Universidad de Sevilla  
Escuela Técnica Superior de Ingeniería Informática

TESTING REPORT INDIVIDUAL – C2



**C2.009**

**Repositorio**: [DP2-2024-2025-C1-009/Acme-ANS-D04](https://github.com/DP2-2024-2025-C1-009/Acme-ANS-D04)

**Autor:**

Gallero Rodríguez, Carlos

cargalrod@alum.us.es

**Fecha:** 28/06/2025

**Tabla de contenidos**

[**Resumen**](#_heading=h.gjdgxs) **2**

[**Tabla de revisión**](#_heading=h.30j0zll) **2**

[**Introducción**](#_heading=h.1fob9te) **2**

[**Contenidos**](#_heading=h.3znysh7) **3**

**Capítulo I: Testing funcional 3**

**Capítulo II: Testing de rendimiento 4**

[**Bibliografía**](#_heading=h.3dy6vkm) **4**

# Resumen

Este informe recoge el proceso de testing llevado a cabo sobre los requisitos individuales de entregas anteriores, con el propósito de analizar los resultados obtenidos y extraer conclusiones orientadas a la mejora del código o del propio procedimiento de pruebas. El documento se estructura en dos capítulos principales, en los que se aplican métodos estadísticos para el cálculo de intervalos de confianza y la realización de contrastes de hipótesis, con el fin de ofrecer una valoración fundamentada de los resultados.

# Tabla de revisión

|  |  |  |
| --- | --- | --- |
| **Versión** | **Fecha** | **Descripción** |
| V1 | 26/05/2025 | Creación y desarrollo del documento |
| V2 | 28/06/2026 | Actualización del documento |

# Introducción

A continuación, veremos separados en dos capítulos, los métodos estadísticos realizados para analizar los resultados tras el testing formal de los requisitos de todas las entregas del vuelo.

# Contenidos

# Capítulo I: Testing funcional

Este capítulo mostrará el listado con los casos de prueba implementados, agrupados por funcionalidades. Para cada funcionalidad a probar se explicarán los casos positivos y negativos probados en los archivos .safe y los intentos de hackeo realizados en los archivos .hack.

**Entidad FlightAssignment**

**flightCrewMember/flightAssignment/create:**

Para probar esta funcionalidad hemos iniciado sesión con el usuario member, hemos accedido a su listado de asignaciones planeadas y hemos pulsado sobre el botón create. Primero hemos intentado crear una asignación con todas las casillas del formulario en blanco y luego poniendo valores erróneos en cada apartado. En ambos casos nos han salido los mensajes de error correspondientes. Por último, hemos creado una asignación con datos correctos y se ha creado correctamente.

En la prueba de hacking, hemos intentado poner un tramo no válido mediante inyección y nos ha saltado el error correspondiente, también hemos intentado crear una asignación sin haber iniciado sesión y también nos ha saltado el error de autorización.

Hemos logrado un 96,8% de cobertura en esta funcionalidad. No hemos alcanzado el 100 % de cobertura debido principalmente al bloque catch del método unbind, el cual se encarga de capturar errores al construir el SelectChoices de tramos.

**flightCrewMember/flightAssignment/delete:**

Para probar esta funcionalidad hemos accedido a una asignación de member1 y lo hemos borrado correctamente. También hemos borrado uno accediendo desde el navegador para lograr la cobertura del método unbind.

En la prueba de hacking, hemos intentado borrar una asignación que ya estaba publicado y también hemos intentado borrar una asignación de otro usuario y sin haber iniciado sesión. En todos los casos hemos obtenido el error esperado.

La cobertura de los test es de un 97,8%. No se ha logrado el 100% de cobertura principalmente porque no hemos eliminado ninguna asignación que tuviera registros de actividad registrados.

**flightCrewMember/flightAssignment/listPlanned:**

Para probar esta funcionalidad hemos iniciado sesión con member1 y hemos accedido a sus listados asignaciones planeadas.

