

Testing Report Student #3
C1.019
Miguel Mir Ceballos

Grupo	C1.019
Repositorio	https://github.com/adolfoborrego/Acme-ANS
Student #1	ID: 29584665H UVUS: XXB5458 Nombre: Borrego González, Adolfo Agustín Roles: Project Manager
Student #2	ID: 77873179D UVUS: SSK0456 Nombre: Martínez Díaz, Ignacio Roles: Analyst
Student #3	ID: 12830191D UVUS: PVL1690 Nombre: Mir Ceballos, Miguel Roles: Operator
Student #4	ID: 52077055H UVUS: TCP2748 Nombre: Sánchez Carmona, Germán Roles: Developer
Student #5	ID: 54794337B UVUS: CFV7375 Nombre: Regidor García, Miguel Roles: Tester

Índice

1. Introducción	2
1.1. Propósito del documento	2
2. Functional Testing	3
2.1. Introducción	3
2.2. Cobertura de las pruebas	4
2.3. Casos de prueba	5
2.4. Conclusiones	7
3. Performance Testing	8
3.1. Introducción	8
3.2. Gráficos de eficiencia medios	9
3.3. Estadísticas descriptivas	10
3.4. Hipótesis y conclusiones	11
4. Historial de versiones	12
5. Bibliografía	13

1. Introducción

1.1. Propósito del documento

El propósito de este documento es presentar de forma estructurada y rigurosa los resultados obtenidos durante el proceso de pruebas del sistema desarrollado por el equipo. El informe tiene como objetivo principal verificar que las funcionalidades implementadas cumplen correctamente con los requisitos definidos y que el sistema ofrece un comportamiento estable, seguro y eficiente tanto a nivel funcional como de rendimiento.

A través de este documento se detalla la ejecución de pruebas funcionales, diferenciando entre versiones inseguras (.hack) y versiones protegidas (.safe), así como el análisis del rendimiento del sistema en diferentes condiciones, incluyendo comparativas estadísticas e intervalos de confianza. Este análisis permite no solo identificar errores o vulnerabilidades, sino también justificar decisiones técnicas tomadas durante el desarrollo para mejorar la calidad final del producto.

2. Functional Testing

2.1. Introducción

En este apartado se documenta la metodología seguida para realizar las pruebas funcionales (functional testing) de las distintas características implementadas. El objetivo principal de este tipo de pruebas es comprobar que las funcionalidades del sistema devuelven los resultados esperados, tanto en condiciones normales como anómalas, garantizando así la calidad del software desde el punto de vista del usuario final.

Siguiendo la metodología formal estudiada, se han diseñado y ejecutado casos de prueba positivos, negativos (.safe) y de hacking (.hack). Cada uno de estos tipos de pruebas ha sido elaborado respetando los principios de repetibilidad, control de datos y cobertura. Se ha prestado especial atención a la verificación de formularios de asignación de vuelos, la visualización de logs de actividad, la gestión de entradas inválidas y la resistencia del sistema ante intentos de uso indebido.

En las siguientes secciones se detallan los casos de prueba realizados, así como la cobertura obtenida para las funcionalidades implementadas por el Student #3 (`flightAssignment` y `activityLog`)

2.2. Cobertura de las pruebas

La siguiente tabla muestra la cobertura alcanzada por los paquetes principales asociados a las funcionalidades desarrolladas para FlightCrewMember. Esta cobertura se ha obtenido a partir de la ejecución de los casos de prueba descritos.

Paquete	Cobertura
flightCrewMember.activityLog	98%
flightCrewMember.flightAssignment	94.3%

A continuación se presenta la cobertura obtenida por servicio dentro del paquete, flightCrewMember.flightAssignment como resultado directo de los casos de prueba realizados sobre la entidad FlightAssignment.

