

Universidad de Sevilla

Escuela Técnica Superior de Ingeniería Informática



Grado en Ingeniería Informática – Ingeniería del Software

Diseño y Pruebas 2

Curso 2024 – 2025

## Reporte de testing estudiante 1

**Miguel Galán Lerate**

**miggaller@alum.us.es**

Repositorio: <https://github.com/DP2-C1-035/Acme-ANS-C2>

Fecha	Versión
-	v1.0

<b>1. Resumen.....</b>	<b>3</b>
<b>2. Control de versiones.....</b>	<b>3</b>
<b>3. Introducción.....</b>	<b>3</b>
<b>4. Contenidos.....</b>	<b>3</b>
<b>4.1. Testing funcional.....</b>	<b>3</b>
<b>4.2. Pruebas de rendimiento.....</b>	<b>9</b>
<b>5. Conclusión.....</b>	<b>11</b>

## 1. Resumen.

En este documento se recoge toda la información relevante del testing llevado a cabo por el Student 1 en el proyecto DP2-C2-35, detallando el plan seguido a la hora de realizar los

casos de prueba. También se detalla el rendimiento obtenido al repetir los tests sin índices, con índices y en otra máquina distinta y una comparativa estadística sobre el rendimiento en cada caso.

## 2. Control de versiones.

Versión	Cambio	Fecha
1.0.0	Primera versión	04/07/2025

## 3. Introducción.

En el presente documento se describe la metodología de testing que ha usado el Student 1 durante el entregable D04.

El documento contiene un resumen ejecutivo que proporciona una visión clara de los puntos más importantes del documento, así como la información necesaria para entenderlo. Además, el documento cuenta con una lista de versiones como se puede comprobar en la tabla de versiones.

En cuanto al contenido de este, está dividido en dos capítulos, el primero con varios apartados que se corresponden con la cobertura total lograda en el código y casos de prueba para las distintas features implementadas. En el segundo apartado se llevará a cabo un análisis de rendimiento del código realizados en diferentes circunstancias.

## 4. Contenidos.

### 4.1. Testing funcional.

#### Cobertura

- **FLIGHT**

Se adjunta a continuación los resultados de la cobertura de las pruebas realizadas para las features de la entidad Flight.

▼  acme.features.manager.flight	 98,8 %
>  ManagerPublishFlightService.java	 98,6 %
>  ManagerDeleteFlightService.java	 98,1 %
>  ManagerShowFlightService.java	 97,5 %
>  ManagerUpdateFlightService.java	 98,8 %
>  ManagerCreateFlightService.java	 99,5 %
>  ManagerFlightController.java	 100,0 %
>  ManagerListFlightService.java	 100,0 %

Se aprecia que se ha obtenido una cobertura total del código cercana al 99% pudiendo determinar así que se han abarcado casi en su totalidad los casos de pruebas para verificar el correcto funcionamiento de las features de la entidad.

Los casos que no han sido probados se tratan de ramas condicionales añadidas en el código con el fin de prevenir errores inesperados. En ningún caso se trata de falta de casos de prueba positivos ni negativos.

- **LEG**

Se adjunta a continuación los resultados de la cobertura de las pruebas realizadas para las features de la entidad Flight.

▼  acme.features.manager.leg	 97,6 %
>  ManagerLegPublishService.java	 94,0 %
>  ManagerLegUpdateService.java	 98,7 %
>  ManagerLegCreateService.java	 99,0 %
>  ManagerLegDeleteService.java	 99,6 %
>  ManagerLegShowService.java	 99,4 %
>  ManagerLegController.java	 100,0 %
>  ManagerLegListService.java	 100,0 %

De igual manera, se aprecia que se ha obtenido una cobertura total del código cercana al 98% pudiendo determinar así que se han abarcado casi en su totalidad los casos de pruebas para verificar el correcto funcionamiento de las features de la entidad.

