Workgroup C1.067

Testing report

25/05/2025

Aarón Jesús Acuña Bellido - [aaracubel@alum.us.es](mailto:aaracubel@alum.us.es)

Javier Arellano López – [javarelop@alum.us.es](mailto:javarelop@alum.us.es)

Raúl Calero Capote – [raucalcap@alum.us.es](mailto:raucalcap@alum.us.es)

Tudor Cristian Lacatus Cosma – [ionlac@alum.us.es](mailto:ionlac@alum.us.es)

Miguel Prado Jiménez – migprajim@alum.us.es

https://github.com/javiarellanoo/Acme-ANS-D04

Table of Contents

[EXECUTIVE SUMMARY: 2](#_Toc199140228)

[REVISION TABLE 3](#_Toc199140229)

[INTRODUCTION 4](#_Toc199140230)

[FUNCTIONAL TESTING 5](#_Toc199140231)

[Operations by administrators on airports 5](#_Toc199140232)

[PERFORMANCE TESTING 7](#_Toc199140233)

[Mean Confidence Interval 8](#_Toc199140234)

[Contrasting information 9](#_Toc199140235)

[CONCLUSIONS 11](#_Toc199140236)

[BIBLIOGRAPHY 12](#_Toc199140237)

# EXECUTIVE SUMMARY:

This document offers a thorough and complete overview of the testing procedures used to validate the functionality and performance of features related to group requirements. The primary goal is to ensure complete security within the application, in compliance with "Ley Orgánica 3/2018."

The functional testing section details the test procedures conducted for each feature available to administrators regarding operations on airports. This testing involved recording and reproducing a variety of legal and hacking scenarios. Particular attention was paid to edge cases within established attribute ranges, input validation, and access control.

The final section of this report focuses on performance testing. By calculating performance across two distinct hardware environments, this report establishes the necessary confidence intervals and compares execution times to identify the more powerful machine.

# REVISION TABLE

|  |  |  |
| --- | --- | --- |
| Revision Number | Date | Description |
| 1.0 | 25/05/2025 | Initial version of the document |
|  |  |  |
|  |  |  |

# INTRODUCTION

This document aims to present an analysis of the testing methodology applied to the application. The content of this document is organized into two primary sections.

The initial section is dedicated to the functional testing process. It will encompass a comprehensive listing of all implemented test cases, categorized by feature. For every test case, a concise explanation of the executed tests will be given, along with an indication of any defects discovered.

The subsequent and final section will cover performance testing. This includes charts illustrating performance metrics, as well as a 95% confidence interval for the time required by the project to process the requests from the functional tests across two distinct hardware configurations. Ultimately, this section will conclude with the identification of the superior performing machine.

# FUNCTIONAL TESTING

## Operations by administrators on airports

Create

Safe Scenarios:

* We submitted the creation form with all null values to ensure no unexpected exceptions were generated.
  + Detection of bugs: None were detected.
* For each form attribute, we comprehensively tested a range of both valid and invalid data sourced from the "Sample-Data" file.
  + Detection of bugs: None were detected.
* A form with entirely valid data was submitted to confirm the correct creation of the flight.
  + Detection of bugs: None were detected.

Hacking Scenarios:

* The feature was requested using alternative realms, and an authorization error was correctly returned.
  + Detection of bugs: None were detected.

List

Safe Scenarios:

* We accessed the list of airports, systematically reviewing every page to confirm accurate rendering of entries.
  + Detection of bugs: None were detected.

Hacking Scenarios:

* Attempts to access the airport listing feature from unauthorized realms (e.g., unauthenticated users or other realms) successfully triggered an authorization exception.
  + Detection of bugs: None were detected.

Show

Safe Scenarios:

* The "show" feature was requested for all airports within the sample dataset. All elements were verified to render correctly.
  + Detection of bugs: None were detected.

Hacking Scenarios:

* The feature was requested with a realm different from the administrator (e.g., a customer or an unauthenticated user). An authorization exception was successfully thrown.
  + Detection of bugs: None were detected.
* The feature was requested for an airport that did not exist. An authorization exception was successfully thrown.
  + Detection of bugs: None were detected.