Servicio	Cobertura
FlightAssignmentDeleteCrewDetailService	90.8
FlightAssignmentUpdateService	91.3
FlightAssignmentDeleteService	92.9
FlightAssignmentPublishService	93.6
FlightAssignmentAddService	96.8
FlightAssignmentShowCrewDetailService	96.1
FlightAssignmentShowService	97.1
FlightAssignmentListCompletedService	91.5
FlightAssignmentListPlannedService	95.3
FlightAssignmentListLeadAttendantCrewService	95.9
FlightAssignmentListCrewsService	95.4
FlightAssignmentCreateService	96.8

Esta otra tabla detalla la cobertura alcanzada por los servicios del paquete, flightCrewMember.activityLog derivada de los casos de prueba diseñados para ActivityLog

Servicio	Cobertura
ActivityLogCreateService	95.7
ActivityLogDeleteService	100
ActivityLogListService	95.3

ActivityLogShowService	97.7
ActivityLogUpdateService	100
ActivityLogPublishService	98

2.3. Casos de prueba

A continuación se detallan los casos de prueba realizados sobre la entidad, Flight Assignment desarrollada por los Flight Crew Member, cubriendo las funcionalidades. La tabla incluye tanto pruebas positivas como pruebas de hacking, con su correspondiente descripción y una valoración de su eficacia.

En cada test de prueba, en especial de los de creación y actualización, se han probado todos los valores posibles para cada atributo.

A continuación se indican los casos de prueba de acciones legales, terminación en “.safe”.

Caso de prueba	Descripción	Eficacia
addCrew	se ha testado las distintas variantes de addCrew, campos vacíos, las variantes de cada atributo de un crew. Sólo podía realizarla el Lead Attendant	No se detectaron bugs
create	se ha creado una asignación de vuelo, solo como Lead Attendant, con las diferentes variantes de los atributos con remarks y sin remarks, y con distintos legs	No se detectaron bugs
delete	Se han eliminado los distintos flight Assignments, siempre y cuando estaban en pending.	No se detectaron bugs
deleteCrewDetail	para cada asignación en Pending, se probó que el Lead Attendant pudiera eliminarla.	No se detectaron bugs
list-completed	listado de las asignaciones de vuelo completadas	No se detectaron bugs
list-Crews	cuando un miembro que no es Lead Attendant quiere listar a sus compañeros de tripulación.	No se detectaron bugs
list-LeadAttendantCrews	Cuando un Lead Attendant quiere listar a sus compañeros en su asignación de vuelo. Tiene algunas funcionalidades extra.	No se detectaron bugs
list-planned	cuando un member quiere listar sus asignaciones de vuelo futuras	No se detectaron bugs

publish	prueba publicar una asignación de vuelo que esté en estado pending, y la pone automáticamente en confirmed	No se detectaron bugs No se detectaron bugs
show	Consiste en ver el detalle de cada asignación de vuelo.	No se detectaron bugs
update	consiste en actualizar una asignación de vuelo siempre y cuando este	No se detectaron bugs

A continuación se listan los casos de

prueba relacionados a los casos de prueba de acciones ilegales, relaciones a acciones .hack

Caso de prueba	Descripción	Eficacia
addCrew	Se han aplicado las diferentes acciones, añadir un assignment a un vuelo sin ser Lead Attendant, o un miembro que no existe. Intentar añadir un member cómo Administrator	No se detectaron bugs
create	Se ha intentado crear una asignación con un id predefinido, con un leg no listado, con un duty que no fuese Lead Attendant	No se detectaron bugs
delete	Se ha intentado eliminar asignaciones que estuvieran publicadas, y asignaciones que solo pudiera eliminarlas la propia persona.	No se detectaron bugs
deleteCrewDetail	Se intentó eliminar las asignaciones de los compañeros. estando estas publicadas y confirmadas, y sin ser Lead Attendant, y asignaciones de vuelos que no fueras parte.	No se detectaron bugs
list-completed	Se ha intentado acceder de otro realm	No se detectaron bugs
list-Crews	Se ha intentado acceder de otro realm, y con ids de asignaciones en las que no formaba parte el principal	No se detectaron bugs

list-LeadAttendantsCrews	Se ha intentado acceder de otro realm, y con ids de asignaciones en las que no formaba parte el principal	No se detectaron bugs
list-planned	Se ha intentado acceder de otro realm	No se detectaron bugs
publish	se ha intentado publicar asignaciones que no eran propias y asignaciones ya publicadas o eliminadas.	No se detectaron bugs No se detectaron bugs
show	Consiste en ver las asignaciones de vuelo que no fueran tuyas, desde otro realm o con otro id.	No se detectaron bugs
update	Consiste en editar las asignaciones de vuelo que no fueran tuyas, desde otro realm o con otro id.	No se detectaron bugs

A continuación se listan los casos de prueba de acciones

legales e ilegales de la entidad Activity Log con terminación en “.safe” y “.hack”, ahora se ponen juntos por ser menos.

