

TESTING REPORT STUDENT 1



Grupo: C1.049

Versión 2.0

Miembros: Alberto José Paleteiro López: albpallop1@alum.us.es

Javier Manrique Riquelme: javmanriq@alum.us.es

Sergio Pons López: serponlop@alum.us.es

Samuel Coco Delfa: samcocdel@alum.us.es

Rafael Bermudo Álvarez: rafberalv@alum.us.es

Fecha: 26/05/2025

Repositorio: <https://github.com/DP2-C1-049/D04.git>

Índice

1. Executive summary.....	3
2. Revision Table.....	4
3. Introduction.....	5
4. Contents.....	6
4.1. Functional Testing.....	6
4.1.1. Functional Testing for Flight.....	6
4.1.2. Cobertura Flight.....	8
4.1.3. Functional Testing for Leg.....	8
4.1.4. Cobertura Leg.....	11
4.2. Performance Testing.....	11
4.2.1. Performance Testing en el primer ordenador.....	11
4.2.2. Performance Testing en el segundo ordenador.....	13
4.2.3. 95%-confidence hypothesis contract.....	14
5. Conclusions.....	15
6. Bibliography.....	16

1. Executive Summary

Este informe presenta un resumen ejecutivo del proceso de pruebas realizado sobre la aplicación Acme-ANS con el objetivo de garantizar su calidad, funcionalidad y estabilidad antes de su implementación en el entorno de producción.

El objetivo principal fue validar que el sistema cumple con los requerimientos funcionales y no funcionales establecidos, identificando defectos críticos y áreas de mejora que pudieran afectar la experiencia del usuario o el desempeño del sistema.

2. Revision Table

Revision Number	Date	Description
1	26/05/2025	Primera versión del testing report
2	29/06/2025	Testing report actualizado con los cambios debidos al arreglo del requisito 9

3. Introduction

Con el objetivo de validar los requisitos funcionales definidos por el Student 1, se llevaron a cabo pruebas específicas orientadas a la gestión de las entidades Flight y Leg. Estas entidades forman parte del sistema bajo prueba, y su correcto funcionamiento es esencial para garantizar la integridad del módulo de gestión de vuelos.

El proceso de pruebas fue implementado utilizando el entorno de desarrollo Eclipse, asegurando una integración adecuada con el resto del sistema. Las pruebas desarrolladas se encuentran estructuradas en archivos con extensión .safe y .hack, organizados dentro de los directorios:

- /manager/flight
- /manager/leg

Estos archivos representan los casos de prueba diseñados para verificar el comportamiento esperado del sistema en distintos escenarios funcionales relacionados con la creación, modificación, eliminación y consulta de vuelos(flights) y tramos (legs).

Durante la ejecución de las pruebas, se recogieron trazas detalladas que reflejan la interacción del sistema frente a cada entrada de prueba. Estas trazas fueron posteriormente utilizadas para:

- La **generación de datos de análisis**, con el fin de identificar patrones, errores o comportamientos inesperados.
- La **elaboración de tablas explicativas**, que resumen y presentan de forma clara los resultados obtenidos, facilitando así la interpretación de los datos por parte del equipo de desarrollo y validación.

Esta introducción proporciona el marco necesario para comprender el enfoque adoptado en la verificación de los requisitos funcionales, así como la estructura de los recursos generados durante el proceso.

4. Contents

4.1. Functional Testing

En esta parte se explicarán las pruebas funcionales de las clases Flight y Leg con el propósito de validar el funcionamiento de los requisitos 8 y 9 del Student 1.

4.1.1. Functional Testing for Flight

List Flight(Cobertura 100%)

Caso	Descripción	Resultado esperado	Bugs	Efectividad
Caso 1	Listar los flights como manager	Mostrar la lista de flights del manager	0	Baja
Caso 2	Listar los flights sin ser manager	Error "Access not authorised"	1	Media

Show Flight(Cobertura 100%)

Caso	Descripción	Resultado esperado	Bugs	Efectividad
Caso 1	Mostrar flight	Mostrar el flight perteneciente al manager	0	Baja
Caso 2	Mostrar el flight sin ser su manager	Error "Access not authorised"	1	Media
Caso 3	Mostrar el flight sin ser manager	Error "Access not authorised"	1	Media