Para la parte de hacking hemos intentado acceder al listado de asignaciones planeadas de member1 habiendo iniciado sesión con member2. También hemos intentado acceder a este listado sin haber iniciado sesión. En ambos casos nos ha saltado el error de autorización correspondiente.

**flightCrewMember/flightAssignment/listCompleted:**

Para probar esta funcionalidad hemos iniciado sesión con member1 y hemos accedido a sus listados de asignaciones completadas.

Para la parte de hacking hemos intentado acceder al listado de asignaciones completadas de member1 habiendo iniciado sesión con member2. También hemos intentado acceder a este listado sin haber iniciado sesión. En ambos casos nos ha saltado el error de autorización correspondiente.

**flightCrewMember/flightAssignment/publish:**

Para probar esta funcionalidad hemos accedido al formulario de una asignación planeado, ya creado y en modo borrador. Hemos puesto las casillas en blanco y hemos intentado publicar, luego hemos puesto datos erróneos y también hemos intentado publicar. En ambos casos nos han salido los mensajes de error correspondientes. Por último, hemos publicado una asignación con datos correctos y hemos comprobado que ya no estaba en modo borrador.

Para la parte de hacking hemos intentado publicar una asignación sin haber iniciado sesión, habiendo iniciado sesión con member1 hemos intentado publicar una asignación de member2 y también hemos intentado publicar habiendo inyectado un tramo erróneo. En todos los casos nos ha saltado el error de autorización correspondiente.

Se ha conseguido un 96,7% de cobertura en esta funcionalidad. El bloque que hace que no se logre mayor cobertura es el mismo que en la funcionalidad de create (el bloque catch del método unbind, el cual se encarga de capturar errores al construir el SelectChoices de tramos).

**flightCrewMember/flightAssignment/show:**

Para probar esta funcionalidad, se ha accedido a la vista de detalles de una asignación planeado y luego de un completado. Se han podido ver los detalles de ambas asignaciones sin problema.

En la prueba de hacking hemos intentado acceder al show de una asignación de otro usuario y también lo hemos intentado sin haber iniciado sesión. En ambos casos nos han salido los errores de autorización correspondientes.

**flightCrewMember/flightAssignment/update:**

Para probar esta funcionalidad hemos accedido al formulario de una asignación planeado, ya creado y en modo borrador. Hemos puesto las casillas en blanco y hemos intentado actualizar, luego hemos puesto datos erróneos y también hemos intentado actualizar. En ambos casos nos han salido los mensajes de error correspondientes. Por último, hemos actualizado una asignación con datos correctos y hemos comprobado que en listado aparecían los nuevos datos actualizados.

Para la parte de hacking hemos intentado publicar una asignación sin haber iniciado sesión y luego habiendo iniciado sesión con member1 hemos intentado publicar una asignación de member. En ambos casos nos ha saltado el error de autorización correspondiente.

Se ha conseguido un 95,5% de cobertura en esta funcionalidad. El bloque que hace que no se logre mayor cobertura es el mismo que en la funcionalidad de create (el bloque catch del método unbind, el cual se encarga de capturar errores al construir el SelectChoices de tramos).

.

**Entidad ActivityLog**

**flightCrewMember/activityLog /create:**

Para probar esta funcionalidad hemos intentado crear un log poniendo todas las casillas del formulario vacías, luego hemos puesto datos erróneos en las casillas y por último hemos creado un registro de actividad correctamente. También hemos intentado crear un registro de actividad para una asignación cuyo tramo aún no había finalizado.

Para la parte de hacking hemos intentado crear un registro de actividad si haber iniciado sesión. También hemos intentado crear un registro de actividad para una de otro miembro. En ambos casos nos ha saltado el error de autorización correspondiente.

