# Testing report

Grupo C2.054 | Diseño y Pruebas II | 30/05/2025

Versiones

|  |  |  |
| --- | --- | --- |
| Fecha | Versión | Descripción |
| 26/05/2025 | 1.0 | Creación plantilla |
| 26/05/2025 | 2.0 | Entrega primera convocatoria |
| 30/06/2025 | 3.0 | Entrega segunda convocatoria |

Miembros:

Esteban López Pérez (estlopper@alum.us.es)

Jesús Pons Morís (jesponmor@alum.us.es)

Kevin Amador Calzadilla (kevamacal@alum.us.es)

Manuel Zurita Fernández (manzurfer@alum.us.es)

Repositorio: <https://github.com/DP2-C1-054/Acme-ANS-C2>

# Índice:

Resumen ejecutivo………………………………………………………………………3

Introducción……………………………………………………………………………..3

Contenido………………………………………………………………………………..4

Pruebas funcionales……………………………………………………………...4

Análisis de rendimiento……………………………………………………….....7

Conclusiones………………………………………………………………………….....9

Comparativa…………………………………………………………………………....10

Bibliografía…………………………………………………………………………….10

# Resumen ejecutivo

El presente informe expone las pruebas funcionales y de rendimiento realizadas sobre el trabajo de mi estudiante. El objetivo principal fue verificar que todas las funcionalidades se comportaran conforme a lo esperado, así como evaluar la capacidad de respuesta del sistema bajo condiciones normales de uso.

Para las pruebas funcionales, los casos de prueba se estructuraron según las distintas funcionalidades del sistema. Cada caso se diseñó para validar un aspecto específico, lo que permitió confirmar el correcto funcionamiento de la aplicación.

En cuanto a las pruebas de rendimiento, se siguió la metodología descrita en la guía de la sesión. Se recopilaron los tiempos de ejecución a partir de los archivos .trace, los cuales fueron analizados mediante Microsoft Excel. Se elaboraron gráficos y se calcularon intervalos de confianza del 95 % con el fin de determinar si los tiempos de respuesta del sistema se mantenían dentro de los límites aceptables. Las pruebas se llevaron a cabo en dos configuraciones distintas: una utilizando la base de datos sin índices adicionales, y otra con índices relevantes aplicados. Posteriormente, se realizó una comparación estadística entre ambas configuraciones para evaluar el impacto del uso de índices en el rendimiento.

En resumen, este informe documenta las pruebas realizadas sobre las principales funcionalidades de la aplicación, complementadas con datos de rendimiento que ofrecen una visión clara del comportamiento del sistema en condiciones reales de operación.

# Introducción

El propósito de este informe es documentar las pruebas realizadas sobre mi proyecto, abarcando tanto los aspectos funcionales como de rendimiento. El objetivo principal fue asegurar que cada funcionalidad operara correctamente y que el sistema fuera capaz de gestionar solicitudes de manera eficiente.

Para ello, en primer lugar, ejecuté un conjunto de pruebas funcionales organizadas por características, con el fin de verificar que la aplicación respondiera conforme a lo esperado. Posteriormente, me centré en las pruebas de rendimiento, analizando los tiempos de respuesta del sistema ante distintas solicitudes. Este análisis permitió identificar posibles cuellos de botella y comparar el rendimiento en diferentes configuraciones.

En conjunto, esta fase de pruebas me proporcionó una visión más clara del comportamiento del sistema en condiciones similares a las de un entorno real.

# Contenido

## Pruebas funcionales

En esta fase de pruebas, he evaluado las funcionalidades de mi proyecto Acme-ANS, con especial énfasis en lo relacionado con las asignaciones de vuelo y los registros de actividad. Las pruebas se encuentran documentadas mediante archivos de seguimiento (trace files) ubicados en los directorios /src/test/resources/flight-crew-member/flight-assignment y /src/test/resources/flight-crew-member/flight-assignment.

Todos los casos de prueba alcanzaron una cobertura del 99 %, como se muestra en la imagen incluida a continuación.

A screenshot of a computer

AI-generated content may be incorrect.

Para conseguir la cobertura mostrada previamente, he tenido no solo que probar los casos en que las funcionalidades son utilizadas correctamente, sino también en los que un usuario puede abusar de la aplicación para realizar acciones no autorizadas.

A continuación, se muestran los tests .safe y .hack realizados para ello:

A screenshot of a computer

AI-generated content may be incorrect. A screenshot of a computer

AI-generated content may be incorrect.

En los tests .safe se reproducen todos los casos en los que un usuario sin intenciones malévolas es capaz de usar las funcionalidades ofrecidas. Por otro lado, en los .hack, se reproducen los casos en los que otros usuarios intentan acceder a funcionalidades de la aplicación de forma no autorizada.

