TESTING REPORT

Acme ANS

DELIVERABLE 4

DESING AND TESTING 2

2024-2025

|  |  |
| --- | --- |
| Date | Version |
| 05/26/2025 | V1.0 |

|  |  |
| --- | --- |
| Group: C1.053 | |
| Members | Email |
| Sánchez Márquez, Julia | [julsanmar2@alum.us.es](mailto:javgarrod5@alum.us.es) |
| García Rodríguez, Javier | [javgarrod5@alum.us.es](mailto:javgarrod5@alum.us.es) |
| González Ortiz, Miguel | [miggonort1@alum.us.es](mailto:miggonort1@alum.us.es) |
| Palomo García, Miguel | [migpalgar1@alum.us.es](mailto:migpalgar1@alum.us.es) |
| Periáñez Franco, Luis Javier | [luiperfra1@alum.us.es](mailto:luiperfra1@alum.us.es) |

GitHub repository: <https://github.com/miggonort1/Acme-ANS-D04>

TABLE OF CONTENTS

Contenido

[Executive Summary 3](#_Toc199176462)

[Revision Table 3](#_Toc199176463)

[Introduction 3](#_Toc199176464)

[Contents 3](#_Toc199176465)

[Conclusion 9](#_Toc199176466)

[Bibliography 9](#_Toc199176467)

# Executive Summary

This report contains the information obtained through the execution of functional and performance tests. In order to show that the indicated methodology has been followed and the conclusions we have reached after the tests.

# Revision Table

|  |  |  |
| --- | --- | --- |
| Revision Number | Date | Description |
| v1r0 | 05/26/2025 | First version |

# Introduction

The content of this report is divided into two parts, one with the functional testing analysis and the other with the performance testing analysis.

# Contents

**Functional testing**

The tests have been performed following the methodology proposed in the theory slides, obtaining as much coverage as possible.

**Imagen que contiene Texto

El contenido generado por IA puede ser incorrecto.Flight**

* List

The test of the list functionality consisted in verifying that a manager can correctly view all flights associated to his account. It was verified that only flights belonging to the authenticated manager were accessible, and that the functionality worked correctly even when the user had no flights registered. For the hacking, an attempt was made to access flights from other managers, without success, as they did not appear in the list. The coverage obtained was 100%.

* Create

The test of the create functionality consisted of generating flights with all possible combinations of attributes, including invalid values to verify that the appropriate error messages were displayed. It was also verified that the flight was created in draft mode by default and that it was correctly associated to the authenticated manager. For the hacking, the creation was attempted without authentication and with a user from another role, without authorization in both cases. The coverage obtained was 89.8%.

* Delete

The test of the delete functionality consisted of deleting flights that were in draft status. We tested the deletion of both flights with and without associated legs, verifying that it was not allowed to delete a flight with assigned legs, and that the legs were correctly deleted together with the flight when appropriate. As for hacking, attempts were made to delete flights not owned, published or non-existent, and all attempts were blocked correctly. The coverage obtained was 74.7%.

* Publish

The test of the publish functionality consisted of checking that a flight can be published correctly as long as it fulfills the necessary conditions: having assigned legs, that all legs are published, and that if there is no auto-transfer there are no stopovers. Tests were run in violation of each of these rules to verify that the corresponding errors were thrown. For the hacking, an attempt was made to publish other flights or flights already published, without success. The coverage obtained was 89.8%.

* Show

The test of the show functionality consisted of viewing the details of a specific flight. It was verified that the manager could only access flights owned by him and that all flight attributes were returned correctly. For the hack, unauthorized access to flights owned by other managers and to non-existent flights was attempted, verifying that in all cases access was prevented. The coverage obtained was 100%.

* Update

The test of the update functionality consisted of modifying the attributes of a flight in draft state. Tests were performed with valid and invalid values to verify that they were validated correctly before saving the changes. It was also tested that it was not allowed to modify flights already published or from other managers. For hacking, attempts were made to update without permissions or with an unauthenticated user, and the system blocked these attempts. The coverage obtained was 69.2%.