**flightCrewMember/activityLog/delete:**

Para probar esta funcionalidad hemos borrado un registro de actividad de una asignación en modo borrador y hemos comprobado que se ha eliminado correctamente. Luego hemos eliminado otro de una asignación que no estaba en modo borrador.

Para el hacking hemos intentado borrar un registro de actividad que ya estaba publicado y también hemos intentado borrar uno de otro crew member. En todos los casos nos ha saltado el error de autorización correspondiente.

**flightCrewMember/activityLog/list:**

Para verificar esta funcionalidad se han accedido a las listas de registros de actividad de member1.

En las pruebas de hacking se ha intentado acceder a estas listas sin haber iniciado sesión y ha saltado el error de autorización

**flightCrewMember/activityLog/publish:**

Para probar esta funcionalidad hemos accedido a un registro de actividad en modo borrador y hemos intentado publicarlo poniendo todas las casillas en blanco. Luego hemos intentado publicarlo poniendo valores erróneos. En ambos casos nos han salido los mensajes de error correspondientes. Por último, hemos publicado correctamente un registro de actividad y hemos comprobado en el listado que el registro de actividad ya no estuviese en modo borrador.

Para el hacking hemos intentado publicar un registro de actividad que ya estuviera publicado, hemos intentado publicar un registro de actividad de otro crew member y hemos intentado publicar un registro de actividad sin haber iniciado sesión. En todos los casos nos ha salido el error de autorización correspondiente.

**flightCrewMember/activityLog/show:**

Para probar esta funcionalidad hemos accedido correctamente a los detalles de distintos registros de actividad.

En las pruebas de hacking se ha intentado acceder a los detalles de registros de actividad sin haber iniciado sesión y ha saltado el error de autorización

**flightCrewMember/activityLog/update:**

Para probar esta funcionalidad hemos accedido a un registro de actividad en modo borrador y hemos intentado actualizarlo poniendo todas las casillas en blanco. Luego hemos intentado actualizarlo poniendo valores erróneos. En ambos casos nos han salido los mensajes de error correspondientes. Por último, hemos actualizado correctamente un registro de actividad y hemos comprobado en el listado que los datos estaban actualizados.

Para el hacking hemos intentado actualizar un registro de actividad que no estuviera en modo borrador, hemos intentado actualizar un registro de actividad de otro crew member y hemos intentado actualizar un registro de actividad sin haber iniciado sesión. En todos los casos nos ha salido el error de autorización correspondiente.

En esta funcionalidad se ha alcanzado un 92,9% de cobertura, sin ningún bloque en rojo. Las líneas que no han sido cubiertas completamente están en amarillo, lo cual indica que han sido ejecutadas parcialmente en algunos flujos, pero no en todas sus posibles combinaciones lógicas. Para alcanzar el 100% sería necesario realizar más pruebas específicas que combinen condiciones concretas como: intentar modificar un registro de actividad cuya asignación de vuelo aún no haya finalizado, o probar situaciones en las que el modo borrador esté desactivado

Interfaz de usuario gráfica

El contenido generado por IA puede ser incorrecto.

ACME.CONSTRAINTS





Aquí podemos comprobar la cobertura en las validaciones a nivel de entidad de ActivityLog y de FlightAssignment (ActivityLogValidator y FlightAssignmentValidator). No se ha llegado al 100% de cobertura en estos casos ya que existen validaciones que no se pueden ejecutar probando la aplicación. Estas validaciones están incluidas para comprobar los datos de prueba.

# Capítulo II: Testing de rendimiento

Se presentan los resultados de las pruebas de rendimiento. Se incluyen gráficos representativos y un análisis estadístico del tiempo de respuesta del sistema con un intervalo de confianza del 95% En primer lugar se ejecutaron los tests sobre el repositorio sin ningún tipo de cambio relacionado con optimizar el rendimiento.

Después se corrieron las mismas pruebas pero haciendo uso de @Index para tratar de optimizar las consultas del repositorio lo máximo posible (según lo aprendido en el S02 de la lección 4) Primero analizaremos los datos de la primera ejecución y luego los compararemos con la segunda.

