Informe de Pruebas – Student #1

**Grupo:** C1.063

**Repositorio:** <https://github.com/DP2-C1-063/Acme-ANS-D04>

**Integrantes:**

* Lucía Ponce García de Sola – [lucpongar@alum.us.es](mailto:lucpongar@alum.us.es)
* Raquel Ortega Almirón – [raqortalm@alum.us.es](mailto:raqortalm@alum.us.es)
* Jose Ángel Herrera Romero – [josherrom@alum.us.es](mailto:josherrom@alum.us.es)
* Manuel Toledo González – [mantolgon@alum.us.es](mailto:mantolgon@alum.us.es)
* Carlos Palma Santos – [carpalsan1@alum.us.es](mailto:carpalsan1@alum.us.es)

**Fecha:** 26/05/2025

Contenido

[1. Resumen ejecutivo 3](#_Toc199119716)

[2. Tabla de revisiones 3](#_Toc199119717)

[3. Introducción 3](#_Toc199119718)

[4. Pruebas funcionales 4](#_Toc199119719)

[5. Pruebas de rendimiento 11](#_Toc199119720)

[6. Conclusiones 11](#_Toc199119721)

[7. Bibliografía 11](#_Toc199119722)

# Resumen ejecutivo

Este informe presenta los resultados de las pruebas funcionales y de rendimiento realizadas sobre el sistema desarrollado por el estudiante #1 como parte del proyecto Acme-ANS del grupo C1.063. El objetivo principal fue validar el correcto funcionamiento de las funcionalidades implementadas y evaluar el rendimiento del sistema bajo condiciones controladas.

Se implementaron 56 casos de prueba, agrupados por funcionalidad y entidad. Algunos de los casos lograron detectar defectos de forma efectiva durante el desarrollo y la mayoría validaron el comportamiento esperado del sistema.

En cuanto a las pruebas de rendimiento, se evaluó el tiempo de respuesta del sistema (wall time) en dos ordenadores diferentes. Se calcularon intervalos de confianza al 95% para los tiempos promedio y se realizó un contraste de hipótesis que permitió determinar cuál de los dos equipos ofrece un rendimiento superior. Los resultados reflejan un desempeño estable del sistema, con diferencias de rendimiento esperadas según las características del hardware.

Este informe sirve como base para la evaluación técnica del sistema y permite identificar áreas de mejora tanto en términos de funcionalidad como de eficiencia.

# Tabla de revisiones

|  |  |  |
| --- | --- | --- |
| Revisión | Fecha | Descripción breve |
| 1.0 | 26/05/2025 | Versión inicial |

# Introducción

Este informe documenta las pruebas funcionales y de rendimiento. El objetivo de las pruebas es verificar que las funcionalidades implementadas operan correctamente y evaluar el rendimiento del sistema en distintos entornos.

Las pruebas funcionales se organizaron por entidad y características clave del sistema. Las pruebas de rendimiento se centraron en medir el tiempo de respuesta del sistema, comparando su comportamiento en dos equipos distintos y aplicando análisis estadísticos.

El documento se estructura de la siguiente manera: la sección 4 describe los casos de prueba funcionales, agrupados por funcionalidad, con una valoración de su efectividad; la sección 5 presenta los resultados de las pruebas de rendimiento, incluyendo gráficos, intervalos de confianza y contraste de hipótesis. El informe finaliza con una conclusión y la bibliografía correspondiente.

# Pruebas funcionales

En esta sección se detallan los casos de prueba implementados, organizados por funcionalidad. Para cada caso de prueba, se incluye una breve descripción y una evaluación de su efectividad en la detección de errores.

