VisualVM* indican que el método con mayor tiempo total de ejecución es load, perteneciente al servicio que gestiona la lista de *claims* en curso. Sin embargo, dado que su tiempo propio es de cero milisegundos, se concluye que el consumo de tiempo no se origina en el método en sí, sino en las diversas llamadas internas que este realiza.

#### **Comparación de rendimiento con otro ordenador**

Finalmente realizaremos una comparativa entre dos ordenadores distintos, pudiendo demostrar como el hardware también influye a la hora de responder peticiones. La comparativa es únicamente tras la optimización de los índices.

El PC\_1 es el mismo utilizado para el reporte, un portátil HP Pavilion sin tarjeta gráfica dedicada, y para el PC\_2 se ha utilizado un portátil HP Victus 16-d1001ns con procesador Intel Core i7 12700H y tarjeta gráfica dedicada (GeForce RTX 3050 Ti) de una de mis compañeras del grupo.

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | *PC\_1* | | | | | |  | | |
|  |  | | |  | | |  | | |
|  | Media | | 29,4799386 | | |  | | |
|  | Error típico | | 1,02110547 | | |  | | |
|  | Mediana | | 25,169 | | |  | | |
|  | Moda | | #N/D | | |  | | |
|  | Desviación estándar | | 21,6849725 | | |  | | |
|  | Varianza de la muestra | | 470,238031 | | |  | | |
|  | Curtosis | | 14,5893557 | | |  | | |
|  | Coeficiente de asimetría | | 2,83491774 | | |  | | |
|  | Rango | | 199,0844 | | |  | | |
|  | Mínimo | | 4,6364 | | |  | | |
|  | Máximo | | 203,7208 | | |  | | |
|  | Suma | | 13295,4523 | | |  | | |
|  | Cuenta | | 451 | | |  | | |
|  | Nivel de confianza(95,0%) | | 2,00672719 | | |  | | |
|  |  | |  | | |  | | |
| Interval (ms) | | 27,4732114 | | | 31,4866658 | | |
| Interval (s) | | 0,02747321 | | | 0,03148667 | | |

|  |  |
| --- | --- |
| *PC\_2* | |
|  |  |
| Media | 12,6886918 |
| Error típico | 0,49305021 |
| Mediana | 10,7922 |
| Moda | #N/D |
| Desviación estándar | 10,4707893 |
| Varianza de la muestra | 109,63743 |
| Curtosis | 20,2888474 |
| Coeficiente de asimetría | 3,66647015 |
| Rango | 87,9854 |
| Mínimo | 2,3992 |
| Máximo | 90,3846 |
| Suma | 5722,6 |
| Cuenta | 451 |
| Nivel de confianza(95,0%) | 0,96896677 |
|  |  |

|  |  |  |
| --- | --- | --- |
| Interval(ms) | 11,719725 | 13,6576586 |
| Interval(s) | 0,01171973 | 0,01365766 |

Podemos observar como el PC\_2, el intervalo de confianza con una cota superior de 13,66 ms, frente a la cuota de 31,49ms del PC\_1. Ante esto podemos concluir con que el PC\_2 tiene un mejor desempeño con respecto a tiempos de respuesta, siendo notable que es menor que la del PC\_1.

#### **Conclusión**

En este informe se analizamos el proceso de testing formal tras completar el proyecto y con el objetivo de encontrar todos los errores posibles y así corregirlos. Las pruebas realizadas indican que se han probado múltiples escenarios posibles y nos permiten documentar ciertos bugs y fallos del sistema para los requisitos 8 y 9 del Student 4, pudiendo reducir la posibilidad de fallo del sistema al mínimo y a la misma vez mejorar el rendimiento del mismo.

Por otra parte, el análisis de rendimiento ha descubierto, mediante un Z-test con

significancia estadística del 95%, que las optimizaciones aplicadas para una mayor rapidez

en la búsqueda en base de datos, no han sido suficientes para mejorar el tiempo de

respuesta del sistema, lo que nos ha llevado a identificar ciertos problemas con algunos

métodos, que a pesar de ser revisados exhaustivamente e incluso refactorizados, no se ha

tenido éxito en la mejora del rendimiento.

En conjunto, los resultados en base a los test realizados demuestran que es un sistema con una calidad funcional bastante robusta y nos asegura el cumplimiento de los estándares de calidad del proyecto, pero con un rendimiento que puede llegar a dar ciertos problemas en algunos dispositivos con sistemas menos avanzados como puede ser este mismo.

De esto sacamos que debemos revisar la eficiencia desde un inicio a la hora de implementar un proyecto, pues a la hora de querer refactorizar puede dar bastantes problemas y resultar en un sistema demasiado lento. Debemos seguir intentando de todas las formas el limitar llamadas o la reducción de código demasiado largo o inservible.

#### **Bibliografía**

Enseñanza virtual: https://ev.us.es