Update

Safe Scenarios:

* An empty form was submitted to verify that no exceptions were thrown and that errors were correctly reported for the relevant attributes.
  + Detection of bugs: None were detected.
* For each form attribute, a comprehensive range of both invalid and valid data sourced from the "Sample-Data" file was considered and tested.
  + Detection of bugs: None were detected.
* A form containing all valid data was submitted to confirm the successful update of the airport.
  + Detection of bugs: None were detected.

Hacking Scenarios:

* The feature was requested by a realm that was not an administrator (e.g., a customer).
  + Detection of bugs: None were detected.
* The feature was requested for an airport that did not exist (e.g., -1 or 9999).
  + Detection of bugs: None were detected.

# PERFORMANCE TESTING

In this final section, we will evaluate the system's performance by measuring the elapsed time required to complete requests during functional testing. The aim is to determine the responsiveness of the application under realistic conditions and identify which device delivers superior performance.

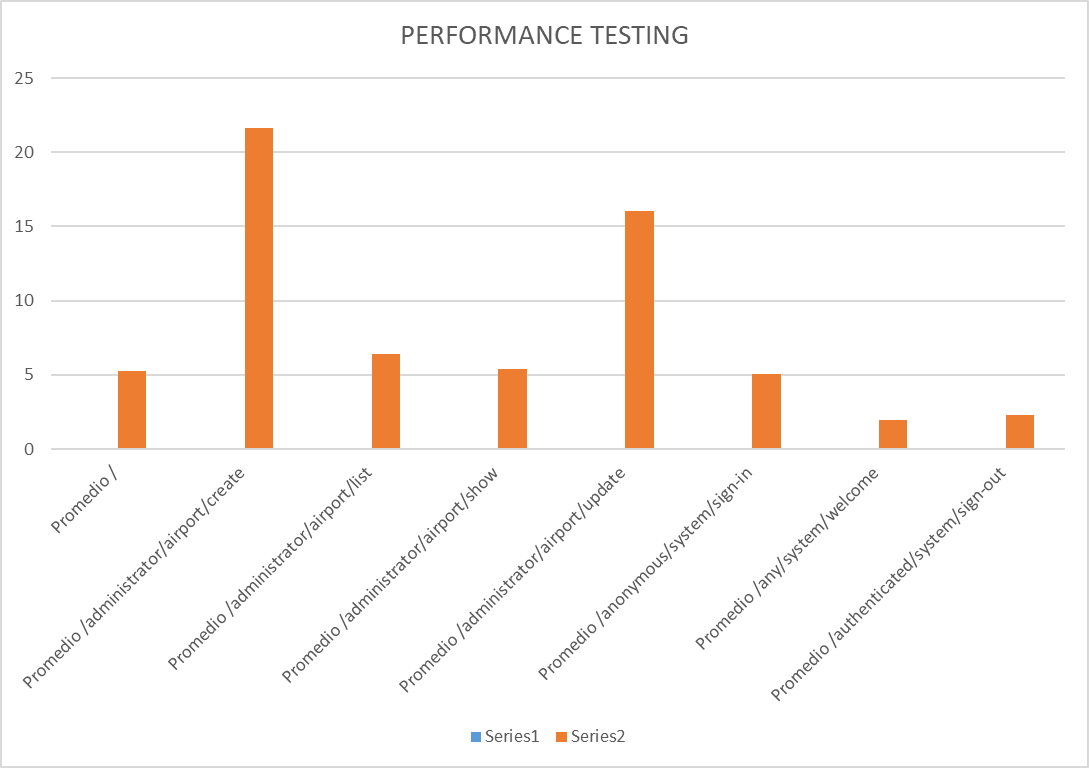
To obtain the necessary data, we will execute the complete suite of functional tests related to the manager-specific features. The tests will be conducted on the following machines:

* HP Victus: 16 GB RAM, 1 TB SSD
* Lenovo Ideapad 5: 32 GB RAM, 512 GB SSD

## Mean Confidence Interval

After cleaning up and treating the data obtained from the trace file that is generated because of replaying tests, we have performed an analysis using the procedure explained in the subject to obtain an average time for each of the requests executed.