### Casos de prueba

#### **Casos de prueba List:**

- **Flight**

Safe	Hack
Para probar el correcto funcionamiento del listado de vuelos, se ha utilizado como usuario de pruebas a manager1 y	Las pruebas de Hack se han realizado como casos negativos de la funcionalidad, tratando de acceder al listado de vuelos desde un usuario

se ha realizado la petición para acceder al listado de vuelos de dicho manager.	anónimo y desde un usuario logueado que no tiene el rol de manager.
---	---

- **Leg**

Safe	Hack
De igual manera, para probar el correcto funcionamiento del listado de tramos de un vuelo, se ha utilizado como usuario de pruebas a manager1 y se ha realizado la petición para acceder al listado de tramos de un vuelo de dicho manager.	Las pruebas de Hack se han realizado como casos negativos de la funcionalidad, tratando de acceder al listado de tramos de un vuelo desde un usuario anónimo y desde un usuario logueado que no tiene el rol de manager.

### Casos de prueba Show:

- **Flight**

Safe	Hack
Para probar la funcionalidad de mostrar un vuelo, se ha usado al usuario manager 1 como usuario de pruebas para acceder al formulario de un vuelo presente en su lista de vuelos.	Las pruebas de Hack se han realizado como casos negativos de la funcionalidad, tratando de acceder a un vuelo desde un usuario anónimo y desde un usuario logueado que no tiene el rol de manager.

- **Leg**

Safe	Hack
Para probar la funcionalidad de mostrar un vuelo, se ha usado al usuario manager 1 como usuario de pruebas para acceder al formulario de un tramo de un vuelo presente en su lista de vuelos.	Las pruebas de Hack se han realizado como casos negativos de la funcionalidad, tratando de acceder a un tramo de un vuelo desde un usuario anónimo y desde un usuario logueado que no tiene el rol de manager. Además

	se probó la confidencialidad de los vuelos intentando acceder a un vuelo de manager 1 logueado como manager 2.
--	--

#### Casos de prueba Create:

- **Flight**

Safe	Hack
Para probar la funcionalidad de crear un vuelo, se ha usado al usuario manager 1 como usuario de pruebas para acceder al formulario de creación de un vuelo. Para probar todos los casos de uso, se introdujeron de manera intencionada datos erróneos en cada uno de los campos 1 a 1 de manera que abarcara todas las posibilidades probando así los casos negativos. Finalmente se introdujeron datos correctos para verificar el correcto funcionamiento de la creación de vuelo.	Las pruebas de Hack se han realizado para probar la seguridad de la funcionalidad. Se ha tratado de acceder al formulario de creación desde un usuario anónimo y desde un usuario con rol distinto a manager.

- **Leg**

Safe	Hack
Para probar la funcionalidad de crear un tramo, se ha seguido la misma metodología aplicada con la entidad Flight.	Las pruebas de Hack se han realizado de igual manera que la entidad Flight. Sin embargo, estas no están presentes en el código debido a posiblemente un error en la configuración de mi entorno de trabajo lo que me permitía grabar correctamente los tests pero al reproducirse se generaban errores sin haber modificado el código(sin dar posibilidad a introducción de nuevos bugs.)

#### Casos de prueba Update:

- **Flight**

Safe	Hack
Para probar la funcionalidad de actualizar un vuelo, se ha seguido la misma metodología aplicada con la creación de la entidad Flight.	Las pruebas de Hack se han realizado de igual manera que en la creación. Además se ha realizado la modificación de campos de solo lectura mediante el uso de herramientas del desarrollador del navegador.

- **Leg**

Safe	Hack
Para probar la funcionalidad de actualizar un vuelo, se ha seguido la misma metodología aplicada con la creación de la entidad Leg.	Las pruebas de Hack se han realizado de igual manera que en la creación. Además se ha realizado la modificación de campos de solo lectura mediante el uso de herramientas del desarrollador del navegador. Sin embargo, estas no están presentes en el código debido a posiblemente un error en la configuración de mi entorno de trabajo lo que me permitía grabar correctamente los tests pero al reproducirse se generaban errores sin haber modificado el código(sin dar posibilidad a introducción de nuevos bugs.)