Create Flight(Cobertura 100%)

Caso	Descripción	Resultado esperado	Bugs	Efectividad
Caso 1	Crear un flight	Crear el flight	0	Baja
Caso 2	Crear un flight con campos Mandatory vacíos	Mostrar un error "May not be null" en los campos Mandatory	1	Media
Caso 3	Crear un flight con valores no válidos(Strings con más caracteres de lo permitida, coste menor que 0 o mayor que 1000000...)	Mostrar un error en aquellos campos con valores no válidos y por qué no son válidos	1	Media
Caso 4	Cambiar el show?id=X por create?id=X	Error "Access not authorised"	1	Media

Caso 5	Entrar en la URL del servicio sin ser manager	Error "Access not authorised"	1	Media
--------	---	-------------------------------	---	-------

Update Flight(Cobertura 99.5%)

Caso	Descripción	Resultado esperado	Bugs	Efectividad
Caso 1	Actualizar un flight	Actualizar el flight	0	Baja
Caso 2	Actualizar un flight con campos Mandatory vacíos	Mostrar un error "May not be null" en los campos Mandatory	1	Media
Caso 3	Actualizar un flight con valores no válidos(Strings con más caracteres de lo permitida, coste menor que 0 o mayor que 1000000...)	Mostrar un error en aquellos campos con valores no válidos y por qué no son válidos	1	Media
Caso 4	Cambiar el show?id=X por update?id=X	Error "Access not authorised"	1	Media

Publish Flight(Cobertura 99.2%)

Caso	Descripción	Resultado esperado	Bugs	Efectividad
Caso 1	Publicar un flight	Publicar el flight	0	Baja
Caso 2	Publicar un flight con campos Mandatory vacíos	Mostrar un error "May not be null" en los campos Mandatory	1	Media
Caso 3	Publicar un flight con valores no válidos(Strings con más caracteres de lo permitida, coste menor que 0 o mayor que 1000000...)	Mostrar un error en aquellos campos con valores no válidos y por qué no son válidos	1	Media
Caso 4	Cambiar el show?id=X por publish?id=X	Error "Access not authorised"	1	Media

Delete Flight(Cobertura 95.5%)

Caso	Descripción	Resultado esperado	Bugs	Efectividad
Caso 1	Borrar un flight	Borrar el flight	0	Baja
Caso 2	Cambiar el show?id=X por delete?id=X	Error "Access not authorised"	1	Media

4.1.2. Cobertura de Flight

acme.features.manager.flight	98,9 %
> ManagerFlightDeleteService.j	95,5 %
> ManagerFlightPublishService.j	99,2 %
> ManagerFlightUpdateService.j	99,5 %
> ManagerFlightController.java	100,0 %
> ManagerFlightCreateService.j	100,0 %
> ManagerFlightListService.java	100,0 %
> ManagerFlightShowService.ja	100,0 %

4.1.3. Functional Testing for Leg

List Leg(Cobertura 100%)

Caso	Descripción	Resultado esperado	Bugs	Efectividad
Caso 1	Listar los legs como manager	Mostrar la lista de legs del manager	0	Baja
Caso 2	Listar los legs sin ser el manager de su flight	Error "Access not authorised"	0	Baja
Caso 3	Listar los legs de un vuelo inexistente	Error "Access not authorised"	0	Baja

Show Leg(Cobertura 95.1%)

Caso	Descripción	Resultado esperado	Bugs	Efectividad
Caso 1	Mostrar leg	Mostrar el leg perteneciente a un flight del manager	0	Baja
Caso 2	Mostrar el leg sin ser el manager de su flight	Error "Access not authorised"	1	Media
Caso 3	Mostrar el flight sin ser manager	Error "Access not authorised"	0	Baja

Create Leg(Cobertura 97.8%)

Caso	Descripción	Resultado esperado	Bugs	Efectividad
Caso 1	Crear un leg	Crear el leg	0	Baja
Caso 2	Crear un leg con campos Mandatory vacíos	Mostrar un error "May not be null" en los campos Mandatory	0	Baja
Caso 3	Crear un leg con valores no válidos(flight number sin formato, fechas menores y mayores que la mínima y máxima...)	Mostrar un error en aquellos campos con valores no válidos y por que no son válidos	1	Media
Caso 4	Cambiar el show?id=X por create?id=X	Error "Access not authorised"	0	Baja
Caso 5	Intentar meter un id erróneo en los campos airports	Mostrar Invalid value en el campo	0	Baja
Caso 6	Intentar establecer la fecha de departure posterior a la de arrival	Mostrar que la fecha departure debe ser anterior a la arrival	0	Baja