**Funcionalidad: Listar vuelos**

* Caso de prueba 1:
  + Descripción: Listar vuelos con sesión iniciada (para managers 0 y 1)
  + Resultado: Éxito
  + Efectividad para detectar errores: Baja
* Caso de prueba 2 (hacking):
  + Descripción: Intentar listar vuelos sin iniciar sesión
  + Resultado: Falla
  + Efectividad: Media

**Funcionalidad: Ver vuelo**

* Caso de prueba 1:
  + Descripción: Mostrar los detalles de un vuelo con sesión iniciada
  + Resultado: Éxito
  + Efectividad para detectar errores: Baja
* Caso de prueba 2 (hacking):
  + Descripción: Ver un vuelo sin iniciar sesión
  + Resultado: Falla
  + Efectividad para detectar errores: Alta
* Caso de prueba 3 (hacking):
  + Descripción: Ver un vuelo inexistente
  + Resultado: Falla
  + Efectividad para detectar errores: Alta
* Caso de prueba 4 (hacking):
  + Descripción: Ver un vuelo que pertenece a otro manager
  + Resultado: Falla
  + Efectividad para detectar errores: Alta

**Funcionalidad: Crear un vuelo**

* Caso de prueba 1:
  + Descripción: Crear un vuelo con todos sus datos nulos
  + Resultado: Error
  + Efectividad para detectar errores: Media
* Caso de prueba 2:
  + Descripción: Introducción de valores inválidos
  + Resultado: Error
  + Efectividad para detectar errores: Alta
* Caso de prueba 3:
  + Descripción: Introducción de valores válidos
  + Resultado: Error / Éxito (cuando todos los valores son válidos simultáneamente)
  + Efectividad para detectar errores: Baja
* Caso de prueba 4 (hacking):
  + Descripción: Creación con id definido en consola (distinto de cero)
  + Resultado: Falla
  + Efectividad para detectar errores: Alta

**Funcionalidad: Actualizar un vuelo**

* Caso de prueba 1:
  + Descripción: Actualización de un vuelo con valores válidos
  + Resultado: Éxito
  + Efectividad para detectar errores: Baja
* Caso de prueba 2 (hacking):
  + Descripción: Actualizar un vuelo que pertenece a otro manager
  + Resultado: Falla
  + Efectividad para detectar errores: Alta
* Caso de prueba 3 (hacking):
  + Descripción: Actualizar un vuelo previamente publicado
  + Resultado: Falla
  + Efectividad para detectar errores: Alta
* Caso de prueba 4 (hacking):
  + Descripción: Actualizar un vuelo que no existe
  + Resultado: Falla
  + Efectividad para detectar errores: Alta

**Funcionalidad: Publicar un vuelo**

* Caso de prueba 1:
  + Descripción: Publicación de un vuelo sin tramos asociados
  + Resultado: Error
  + Efectividad para detectar errores: Alta
* Caso de prueba 2:
  + Descripción: Publicación de un vuelo con tramos sin publicar
  + Resultado: Error
  + Efectividad para detectar errores: Alta
* Caso de prueba 3 (hacking):
  + Descripción: Publicación de un vuelo que no existe
  + Resultado: Falla
  + Efectividad para detectar errores: Media
* Caso de prueba 4 (hacking):
  + Descripción: Publicación de un vuelo que pertenece a otro manager
  + Resultado: Falla
  + Efectividad para detectar errores: Alta
* Caso de prueba 5 (hacking):
  + Descripción: Publicación de un vuelo previamente publicado
  + Resultado: Falla
  + Efectividad para detectar errores: Baja

**Funcionalidad: Eliminar un vuelo**

* Caso de prueba 1:
  + Descripción: Eliminación de un vuelo y sus tramos asociados. Se probó también la eliminación de reclamaciones y reservas asociadas, y todas las entidades de las que estas dependían. Por esto podemos observar la funcionalidad de creación de *booking records* durante las pruebas de rendimiento, pues fue necesario generar un dato de ejemplo.
  + Resultado: Éxito
  + Efectividad para detectar errores: Alta
* Caso de prueba 2 (hacking):
  + Descripción: Eliminación de un vuelo que pertenece a otro manager
  + Resultado: Fallo
  + Efectividad para detectar errores: Alta
* Caso de prueba 3 (hacking):
  + Descripción: Eliminación de un vuelo previamente publicado
  + Resultado: Fallo
  + Efectividad para detectar errores: Alta
* Caso de prueba 4 (hacking):
  + Descripción: Eliminación de un vuelo inexistente
  + Resultado: Fallo
  + Efectividad para detectar errores: Baja