Caso de prueba	Descripción	Eficacia
list.safe	consiste en listar las activity Logs de un member en una asignación completada concreta	No se detectaron bugs
list.hack	Se ha intentado acceder a asignaciones de otros miembros, o registros en asignaciones en las que no se formaba parte.	No se detectaron bugs
show.safe	Consiste en ver el detalle de un registro de actividad, si eran editables o no.	No se detectaron bugs
show.hack	Consiste en acceder al detalle de registros de actividad de otros miembros.	No se detectaron bugs
create.safe	Consiste en crear registros de vuelo, se ha probado los diferentes valores de cada atributo, valores máximos, mínimos y singulares.	No se detectaron bugs

create.hack	Se intentó crear registros de vuelo con ids preestablecidos y con datos incorrectos.	No se detectaron bugs
update.safe	Consiste en actualizar registros de vuelo, se ha probado los diferentes valores de cada atributo, valores máximos, mínimos y singulares.	No se detectaron bugs
update.hack	Se intentó actualizar registros de vuelo con ids preestablecidos y con datos incorrectos.	No se detectaron bugs
delete.safe	Se borran los registros que estuviesen en modo borrador.	No se detectaron bugs
delete.hack	Se intentó borrar los registros que no eran eliminables y de otros miembros.	No se detectaron bugs
publish.safe	Se publican los registros que estuviesen en modo borrador.	No se detectaron bugs
publish.hack	Se intentó publicar registros que no eran publicables y de otros miembros.	No se detectaron bugs

2.4. Conclusiones

A lo largo del proceso de functional testing, se han diseñado y ejecutado numerosos casos de prueba que abarcan tanto situaciones normales como excepcionales, incluyendo intentos de acceso no autorizado (hacking). Estos casos se han aplicado sobre las funcionalidades desarrolladas para los roles de Flight Crew Member en las entidades Flight Assignment y Activity Log. Las pruebas han demostrado ser eficaces no solo para validar el comportamiento esperado del sistema ante entradas válidas, sino también para identificar errores lógicos en los métodos de autorización y validación.

En particular, se detectaron fallos en la validación de tracking logs durante el proceso de publicación, los cuales fueron corregidos a raíz de los resultados de las pruebas. Además, la cobertura alcanzada como resultado de la ejecución de estos casos ha sido notablemente alta. En concreto, se ha superado el 90 % de cobertura en todos los servicios, y se han alcanzado cifras cercanas al 95 % en la mayoría de ellos. Esto indica que los test realizados han ejercitado prácticamente todas las instrucciones relevantes del código, lo que refuerza la fiabilidad y robustez de las funcionalidades implementadas.

En resumen, el proceso de pruebas funcionales ha sido clave para garantizar la calidad del sistema, permitiendo detectar errores, reforzar los controles de seguridad y confirmar que los requisitos funcionales se han cumplido adecuadamente.