### Casos de prueba Publish:

- **Flight**

Safe	Hack
Para probar la funcionalidad de publicar un vuelo, se ha seguido la misma metodología aplicada con la creación de la entidad Leg. Además se ha tratado de llegar a situaciones donde por el contexto del requisito no sea	Las pruebas de Hack se han realizado de igual manera que en la creación y actualización. Empleando la modificación de campos de solo lectura mediante el uso de herramientas del desarrollador del navegador.

<p>posible publicar el vuelo a pesar de tener datos correctos en el formulario. Finalmente se introdujeron los campos precisos para permitir la publicación de un vuelo de manera que se verificase el correcto funcionamiento de la funcionalidad.</p>	
---	--

- **Leg**

Safe	Hack
<p>Para probar la funcionalidad de publicar un vuelo, se ha seguido la misma metodología aplicada con la creación de la entidad Leg. Además se ha tratado de llegar a situaciones donde por el contexto del requisito no sea posible publicar el vuelo a pesar de tener datos correctos en el formulario. Finalmente se realizaron</p>	<p>Las pruebas de Hack se han realizado de igual manera que en la creación y actualización. Empleando la modificación de campos de solo lectura mediante el uso de herramientas del desarrollador del navegador. Sin embargo, estas no están presentes en el código debido a posiblemente un error en la configuración de mi entorno de trabajo lo que me permitía grabar correctamente los tests pero al reproducirse se generaban errores sin haber modificado el código(sin dar posibilidad a introducción de nuevos bugs.)</p>

### Casos de prueba Delete:

- **Flight**

Safe	Hack
<p>Para probar la funcionalidad de eliminación de un vuelo simplemente se presionó el botón de borrado presente en el formulario de aquellos vuelos que aún no han sido publicados. Se ha verificado que aquellos vuelos ya</p>	<p>Las pruebas de Hack se han realizado para comprobar que únicamente el usuario asociado a dicho vuelo pueda eliminar un vuelo. Para ello se han realizado las pruebas pertinentes para comprobar dicha funcionalidad. Intentando eliminar vuelos de otro</p>



publicados no se permite la eliminación de los mismos.	manager, logueado como rol diferente a manager y desde un usuario anónimo.
--	--

- **Leg**

Safe	Hack
Para probar la funcionalidad de eliminación de un tramo se ha llevado a cabo la misma lógica que con la entidad Flight.	Las pruebas de Hack se han realizado de igual manera siguiendo la misma lógica que en la entidad Flight.

### Bugs detectados

Durante la realización de los tests se detectaron errores de validaciones para la edición y publicación de las entidades Flight y Leg. Además se detectaron también problemas con la internacionalización de las vistas asociadas a la entidad Leg faltando la traducción de algunos errores de validación en la edición y publicación de la misma.

Respecto a los tests de hack, no se han identificado errores inesperados relacionados con la seguridad del sistema de información. Habiendo pasado así los requisitos mínimos de seguridad, todo ello gracias a implementar toda funcionalidad siguiendo el marco de trabajo del framework lo que ha ayudado a mitigar la mayoría de vulnerabilidades.