**Funcionalidad: Listar tramos (legs)**

* Caso de prueba 1:
  + Descripción: Listar los tramos de un vuelo propio con sesión iniciada
  + Resultado: Éxito
  + Efectividad para detectar errores: Baja
* Caso de prueba 2 (hacking):
  + Descripción: Intentar listar tramos sin iniciar sesión
  + Resultado: Falla
  + Efectividad para detectar errores: Media
* Caso de prueba 3 (hacking):
  + Descripción: Listar tramos de un vuelo perteneciente a otro manager
  + Resultado: Falla
  + Efectividad para detectar errores: Alta

**Funcionalidad: Ver tramo (leg)**

* Caso de prueba 1:
  + Descripción: Mostrar el detalle de un tramo con sesión iniciada
  + Resultado: Éxito
  + Efectividad para detectar errores: Baja
* Caso de prueba 2 (hacking):
  + Descripción: Ver un tramo sin iniciar sesión
  + Resultado: Falla
  + Efectividad para detectar errores: Alta
* Caso de prueba 3 (hacking):
  + Descripción: Ver un tramo inexistente
  + Resultado: Falla
  + Efectividad para detectar errores: Alta
* Caso de prueba 4 (hacking):
  + Descripción: Ver un tramo cuyo vuelo pertenece a otro manager
  + Resultado: Falla
  + Efectividad para detectar errores: Alta

**Funcionalidad: Crear tramo**

* Caso de prueba 1:
  + Descripción: Crear un tramo con todos sus datos nulos
  + Resultado: Error
  + Efectividad para detectar errores: Media
* Caso de prueba 2:
  + Descripción: Introducción de valores inválidos. Entre ellos: número de vuelo previamente utilizado, momento de aterrizaje previo al despegue, duración mayor a 1000 minutos, duración menor a 1 minuto y aeropuerto de salida y llegada idénticos.
  + Resultado: Error
  + Efectividad para detectar errores: Alta
* Caso de prueba 3:
  + Descripción: Introducción de valores válidos
  + Resultado: Error / Éxito (cuando todos los valores son válidos simultáneamente)
  + Efectividad para detectar errores: Baja
* Caso de prueba 4 (hacking):
  + Descripción: Creación con id definido en consola (distinto de cero)
  + Resultado: Falla
  + Efectividad para detectar errores: Alta
* Caso de prueba 5 (hacking):
  + Descripción: Creación de un tramo cuyo vuelo pertenece a otro manager
  + Resultado: Falla
  + Efectividad para detectar errores: Alta
* Caso de prueba 6 (hacking):
  + Descripción: Creación de un tramo cuyo vuelo está publicado
  + Resultado: Falla
  + Efectividad para detectar errores: Alta
* Caso de prueba 7 (hacking):
  + Descripción: Creación con aeropuerto de salida con un id inexistente
  + Resultado: Falla
  + Efectividad para detectar errores: Alta
* Caso de prueba 8 (hacking):
  + Descripción: Creación con aeropuerto de llegada con un id inexistente
  + Resultado: Falla
  + Efectividad para detectar errores: Alta
* Caso de prueba 9 (hacking):
  + Descripción: Creación con aeronave con un id inexistente
  + Resultado: Falla
  + Efectividad para detectar errores: Alta

**Funcionalidad: Actualizar un tramo**

* Caso de prueba 1:
  + Descripción: Actualización con valores válidos
  + Resultado: Éxito
  + Efectividad para detectar errores: Baja
* Caso de prueba 2 (hacking):
  + Descripción: Actualización de un tramo cuyo vuelo pertenece a otro manager
  + Resultado: Falla
  + Efectividad para detectar errores: Alta
* Caso de prueba 3 (hacking):
  + Descripción: Actualización de un tramo publicado
  + Resultado: Falla
  + Efectividad para detectar errores: Alta
* Caso de prueba 4 (hacking):
  + Descripción: Actualización con aeropuerto de salida con un id inexistente
  + Resultado: Falla
  + Efectividad para detectar errores: Alta
* Caso de prueba 5 (hacking):
  + Descripción: Actualización con aeropuerto de llegada con un id inexistente
  + Resultado: Falla
  + Efectividad para detectar errores: Alta
* Caso de prueba 6 (hacking):
  + Descripción: Actualización con aeronave con un id inexistente
  + Resultado: Falla
  + Efectividad para detectar errores: Alta