3. Performance Testing

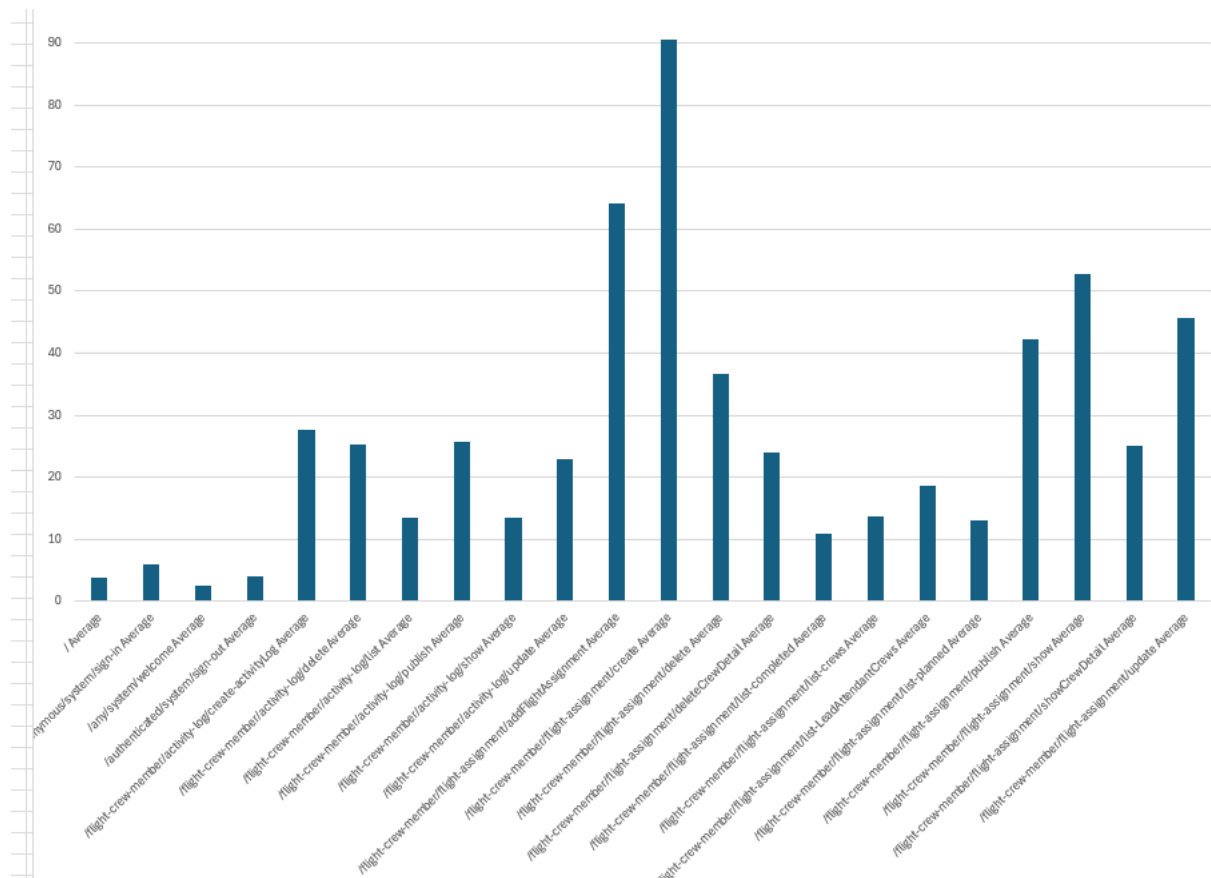
3.1. Introducción

El propósito de este apartado es analizar el rendimiento del sistema mediante pruebas de tipo performance testing. A diferencia de las pruebas funcionales, cuyo objetivo principal es verificar el comportamiento correcto del sistema, las pruebas de rendimiento se centran en medir el tiempo de respuesta de las distintas funcionalidades bajo condiciones controladas. Para llevar a cabo este análisis, se han reutilizado los casos de prueba funcionales ya existentes, reproduciéndose (replay) en dos equipos portátiles diferentes. De este modo, se ha obtenido un conjunto de datos reales de ejecución a partir de los cuales se ha podido evaluar la eficiencia del sistema y comparar el rendimiento relativo de ambos entornos. El análisis se ha realizado siguiendo la metodología propuesta, que incluye el cálculo de intervalos de confianza para el tiempo medio de respuesta y la aplicación de una prueba Z para dos muestras independientes, con el objetivo de determinar si las diferencias observadas entre ambos portátiles son estadísticamente significativas. En las siguientes secciones se presentan los resultados obtenidos, tanto en forma de gráficos de eficiencia media como de análisis estadísticos que permiten extraer conclusiones fundamentadas sobre el rendimiento de la aplicación.

