Logotipo

El contenido generado por IA puede ser incorrecto.

Testing Report

Mohanad JD Abulatifa

DP II 24-25

03/07/2025

**Contenido**

[**Tabla de versiones 1**](#_Toc202479177)

**[Resumen ejecutivo 2](#_Toc202479178)**

[**Introducción 2**](#_Toc202479179)

[**Formal testing 3**](#_Toc202479180)

[**Performance testing 6**](#_Toc202479181)

[**Comparativa completa de rendimiento, cobertura y entorno 8**](#_Toc202479182)

[**Resultados en entorno 1 (sin índices, en máquina virtual) 8**](#_Toc202479183)

[**Resultados en entorno 2 (con índices, en otro ordenador) 8**](#_Toc202479184)

[**Comparativa técnica 9**](#_Toc202479185)

[**Consideración sobre el entorno virtual 9**](#_Toc202479186)

[**Conclusión 9**](#_Toc202479187)

|  |  |  |
| --- | --- | --- |
| Versión | Fecha | Descripción |
| 1.0 | 03/07/2025 | Generación del informe |

# Tabla de versiones

# Resumen ejecutivo

El contenido de este informe documenta clara y concisamente el valor aportado por la realización de pruebas formales en nuestro proyecto, así mismo, se expone los resultados y los puntos de mejora que esto ha conseguido, haciendo de una aplicación totalmente funcional y que deseamos que sea aceptada formalmente por nuestro cliente.

# Introducción

En cualquier proyecto es importante asegurarnos que la calidad del producto o servicio que proporcionamos a nuestro cliente es suficiente como para una aceptación formal y un reconocimiento de estos hechos. Por y para ello, las pruebas formales han sido cruciales, sin quitarle mérito a las pruebas informales realizadas por los desarrolladores durante el desarrollo de este.

Durante este documento recorreremos los diferentes puntos donde se detallan los resultados y las métricas obtenidas al ejecutar las pruebas formales en nuestro equipo, además se han replicado estas pruebas en otro equipo para comparar las métricas obtenidas y tratar de entender si los requerimientos de la máquina han podido afectar al resultado obtenido.

# Formal testing

A continuación, se van a describir brevemente las diferentes pruebas realizadas agrupados por cada una de las funcionalidades tal y como se establece en la metodología acordada con el cliente. Con el fin de no repetir información indicar al cliente que se han seguido las instrucciones para garantizar el correcto funcionamiento de la aplicación y proporcionado un conjunto de datos que cumple con los estándares de calidad impuestos.

Texto

Descripción generada automáticamente

Texto

Descripción generada automáticamente

**LIST**

Se han realizado pruebas completas de tipo safe, safe-negative y hack para validar el correcto funcionamiento de las funcionalidades de listado (List) tanto en FlightAssignment como en ActivityLog.  
En ActivityLog, se ha alcanzado un 100% de cobertura, garantizando que el sistema muestra correctamente los registros asociados, incluyendo casos con múltiples logs, un único log y sin registros. Además, se ha verificado el acceso únicamente por parte del usuario legítimo, así como la correcta navegación de relaciones internas.  
En FlightAssignment, la cobertura ha sido del 86,1%, debido a que algunas líneas del método unbind (relacionadas con el cálculo de disponibilidad y carga de elecciones auxiliares) no se ejecutan de forma directa en la traza, pero no afectan al comportamiento funcional ni a la seguridad.  
Las pruebas han incluido diferentes contextos de usuario (miembros con muchas, pocas o ninguna asignación) y han confirmado la correcta respuesta del sistema en todos los casos, además del rechazo de accesos no autorizados mediante pruebas de tipo hack.  
El rendimiento observado en ambos casos ha sido razonable, aunque en FlightAssignment se apreció cierta penalización por la carga adicional de entidades relacionadas como Leg y Flight, lo que justifica técnicamente la incorporación de índices sobre los campos flightCrewMember\_id y flightAssignment\_id respectivamente.

**CREATE**

Esta funcionalidad ha sido esencial para validar las reglas del sistema y prevenir inserciones maliciosas o incoherentes.  
En FlightAssignment, se ha comprobado que:

* No se puede crear una asignación con legs incompatibles.
* Un mismo miembro no puede tener dos asignaciones superpuestas.
* Solo miembros con estado AVAILABLE pueden crear nuevas asignaciones (Member2, Member3) no están en estado AVAILABLE.  
  En ActivityLog, se ha validado que solo se puedan registrar logs si:
* El leg ha finalizado.
* El flightAssignment está confirmado.
* El registro es posterior a la llegada.  
  Los índices sobre leg\_id, flightCrewMember\_id y flightAssignment\_id resultan críticos para mejorar el rendimiento de estas validaciones.