**Funcionalidad: Publicar un tramo**

* Caso de prueba 1:
  + Descripción: Publicación con valores válidos
  + Resultado: Éxito
  + Efectividad para detectar errores: Baja
* Caso de prueba 2:
  + Descripción: Publicación con valores inválidos. Entre ellos: Aeronave en uso durante un tramo que se solapa con este, solapamiento con el tramo anterior dentro del mismo vuelo (la llegada del tramo anterior sucede después de que este haya despegado) y solapamiento con el tramo siguiente dentro del mismo vuelo (el despegue del siguiente tramo sucede antes de que este aterrice).
  + Resultado: Éxito
  + Efectividad para detectar errores: Alta
* Caso de prueba 3 (hacking):
  + Descripción: Publicación de un tramo cuyo vuelo pertenece a otro manager
  + Resultado: Falla
  + Efectividad para detectar errores: Alta
* Caso de prueba 4 (hacking):
  + Descripción: Publicación con aeropuerto de salida con un id inexistente
  + Resultado: Falla
  + Efectividad para detectar errores: Alta
* Caso de prueba 5 (hacking):
  + Descripción: Publicación con aeropuerto de llegada con un id inexistente
  + Resultado: Falla
  + Efectividad para detectar errores: Alta
* Caso de prueba 6 (hacking):
  + Descripción: Publicación con aeronave con un id inexistente
  + Resultado: Falla
  + Efectividad para detectar errores: Alta
* Caso de prueba 7 (hacking):
  + Descripción: Petición GET en lugar de POST
  + Resultado: Falla
  + Efectividad para detectar errores: Baja

**Funcionalidad: Eliminar un tramo**

* Caso de prueba 1:
  + Descripción: Eliminación de un tramo. Se probó también la eliminación de reclamaciones y reservas asociadas, y todas las entidades de las que estas dependían. Por esto podemos observar la funcionalidad de creación de *booking records* durante las pruebas de rendimiento, pues fue necesario generar un dato de ejemplo.
  + Resultado: Éxito
  + Efectividad para detectar errores: Alta
* Caso de prueba 2 (hacking):
  + Descripción: Eliminación de un tramo que pertenece a otro manager
  + Resultado: Fallo
  + Efectividad para detectar errores: Alta
* Caso de prueba 3 (hacking):
  + Descripción: Eliminación de un tramo previamente publicado
  + Resultado: Fallo
  + Efectividad para detectar errores: Alta
* Caso de prueba 4 (hacking):
  + Descripción: Eliminación de un tramo inexistente
  + Resultado: Fallo
  + Efectividad para detectar errores: Baja

# Pruebas de rendimiento

Esta sección presenta los resultados del análisis de rendimiento del proyecto.

No ha sido necesario realizar el análisis comparando el código con índices y sin ellos ya que, en este caso, utilizar índices no tendrá impacto de manera significativa en el rendimiento debido a que todas las cláusulas *where* contienen exclusivamente atributos identificativos (ids).

A continuación, se observan los gráficos y tablas que permiten comparar dos equipos durante la ejecución de las pruebas funcionales.

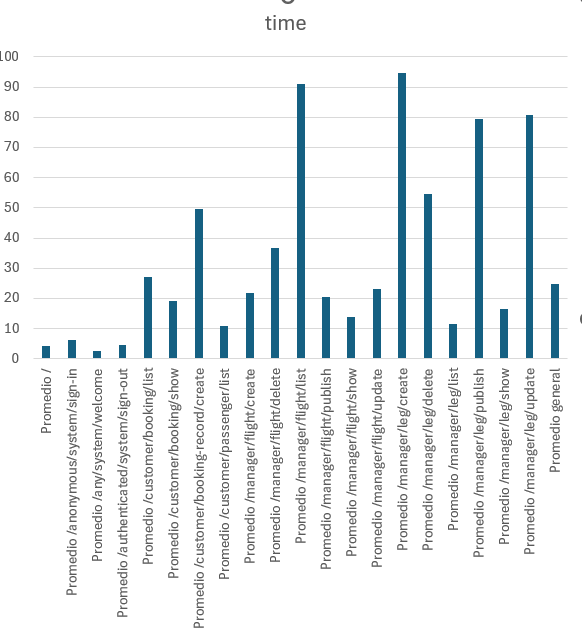


Figura . Gráfica de tiempos de respuesta promedios para Equipo 1.