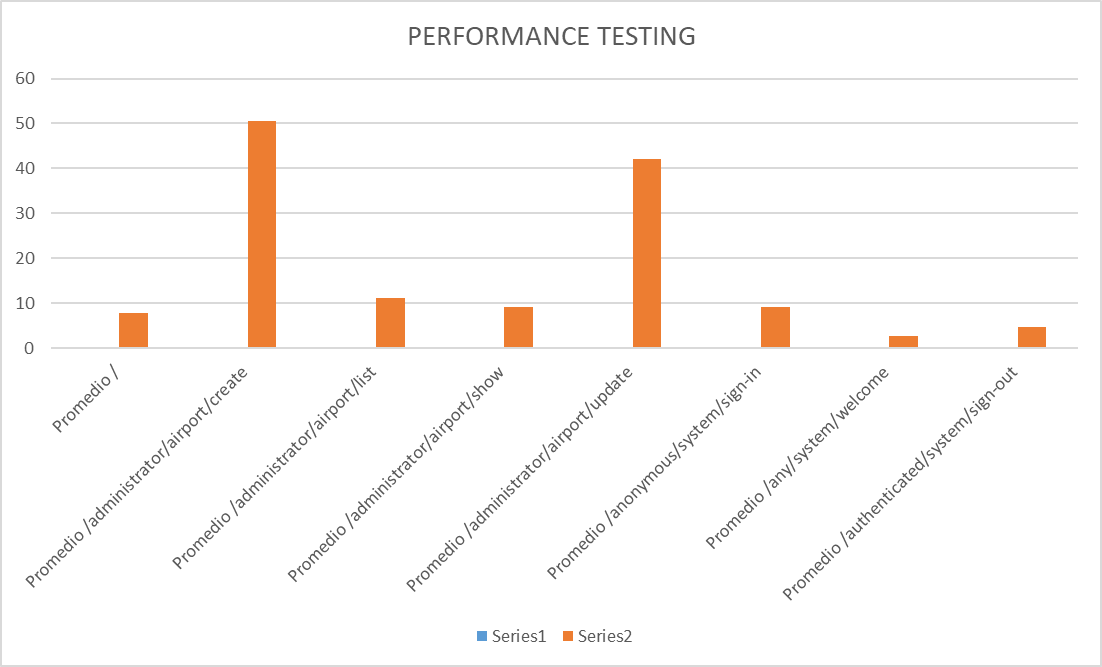
#### HP Victus



For the first computer, we obtained a grand average of 10.41 ms. As can be appreciated, we see that the MIR is the airport create, whose average is of 21.66ms.

Using the data analyzer extension from Excel, we obtained that the amplitude of the confidence interval at 95% is 0.88 ms. By removing and adding this value from the average, we obtain the confidence interval: [0.0095 s – 0.0112 s]

#### Lenovo Ideapad 5

When doing the same with the Lenovo Ideapad 5, we obtained a similar graph. In this case, the grand average stands at 23.21 ms. The apparent MIR is the airport create feature as in the previous, with an average of 50.53 ms.

Using the data analyzer, we get that the amplitude of the confidence interval at 95% is 2.19 ms. We would obtain the following confidence interval: [0.021 s – 0.025s]

## Contrasting information

Given the previous results, we can induce that the computer with the best performance is the HP Victus one, as in general, all its average times are lower. Let us use the Z-Test with alpha 0.05 to verify our hypothesis.  
Once the Z-Test has been conducted, the results are the following:



As it can be clearly seen, the first computer is way faster than the second one, as we initially supposed.

# CONCLUSIONS

This document serves as a formal record of the results obtained from the execution of test cases, with the objective of enabling future reference. The recorded analysis offers a detailed evaluation of each test case, aiming to detect potential bugs in the code and to support informed decision-making to improve the system and ensure protection against possible hacking attempts.

Each test case was recorded systematically, grouped by feature and the bugs found, if any, have been reported. This serves as a transparent reference of the testing process.

Through the analysis of the performance on the two different computers, we have demonstrated that the HP Victus computer has consistently obtained a better performance. As expected, both systems showed similar behavior in terms of request distribution, obtaining the same MIR.

This testing report provides concise descriptions of the tests performed, in hopes that they serve as solid foundation for quality assurance in the near future.

# BIBLIOGRAPHY

Intentionally blank