**SHOW**

Se ha conseguido una cobertura del **100 %** en ambas funcionalidades.  
En FlightAssignment, se han evaluado correctamente las relaciones y la navegación entre flight, leg y flightCrewMember. También se comprobó que un miembro solo puede ver sus propias asignaciones.  
En ActivityLog, se validó que solo el autor del log puede visualizarlo. Las pruebas garantizaron que no se pudiera alterar el contenido desde la vista.  
Índices utilizados: id (por defecto) y flightAssignment\_id para mejorar la recuperación de datos.

**UPDATE**

Cobertura alta en ambos servicios. Se utilizó la herramienta de pruebas para identificar posibles inconsistencias.  
En FlightAssignment se validaron reglas como el estado del miembro, restricciones de tiempo y exclusividad de piloto/copiloto.  
En ActivityLog, la edición se bloquea si los requisitos iniciales (fecha posterior, assignment confirmado) no se cumplen.  
Resumen de testing – FlightAssignment y ActivityLog: se realizaron refactorizaciones menores. Los índices sobre leg\_id y flightAssignment\_id siguen siendo relevantes.

**DELETE**

Las pruebas fueron efectivas y se alcanzó una cobertura del **97 %** en ambos servicios.  
En FlightAssignment, solo es posible eliminar si la asignación está en draftMode y sin logs asociados.  
En ActivityLog, la eliminación está restringida al creador. Se evitaron casos de acceso indebido mediante pruebas hack bien diseñadas.

**PUBLISH**

Funcionalidad similar a Update, pero más restrictiva ya que marca el estado definitivo del recurso.  
se comprobó que existan los roles necesarios y sin duplicidades antes de publicar.  
Se han validado correctamente los flags draftMode y se han prevenido accesos no autorizados.

## Performance testing

Interfaz de usuario gráfica, Texto, Aplicación, Tabla

Descripción generada automáticamente

Tabla

Descripción generada automáticamente

Gráfico

Descripción generada automáticamente

## Comparativa completa de rendimiento, cobertura y entorno

Se ha llevado a cabo un análisis detallado del rendimiento y cobertura de la aplicación en dos entornos distintos, tanto **con como sin índices** en base de datos. Además, se ha evaluado la diferencia en tiempos de respuesta y cobertura entre ambos entornos, con el objetivo de identificar mejoras aplicables.

### Resultados en entorno 1 (sin índices, en máquina virtual)

* **Promedio general**: **47,21 ms**
* **Cobertura media FlightAssignment**: 94,7 %
* **Cobertura media ActivityLog**: 98,3 %

Destacan tiempos elevados en:

* /flight-assignment/delete: **230 ms**
* /flight-assignment/create: **119 ms**
* /activity-log/create: **107 ms**

Pese a realizar correctamente todos los tests. safe, negative y .hack, se detectó que ciertas consultas **dependían de atributos relacionales no indexados** (como leg, flight, assignment…), generando sobrecarga en los accesos.

### Resultados en entorno 2 (con índices, en otro ordenador)

* **Promedio general**: **8,15 ms**
* **Cobertura media FlightAssignment**: 94,7 %
* **Cobertura media ActivityLog**: 98,3 %

Tiempos significativamente reducidos:

* /flight-assignment/delete: **12,75 ms**
* /flight-assignment/create: **25,91 ms**
* /activity-log/create: **23,43 ms**

Los índices añadidos sobre columnas como flightCrewMember\_id, flightAssignment\_id, leg\_id demostraron ser muy efectivos para operaciones de lectura y filtrado masivo. Los resultados muestran una mejora **más del 80 %** en el rendimiento general de ejecución.

### Comparativa técnica

| **Métrica** | **Sin índices (VM)** | **Con índices (otro PC)** |
| --- | --- | --- |
| Promedio global | 47,21 ms | 8,15 ms |
| Máximo observado | 230 ms | 26,3 ms |
| Mínimo observado | 13,78 ms | 1,56 ms |
| Rango | 216 ms | 24,76 ms |
| Cobertura promedio FA | 94,7 % | 94,7 % |
| Cobertura promedio AL | 98,3 % | 98,3 % |

### Consideración sobre el entorno virtual

Es importante señalar que el entorno con menor rendimiento fue ejecutado en una **máquina virtual**, lo que **puede haber influido negativamente en los tiempos de respuesta observados**. Las máquinas virtuales generalmente cuentan con recursos limitados, mayor latencia de disco y CPU compartida, factores que agravan el rendimiento en operaciones intensivas de E/S, como las pruebas de rendimiento sobre la base de datos.  
Por tanto, aunque la ausencia de índices es el principal factor que explica la diferencia, **el uso de una máquina virtual también es una causa secundaria plausible del rendimiento degradado**.

Conclusión

Finalmente, cabe destacar la incuestionable labor de nuestro equipo generando un conjunto de pruebas que han permitido que el cliente haga un uso efectivo y satisfactorio de la aplicación. Nos gustaría agradecer expresamente al cliente su alta participación en este proyecto y la comunicación continua con el mismo.