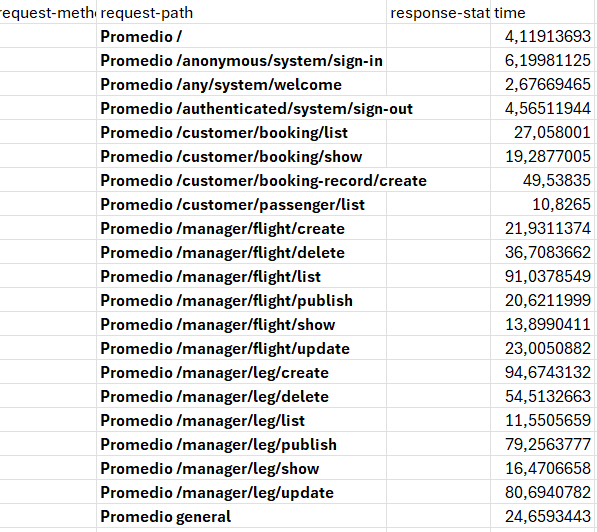


Figura . Promedio de tiempos de respuesta para el Equipo 1.

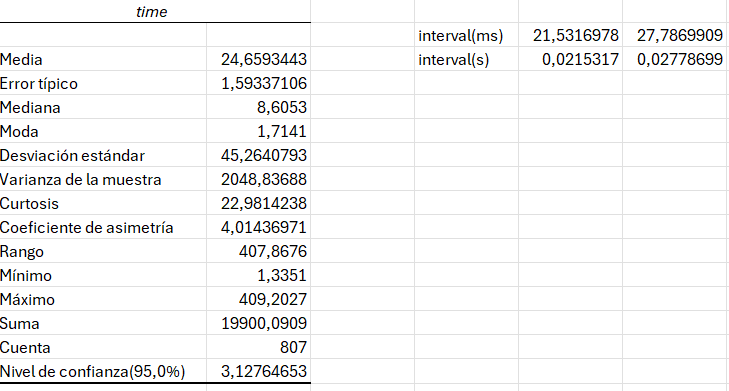


Figura . Intervalo de confianza para el Equipo 1.

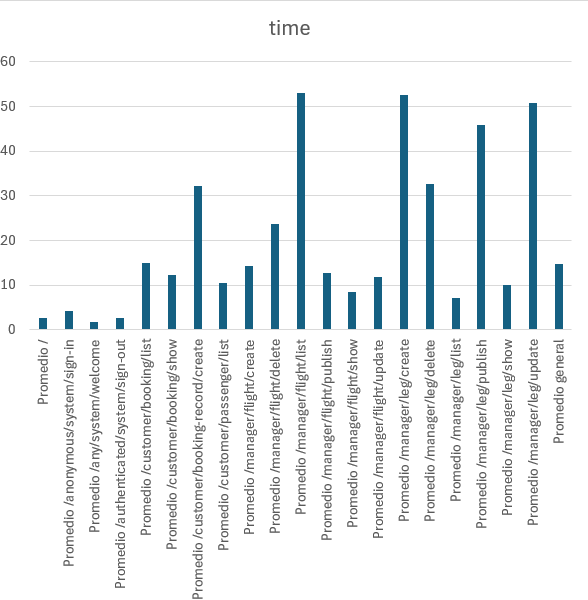


Figura . Gráfica de tiempos de respuesta promedios para Equipo 2.