Activity log:

List.safe: Verifica que un miembro de la tripulación es capaz de listar sus registros de actividad. El test asegura que los registros son devueltos correctamente.

Show.safe: Garantiza la correcta visualización del detalle de un registro de actividad específico. Se comprueba que la información presentada corresponde a un registro legítimo del miembro de la tripulación.

Create.safe: Evalúa la creación de nuevos registros de actividad mediante datos válidos. Además, se prueba la validación del sistema intentando crear registros con campos vacíos, cadenas de texto excesivamente largas o valores no válidos, como fechas incorrectas.

Update.safe: Prueba la actualización de registros existentes pertenecientes al miembro de la tripulación, tanto con datos válidos como con entradas inválidas, replicando las condiciones exploradas en create.safe.

Publish.safe: Confirma que un miembro de la tripulación puede publicar sus registros válidos. El sistema impide la publicación de registros incompletos o inválidos, mostrando los errores correspondientes.

Delete.safe: Prueba que un miembro de la tripulación pueda eliminar un registro existente del sistema.

Show.hack: Simula intentos de acceso no autorizado mediante la manipulación de URLs para visualizar registros que no pertenecen al usuario autenticado o que no existen. También se prueban accesos a registros en estado no visible.

Create.hack: Reproduce un mal uso de herramientas de desarrollo del navegador (F12) para modificar el ID o el payload de la solicitud durante la creación, intentando interferir con registros ajenos o duplicar entradas.

Update.hack: Incluye simulaciones de ataques mediante solicitudes POST manipuladas, con el objetivo de actualizar registros no pertenecientes al usuario o que se encuentran bloqueados para edición.

Publish.hack: Simula intentos de publicación no autorizada mediante manipulación de URLs o datos. El sistema bloquea correctamente estos intentos y evita cambios indebidos.

Delete.hack: Confirma que un usuario no autenticado o perteneciente a un realm no autorizado no pueda eliminar un registro de actividad existente. También se comprueba que un usuario del mismo realm no pueda borrar un registro que no es suyo.

Flight assignment:

List.safe: Verifica que el miembro de la tripulación puede visualizar correctamente las listas de asignaciones de vuelo, tanto planificados como completados. Se asegura que solo se muestren asignaciones asociadas al usuario autenticado.

Show.safe: Evalúa la visualización detallada de una asignación de vuelo. Se comprueba que el usuario solo acceda a información relevante de sus propias asignaciones.

Create.safe: Prueba la creación de asignaciones de vuelo con datos válidos. También se incluyen intentos con datos incompletos o inválidos (campos vacíos, valores fuera de rango), asegurando que el sistema responda con validaciones adecuadas.

Update.safe: Verifica que el miembro de la tripulación pueda modificar ciertos aspectos de sus asignaciones (cuando esté permitido).

Publish.safe: Confirma que una asignación puede ser publicada correctamente cuando cumple con todos los requisitos (información completa, fechas válidas, etc.). El sistema impide la publicación de asignaciones mal configuradas o con datos faltantes, mostrando mensajes de error pertinentes.

List.hack: Evalúa la resistencia del sistema frente a accesos ilegítimos a listas de asignaciones ajenas, tanto desde el mismo realm y diferente usuario, como desde diferente realm, como desde un usuario no autenticado. Se manipulan rutas y parámetros con el fin de exponer información de otros usuarios, sin éxito gracias a los controles de seguridad implementados.

Show.hack: Simula el intento de acceder al detalle de una asignación no autorizada (ya sea inexistente o perteneciente a otro miembro de la tripulación). El sistema bloquea el acceso y evita la exposición de datos.

Create.hack: Simula el intento de crear una asignación de vuelo desde un usuario no autenticado, desde otro realm y desde el realm correcto, pero manipulando el cuerpo de la solicitud con intenciones de introducir valores inválidos.