Update Leg(Cobertura 98.5%)

Caso	Descripción	Resultado esperado	Bugs	Efectividad
Caso 1	Actualizar un leg	Actualizar el leg	0	Baja
Caso 2	Actualizar un leg con campos Mandatory vacíos	Mostrar un error "May not be null" en los campos Mandatory	1	Media
Caso 3	Actualizar un leg con valores no válidos(flight number sin formato, fechas menores y mayores que la mínima y máxima...)	Mostrar un error en aquellos campos con valores no válidos y por qué no son válidos	1	Media

Caso 4	Cambiar el show?id=X por update?id=X	Error "Access not authorised"	1	Media
Caso 5	Intentar meter un id erróneo en los campos airports	Mostrar Invalid value en el campo	0	Baja
Caso 6	Intentar establecer la fecha de departure posterior a la de arrival	Mostrar que la fecha departure debe ser anterior a la arrival	0	Baja

Publish Leg(Cobertura 98.8%)

Caso	Descripción	Resultado esperado	Bugs	Efectividad
Caso 1	Publicar un leg	Publicar el leg	0	Baja
Caso 2	Publicar un leg con campos Mandatory vacíos	Mostrar un error "May not be null" en los campos Mandatory	1	Media
Caso 3	Publicar un leg con valores no válidos(flight number sin formato, fechas menores y mayores que la mínima y máxima...)	Mostrar un error en aquellos campos con valores no válidos y por qué no son válidos	1	Media
Caso 4	Cambiar el show?id=X por publish?id=X	Error "Access not authorised"	1	Media
Caso 5	Intentar meter un id erróneo en los campos airports	Mostrar Invalid value en el campo	0	Baja
Caso 6	Intentar establecer la fecha de departure posterior a la de arrival	Mostrar que la fecha departure debe ser anterior a la arrival	0	Baja

Delete Leg(Cobertura 97.7%)

Caso	Descripción	Resultado esperado	Bugs	Efectividad
Caso 1	Borrar un leg	Borrar el leg	0	Baja

Caso 2	Cambiar el show?id=X por delete?id=X	Error "Access not authorised"	1	Media
--------	--------------------------------------	-------------------------------	---	-------

4.1.4. Cobertura de Leg

acme.features.manager.leg	98,0 %
ManagerLegShowService.java	95,1 %
ManagerLegCreateService.java	97,8 %
ManagerLegUpdateService.java	98,5 %
ManagerLegPublishService.java	98,8 %
ManagerLegDeleteService.java	97,7 %
ManagerLegController.java	100,0 %
ManagerLegListService.java	100,0 %