**Antes**

Los resultados obtenidos se pueden visualizar en las siguientes imágenes:

Tabla

El contenido generado por IA puede ser incorrecto.

Estos son los datos estadísticos resumidos obtenidos para la primera prueba sin realizar los índices.

Gráfico, Gráfico de barras

El contenido generado por IA puede ser incorrecto.

En esta gráfica podemos observar el tiempo medio de respuesta por tipo de petición. Podemos comprobar que el MIR es el show de flightAssignment. Sin embargo, no hay ninguna funcionalidad que resaltes sobre las demás.

Tras obtener estos datos consulte el servicio de show de flightAssignments para comprobar posibles mejoras a simple vista.

Antes de hacer cambios en el código quería realizar las pruebas habiendo incluidos los índices porque podrían estar muy relacionados con las peticiones que más tardan en ejecutarse.

Interfaz de usuario gráfica, Aplicación, Tabla, Excel

El contenido generado por IA puede ser incorrecto.

**Después**

Tras agregar los índices en las entidades de ActivityLog y FlightAssignment obtuve las siguientes estadísticas.

Tabla

El contenido generado por IA puede ser incorrecto.

Como podemos observar, ha habido un cambio significativo con respecto a “antes” en los datos estadísticos proporcionados por la columna time.

Para comparar adecuadamente los intervalos de confianza calculados tras ejecutar las pruebas, se ha realizado la prueba Z-Test sobre las columnas de tiempos generados.

Los datos obtenidos son los siguientes:

Interfaz de usuario gráfica, Aplicación, Tabla

El contenido generado por IA puede ser incorrecto.

El valor que nos interesa es Valor crítico de z (dos colas), el que aparece arriba.

Dicho valor nos indica que nos encontramos en el intervalo ( 0.00 - alpha ], donde, alpha = 1 - Intervalo de Confianza = 1 - 0.95 = 0.05 Al ser z = 0 tenemos que: 0 = < z < alpha. El p-value está en el intervalo [0.00,alpha).

Tras la obtención y análisis de estos datos podemos concluir que se ha aumentado el rendimiento y se ha reducido el tiempo medio de petición. Por lo tanto, la implementación de los índices en las entidades ha sido un éxito.

**Comparativa con el ordenador de un compañero**

Este apartado consiste en comparar y analizar las diferencias entre el rendimiento obtenido desde dos máquinas distintas.

En esta ocasión haré la comparativa con el ordenador de mi compañero de grupo Jaime Gómez Marín. Para la comparativa he usado la versión sin índices.

Ordenador personal: Ryzen 7 4700U, 16GB RAM, 512 SSD

Ordenador de Jaime: Intel Core i7-1165G7, 16 GB RAM, 512 GB SSD

Tabla

El contenido generado por IA puede ser incorrecto.

Gráfico

El contenido generado por IA puede ser incorrecto.

Interfaz de usuario gráfica, Aplicación, Tabla, Excel

El contenido generado por IA puede ser incorrecto.

Interfaz de usuario gráfica, Aplicación, Tabla

El contenido generado por IA puede ser incorrecto.

Tras analizar los datos podemos concluir que el ordenador de mi compañero Jaime tiene mucho mejor rendimiento para este tipo de trabajos. En la gráfica de barras podemos comprobar como, aunque baja en todos los tipos de peticiones, la que consume más tiempo en este caso es el publish de flightAssignment. El p-value obtenido es 0, valor perteneciente al rango [0.00, alpha), siendo alpha 0.05. Esto nos resalta el cambio tan grande de rendimiento entre una máquina y otra.

# Bibliografía

-Documento 08 Annexes, de los contenidos de la plataforma virtual de esta asignatura.

-S02 Performance testing, transparencias del módulo de testing L04 de la asignatura