## 4.2.Pruebas de rendimiento.

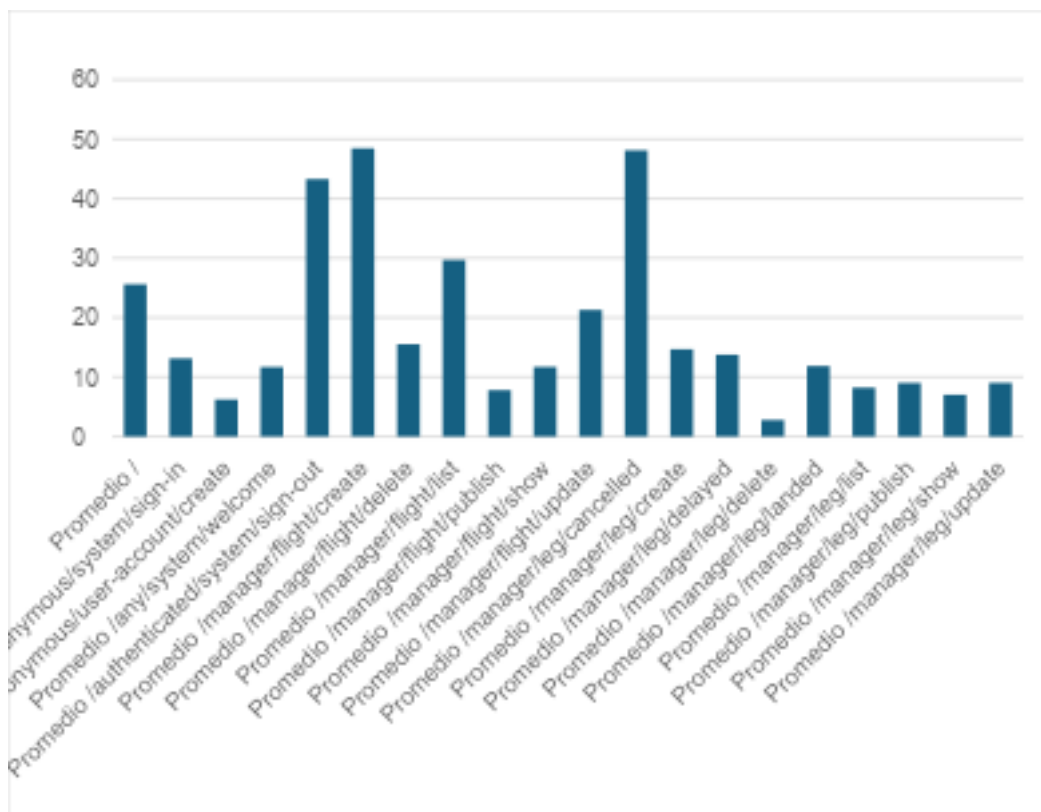
Una vez recopilado todos los datos de prueba se ha realizado una comparación de los tiempos de ejecución con índices y sin índices en las entidades afectadas por los tests.

Se obtuvieron los siguientes resultados:

### **Pruebas de rendimiento sin índices**

Ruta	Tiempo
Promedio /	25.60
Promedio /anonymous/system/sign-in	14.23
Promedio /anonymous/user-account/create	7.22
Promedio /any/system/welcome	11.56
Promedio /authenticated/system/sign-out	43.43
Promedio /manager/flight/create	48.42
Promedio /manager/flight/delete	15.51
Promedio /manager/flight/list	29.68