4.2. Performance testing

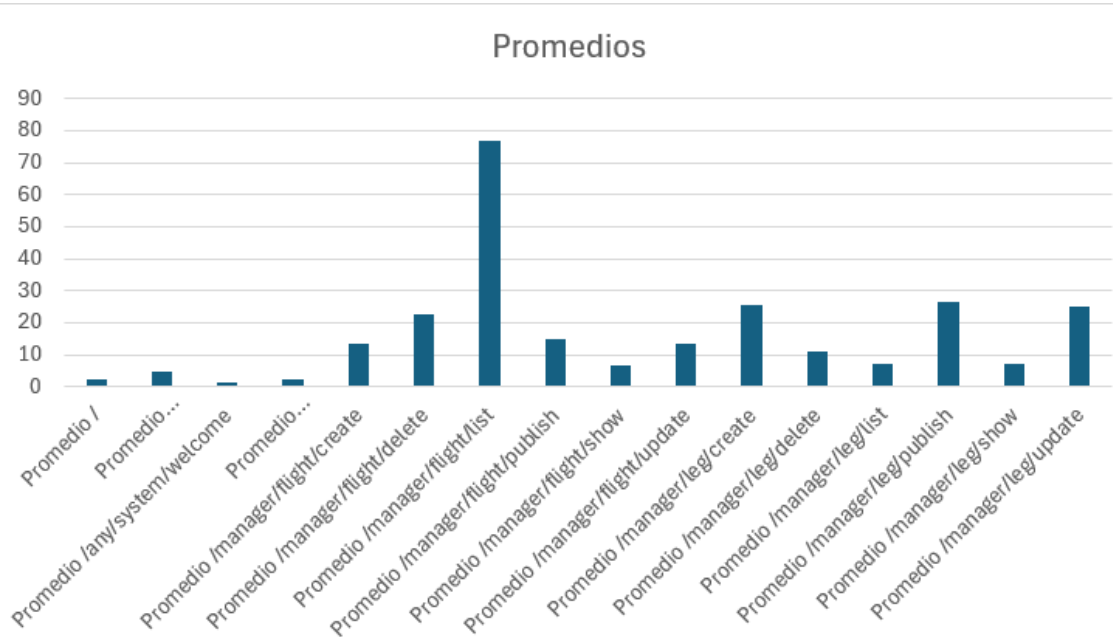
Los casos de prueba mostrados en el apartado anterior fueron ejecutados en eclipse y dejaron un tester.trace. En este apartado vamos a analizar los datos de estos tester.trace antes y después de la refactorización.

4.2.1. Performance testing en el primer ordenador

Antes de realizar la refactorización, un primer ordenador testeo los casos de prueba y dio lugar a los estos resultados:

Promedio /	2.51365366
Promedio /anonymous/syst	4.66079211
Promedio /any/system/welc	1.36945361
Promedio /authenticated/sy	2.16392667
Promedio /manager/flight/c	13.55665
Promedio /manager/flight/c	22.4121833
Promedio /manager/flight/li	76.8829547
Promedio /manager/flight/p	14.7690261
Promedio /manager/flight/s	6.70998496
Promedio /manager/flight/u	13.2445966
Promedio /manager/leg/cre	25.7694574
Promedio /manager/leg/del	10.9464667
Promedio /manager/leg/list	7.044312
Promedio /manager/leg/pul	26.3724976
Promedio /manager/leg/shc	7.04974717
Promedio /manager/leg/upc	25.0272179
Promedio general	16.7091279

Estos datos se pueden apreciar mejor si o vemos en la siguiente gráfica:



Lo que nos devuelve un nivel de confianza del 95% de:

Before	
Media	18.2479254
Error típico	0.99136327
Mediana	7.2731
Moda	1.1801
Desviación estándar	27.1857766
Varianza de la muestra	739.066449
Curtosis	37.3877883
Coefficiente de asimetría	4.42482449
Rango	357.9623
Mínimo	0.6662
Máximo	358.6285
Suma	13722.4399
Cuenta	752
Nivel de confianza(95.0%)	1.9461728

Con los siguientes intervalos:

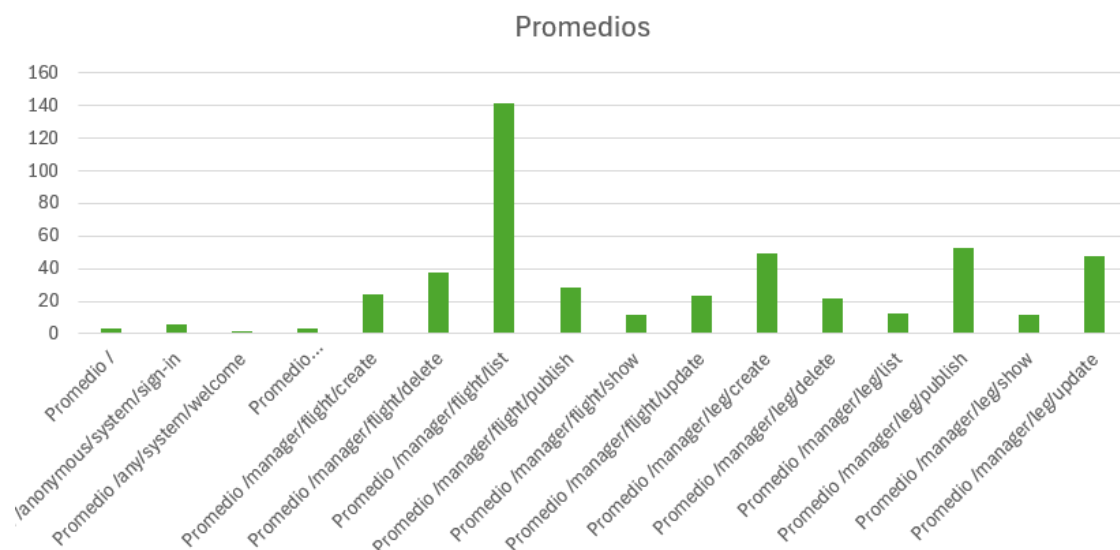
Interval(ms)	16.3017526	20.1940982
Interval(s)	0.01630175	0.0201941