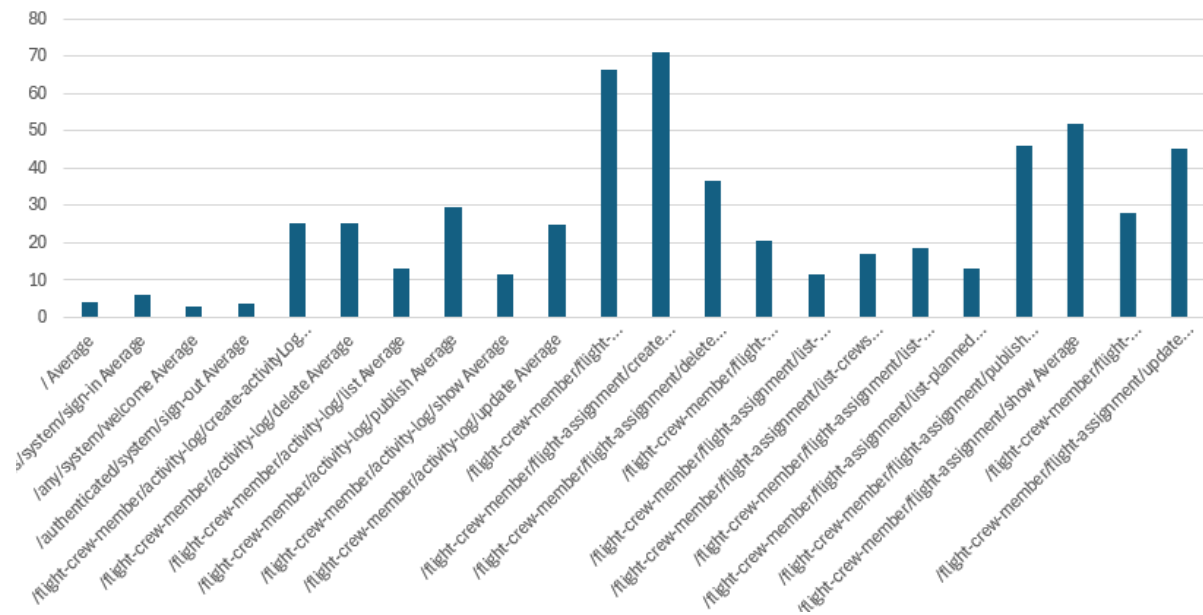
3.2. Gráficos de eficiencia medios

Con el objetivo de analizar visualmente el rendimiento del sistema, se han generado gráficos de eficiencia que muestran el tiempo medio de respuesta por funcionalidad (feature). Estos gráficos permiten identificar de forma clara las funcionalidades que presentan un mayor coste computacional, denominadas MIR (Most Inefficient Request), las cuales podrían convertirse en cuellos de botella en un entorno real.

A continuación, se muestra el gráfico del rendimiento del ordenador 1.



A continuación se muestra el grafico del ordenador 2



3.3. Estadísticas descriptivas

Ordenador 1				Ordenador 2		
Mean	14.77417			Mean	15.0376803	
Standard E	0.653448			Standard E	0.73355818	
Median	7.19265			Median	7.1417	
Mode	2.7525			Mode	2.7232	
Standard D	20.47701			Standard D	22.9874242	
Sample Va	419.308			Sample Va	528.42167	
Kurtosis	10.96522			Kurtosis	25.3006103	
Skewness	2.888588			Skewness	4.04845991	
Range	159.8935			Range	258.4999	
Minimum	1.596601			Minimum	1.614	
Maximum	161.4901			Maximum	260.1139	
Sum	14508.23			Sum	14767.0021	
Count	982			Count	982	
Confidenc	1.282316			Confidenc	1.43952367	
Interval (m	13.49185	16.05648		Interval (m	13.5981567	16.4772
Interval(s)	0.013492	0.016056		Interval(s)	0.01359816	0.016477

Ambos portátiles cumplen con el supuesto de que la media de respuesta se mantiene muy por debajo del segundo (1 s), cumpliendo así el requisito de rendimiento utilizado como referencia.

3.4. Hipótesis y conclusiones

Tras obtener los intervalos de confianza y las estadísticas descriptivas de ambos portátiles, se ha realizado un contraste de hipótesis mediante una prueba Z para dos muestras independientes, con el objetivo de determinar si la diferencia entre los tiempos medios de respuesta es estadísticamente significativa. La prueba Z se ha aplicado sobre los conjuntos de datos recogidos en ordenador 1 y ordenador 2, considerando un nivel de significación del 5 % ($\alpha = 0.05$).

z-Test: Two Sample for Means		
	14.77416688	15.0377
Mean	1252.210141	1297.079
Known Variance	419.308	528.4217
Observations	13	13
Hypothesized Mean	0	
z	-5.255012359	
P(Z<=z) one-tail	7.40073E-08	
z Critical one-tail	1.644853627	
P(Z<=z) two-tail	1.48015E-07	
z Critical two-tail	1.959963985	

Dado que el valor de $P(Z \leq z)$ dos colas se encuentra en el intervalo $(\alpha, 1]$, podemos afirmar que no existen grandes diferencias en el rendimiento entre ambos portátiles.

4. Historial de versiones

Versión	Fecha	Descripción
0.0	25/05/2025	Borrador inicial
1.0	26/05/2025	Correcciones de funcional y performance

5. Bibliografía

Intentionally Blank