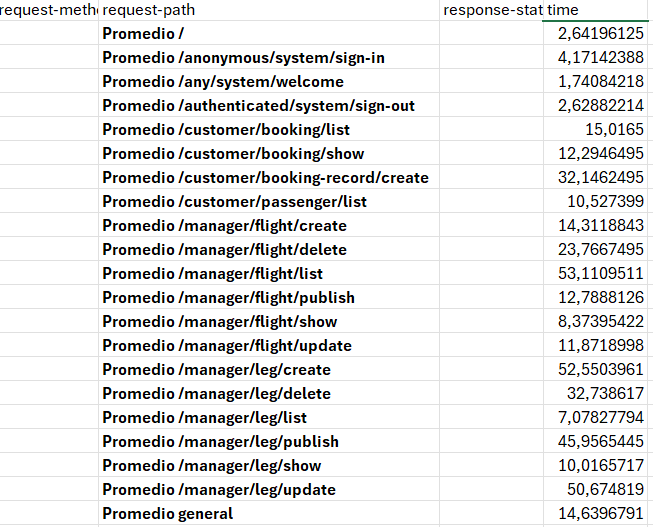


Figura . Promedio de tiempos de respuesta para el Equipo 2.

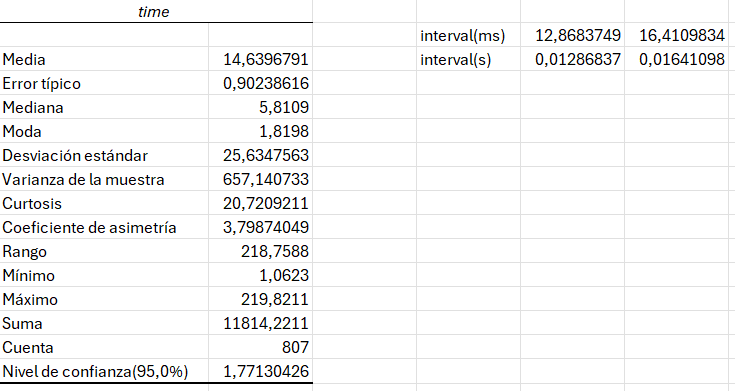


Figura . Intervalo de confianza para el Equipo 2.

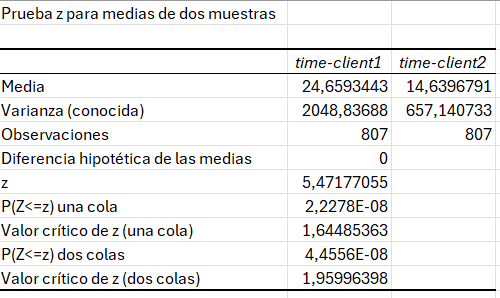


Figura . Prueba z para medias de ambas muestras

**Equipo 1:**

* **Promedio de wall time:** 24,6593443 ms
* **Intervalo de confianza (95%):** 21,5316978 ms – 27,7869909 ms

**Equipo B:**

* **Promedio de wall time:** 14,6396791 ms
* **Intervalo de confianza (95%):** 12,8683749 ms – 16,4109834 ms

**Contraste de hipótesis:**

Dado que nuestro *p-value two-tail* (4,4556\*10-8) se encuenta en el intervalo entre 0 y *alpha* (cuyo valor es 0,05), podemos comparar los promedios de wall-time y rechazamos la hipótesis nula.

24,6593443 ms > 14,6396791 ms

Por tanto, podemos concluir que, por norma general, el equipo 1 es menos potente que el equipo 2.

# Conclusiones

A lo largo del proceso de validación del sistema, se han ejecutado un total de **56 casos de prueba funcionales**, cubriendo tanto escenarios seguros como maliciosos. Estas pruebas han permitido verificar el correcto funcionamiento de las funcionalidades clave asociadas al estudiante #1, así como su robustez ante intentos de acceso indebido. En general, la cobertura de pruebas ha sido adecuada, con aproximadamente un 97% de código probado, y ha contribuido significativamente a la detección y resolución de errores durante el desarrollo.

En cuanto al rendimiento, aunque no se incluyó una evaluación específica sobre el impacto del uso de índices en las consultas, se ha podido realizar una comparación entre dos equipos distintos. Los resultados obtenidos permiten concluir, con un 95% de confianza, que el **Equipo 1 presenta un rendimiento inferior al Equipo 2**, según los tiempos de respuesta medidos en las pruebas funcionales.

En conjunto, los resultados del informe reflejan un sistema funcionalmente estable y seguro, con un rendimiento aceptable, y establecen una base sólida para futuras mejoras y optimizaciones.

# Bibliografía

"En blanco intencionalmente".