**Leg**

**Pantalla de computadora con fondo negro

El contenido generado por IA puede ser incorrecto.**

* List

The test of the list functionality consisted in verifying that a manager can correctly view the list of legs associated to one of his flights. It was verified that only managers with access to the flight can consult the list and that this functionality cannot be accessed from other roles or without being authenticated. For the hacking, an attempt was made to list legs belonging to flights belonging to other managers or non-existing flights. The coverage obtained was 99.2%.

* Create

The test of the create functionality consisted in the creation of new legs associated to a flight in draft mode. Multiple combinations of valid and invalid attributes were tested (past dates, same airports, legs not connected with the previous ones, non-existent aircraft or aircraft not belonging to the manager). For the hacking, an attempt was made to create a leg without being authenticated, or with a manager that did not have permissions on the flight in question. The coverage obtained was 95.4%.

* Delete

The test of the delete functionality consisted of verifying that a manager can successfully delete a leg while the flight is in draft mode. The standard delete case was tested, as well as attempts to delete legs that are no longer in draft. For the hacking, attempts were made to delete from other roles, from a manager that did not own the flight, and legs that did not exist. The coverage obtained was 52.8%, indicating that there is still room to cover more scenarios, especially negative cases and boundary conditions.

* Publish

The test of the publish functionality consisted of checking that a leg in draft mode can be published correctly by its manager as long as the flight to which it belongs is also in draft mode. The operation changes the draftMode state to false. For the hacking, an attempt was made to publish legs from unauthorized users, or in invalid states (already published or with published flights). The coverage obtained was 54.5%, so it would be advisable to add more tests, especially for anomalous or invalid situations.

* Show

The test of the show functionality consisted of viewing the details of a specific leg by the owner manager. The displayed data was tested, including the drop-down options for airports, status and aircraft. For the hack, access was attempted with a non-owner manager, unauthenticated users, and non-existent leg IDs. The coverage obtained was 99.4%.

**Performance testing**

During the performance analysis, the performance before and after the indexes were implemented was analyzed.

* Analysing performance

Average of the times obtained before the implementation of indexes:

Interfaz de usuario gráfica, Gráfico

El contenido generado por IA puede ser incorrecto.

As can be seen, the grand average is 13.46 milliseconds before the implementation of indexes. It can also be seen that the methods with the highest average milliseconds are manager/leg/create and manager/leg/publish.

Average of the times obtained after the implementation of indexes:

Gráfico

El contenido generado por IA puede ser incorrecto.

As can be seen, the grand average is 9.42 milliseconds after the implementation of indexes. After index addition, the methods with the highest average milliseconds are manager/leg/create and manager/flight/delete.

Comparison of the analyses obtained before and after the implementation of the indexes:

Interfaz de usuario gráfica, Aplicación, Tabla, Excel

El contenido generado por IA puede ser incorrecto.

As can be seen both intervals meets our 1-second sample requirement comfortably.

Z-test analysis:

Tabla

El contenido generado por IA puede ser incorrecto.

After the z-test the p-value (Valor critico de z (dos colas) ) is in the range ([0.00, α), where alpha = 0.05.

* Profiling software

Tabla

El contenido generado por IA puede ser incorrecto.

This was the result obtained after profiling software. As can be seen, the execution time of the bind method of the project publish is notably superior to the rest of the methods. However, its self time is 0 so it’s not that method that consumes too much time, but the method that it invokes.

* Interfaz de usuario gráfica

  El contenido generado por IA puede ser incorrecto.Profiling hardware

This was the result obtained after profiling hardware.

The memory of this computer is being moderately used, but it is not a clear bottleneck, the CPU and the network are little used so they are far from a bottleneck as well as the other components.

# Conclusion

Everything went as expected.

# Bibliography

Intentionally blank