4.2.2. Performance testing en el segundo ordenador

Después de la refactorización, en la que se añadieron unos índices, se volvieron a ejecutar los tests en un segundo ordenador, lo que dejó otros `tester.trace` que nos revela los siguientes resultados:

Promedio /	3.16863298
Promedio /anonymous/syst	5.75881559
Promedio /any/system/welc	2.01438546
Promedio /authenticated/sy	2.9770334
Promedio /manager/flight/c	24.0242115
Promedio /manager/flight/c	37.5888832
Promedio /manager/flight/li	141.344521
Promedio /manager/flight/p	28.7831087
Promedio /manager/flight/s	11.3335174
Promedio /manager/flight/u	23.5616309
Promedio /manager/leg/cre	49.1037127
Promedio /manager/leg/del	22.0691892
Promedio /manager/leg/list	12.318582
Promedio /manager/leg/pub	52.940478
Promedio /manager/leg/shc	11.2855792
Promedio /manager/leg/upc	47.6741152
Promedio general	30.2835481

Que se pueden apreciar mejor en la siguiente gráfica:



Lo cual nos devuelve un nivel de confianza del 95% de:

<i>After</i>	
Media	33.2288043
Error típico	1.68040768
Mediana	12.0017505
Moda	1.733299
Desviación estándar	46.0811786
Varianza de la muestra	2123.47502
Curtosis	7.46885716
Coefficiente de asimetría	2.39502523
Rango	385.0424
Mínimo	1.1288
Máximo	386.1712
Suma	24988.0609
Cuenta	752
Nivel de confianza(95.0%)	3.29885505

Con unos intervalos de:

Intervalo(ms)	29.9299493	36.5276594
Intervalo(s)	0.02992995	0.03652766

4.2.3. 95%-confidence hypothesis contrast

A estos resultados de los casos de prueba antes y después de la refactorización se les sometió a la prueba Z para medias de dos muestras, obteniendo estos resultados para un valor Alpha de 0.05:

Prueba z para medias de dos muestras		
	<i>Before</i>	<i>After</i>
Media	16.7009205	30.2732603
Varianza (conocida)	739.066449	2123.47502
Observaciones	835	835
Diferencia hipotética de las m	0	
z	-7.33030536	
P(Z<=z) una cola	1.148E-13	
Valor crítico de z (una cola)	1.64485363	
Valor crítico de z (dos colas)	2.2959E-13	
Valor crítico de z (dos colas)	1.95996398	

Como se puede observar, el valor de P(Z=z) para dos colas es menor que 0.05 pero el valor de las medias ha crecido por lo que se puede decir que no se ha mejorado el desempeño.

5. Conclusions

Como se ha podido ver en este report, el sistema cumple adecuadamente con los requisitos funcionales del Student 1, soportando también diversos intentos de hacking y valores inválidos. A partir de los casos definidos en los archivos .safe y .hack, y ejecutados en el entorno Eclipse, se obtuvo una cobertura adecuada de los escenarios clave relacionados con la gestión de vuelos y tramos.

Los análisis de datos mostrados en el performance testing nos indican que el sistema funciona dentro de unos márgenes de rendimiento considerables.

Para intentar mejorar este rendimiento, se realizó una refactorización, añadiendo un índice a las entidades utilizadas en estos tests. Sin embargo, los resultados, mostraron que no hubo una mejora en el desempeño de los tests por lo que los índices añadidos se volvieron a quitar.

6. Bibliografía

Intentionally blank.