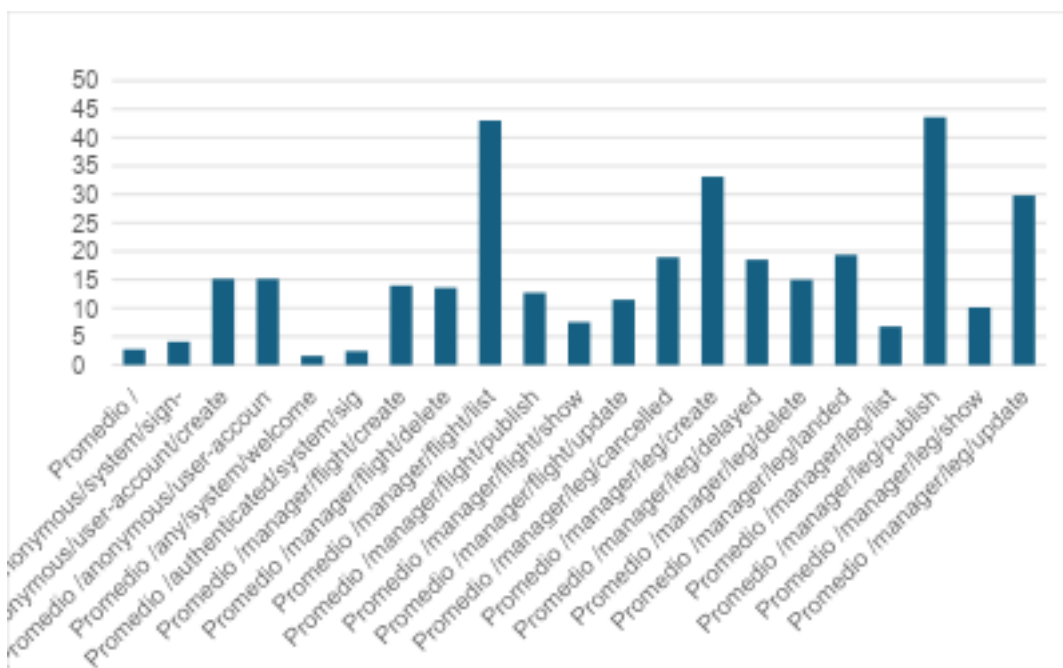
Promedio /manager/flight/publish	7.79
Promedio /manager/flight/show	11.70
Promedio /manager/flight/update	21.29
Promedio /manager/leg/cancelled	48.12
Promedio /manager/leg/create	14.67
Promedio /manager/leg/delayed	13.76
Promedio /manager/leg/delete	2.78
Promedio /manager/leg/landed	11.85
Promedio /manager/leg/list	8.21
Promedio /manager/leg/publish	9.00
Promedio /manager/leg/show	7.02
Promedio /manager/leg/update	9.02
Promedio general	16.74



### Pruebas de rendimiento con índices

Ruta	Tiempo
Promedio /	2.74
Promedio /anonymous/system/sign-in	4.08
/anonymous/user-account/create	15.14
Promedio /anonymous/user-account/create	15.14
Promedio /any/system/welcome	1.60

Promedio /authenticated/system/sign-out	2.43
Promedio /manager/flight/create	13.97
Promedio /manager/flight/delete	13.58
Promedio /manager/flight/list	42.94
Promedio /manager/flight/publish	12.70
Promedio /manager/flight/show	7.53
Promedio /manager/flight/update	11.48
Promedio /manager/leg/cancelled	18.94
Promedio /manager/leg/create	33.11
Promedio /manager/leg/delayed	18.52
Promedio /manager/leg/delete	15.00
Promedio /manager/leg/landed	19.37
Promedio /manager/leg/list	6.77
Promedio /manager/leg/publish	43.64
Promedio /manager/leg/show	10.13
Promedio /manager/leg/update	29.82
Promedio general	15.75



## 5. Conclusión.

A lo largo de este informe se ha detallado de manera exhaustiva el proceso de testing realizado sobre las funcionalidades implementadas en el sistema, abarcando tanto pruebas funcionales como de rendimiento. Se han desarrollado y ejecutado más de 100 casos de prueba que cubren una amplia variedad de escenarios, incluyendo pruebas positivas,

negativas y de hacking, lo cual ha permitido alcanzar un alto grado de cobertura y fiabilidad en las features de flight y leg.

Además, el análisis de rendimiento ha ofrecido información valiosa sobre el impacto de los índices en el sistema y sobre las diferencias entre distintos entornos de ejecución. El uso riguroso de test cases bien diseñados ha sido clave para detectar errores lógicos, evitar excepciones como null pointer y refinar las validaciones del sistema. En conjunto, este trabajo ha contribuido de forma decisiva a asegurar la calidad, estabilidad y robustez del sistema desarrollado.