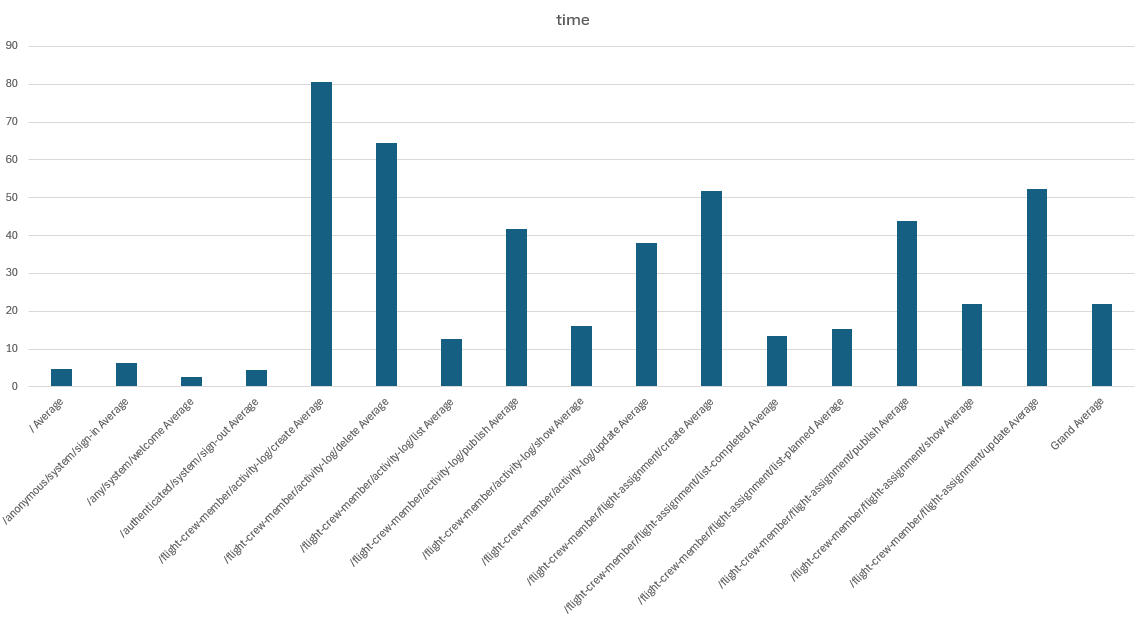
Update.hack: Prueba modificaciones no permitidas a asignaciones ajenas o ya publicadas, mediante manipulación del cuerpo de la solicitud o URLs. El sistema detecta estos intentos y responde de forma segura, sin aplicar cambios.

Publish.hack: Simula intentos no autorizados de publicar asignaciones mediante manipulación del cuerpo de la solicitud o URLs. El sistema rechaza correctamente estas acciones si el usuario no tiene permisos o si la asignación no le pertenece.

## Análisis de rendimiento

Ahora se procederá a realizar un análisis de rendimiento del sistema mediante la ejecución de las pruebas funcionales previamente descritas. Estas pruebas se han llevado a cabo bajo dos escenarios distintos:

Sin índices para la optimización de consultas: Se han calculado los resultados promedio de rendimiento de las operaciones de prueba, los cuales se ilustran en el siguiente gráfico:



Podemos comprobar que la feature más ineficiente es la que se ocupa de la creación de los registros de actividad, la cual sobrepasa los 80 milisegundos.

Estadísticas sin índices:

A screenshot of a spreadsheet

AI-generated content may be incorrect.

Con índices para la optimización de consultas y perfilado de software y hardware:

Índices utilizados en la entidad ActivityLog:

Texto

El contenido generado por IA puede ser incorrecto.

Índices utilizados en la entidad FlightAssignment:

Texto

El contenido generado por IA puede ser incorrecto.

Se han añadido estos índices con motivo de reducir el tiempo de ejecución de la solicitud de creación de un registro de actividad, en la cual se utilizan varios métodos de los repositorios. En concreto, se ha intentado reducir el tiempo de ejecución de las siguientes consultas:







A continuación, se muestran los resultados promedio de rendimiento de las operaciones de testing con índices, tal como se muestra en el gráfico siguiente:

A graph with blue bars

AI-generated content may be incorrect.

Como podemos comprobar, la MIR (most inefficient request) sigue siendo la de la creación de registros de actividad, sin embargo, al introducir los índices, su tiempo de ejecución ha bajado considerablemente, al igual que el resto de solicitudes.

Estadísticas con índices:

A screenshot of a spreadsheet

AI-generated content may be incorrect.

# Comparativa

Para realizar la comparativa, se ha utilizado un z-test, ya que intentar comparar los intervalos de confianza obtenidos es una tarea poco intuitiva.

Resultados del z-test:

A screenshot of a data

AI-generated content may be incorrect.

Podemos comprobar que el valor de two-tail p-value se encuentra en el intervalo [0, 0.05), por lo que está claro que se ha conseguido una mejora significativa real.

# Conclusiones

Gracias al análisis realizado, podemos observar que se ha obtenido una cobertura superior al 99%, tanto en las features del requisito 8 como en las del 9, aportando un gran valor a la hora de la detección de bugs y hackeos, ayudando al cumplimiento de la Ley Orgánica 3/2018, de 5 de diciembre, de Protección de Datos Personales y garantía de los derechos digitales. También se ha podido detectar la MIR, para posteriormente introducir índices que han mejorado considerablemente la eficiencia de nuestra aplicación.

# Bibliografía

[En blanco intencionadamente]