GRUPAL 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 |
| García Rodríguez, Javier | javgarrod5@alum.us.es |
| González Ortiz, Miguel | miggonort1@alum.us.es |
| Palomo García, Miguel | migpalgar1@alum.us.es |
| Periáñez Franco, Luis Javier | luiperfra1@alum.us.es |

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

TABLE OF CONTENTS

Contenido

[Executive summary 4](#_Toc199191541)

[This report presents the results obtained from the execution of both functional and performance tests. Its purpose is to demonstrate that the established methodology has been applied correctly and to explain the conclusions drawn from the testing process. 4](#_Toc199191542)

[REVISION TABLE 5](#_Toc199191543)

[INTRODUCTION 6](#_Toc199191544)

[This report is structured in two sections: one focused on the analysis of functional testing, and the other on performance testing. 6](#_Toc199191545)

[Throughout the testing process, it was necessary to modify the authorization logic of a method that was not properly implemented. Additionally, following the performance tests, I applied indexing improvements based on the recommendations provided in the theoretical materials. 6](#_Toc199191546)

[CONTENTS 6](#_Toc199191547)

[​​ FUNCTIONAL TESTING 6](#_Toc199191548)

[PERFORMANCE TESTING 7](#_Toc199191549)

[CONCLUSIONS 7](#_Toc199191550)

[Everything went as expected. 7](#_Toc199191551)

[BIBLIOGRAPHY 7](#_Toc199191552)

# 

## Executive summary

# This report presents the results obtained from the execution of both functional and performance tests. Its purpose is to demonstrate that the established methodology has been applied correctly and to explain the conclusions drawn from the testing process.

## REVISION TABLE

|  |  |  |
| --- | --- | --- |
| Revision Number | Date | Description |
| V1.0 | 05/26/2025 | First Version |

# 

## INTRODUCTION

# This report is structured in two sections: one focused on the analysis of functional testing, and the other on performance testing.

# Throughout the testing process, it was necessary to modify the authorization logic of a method that was not properly implemented. Additionally, following the performance tests, I applied indexing improvements based on the recommendations provided in the theoretical materials.

## CONTENTS

### ​​ FUNCTIONAL TESTING

The tests were conducted following the methodology outlined in the theory slides, aiming to achieve the highest possible coverage.

Full (100%) coverage was not reached, primarily because lines such as assert object != null; are never triggered by a null value. Despite this, it is considered good practice to retain these assertions, as recommended.

Moreover, certain lines of code were not fully covered due to representing scenarios that cannot occur under normal execution. These lines have been deliberately preserved, and further details about them will be provide d where relevant.

**Airport**

Interfaz de usuario gráfica

El contenido generado por IA puede ser incorrecto.

* List

The test of the list-completed functionality consisted of viewing the list of airports. For the hacking, an attempt was made to access this list without having sufficient permissions, in this case trying to log in a non-logged in user, a correct realm but wrong user and an incorrect realm. The coverage obtained was 100%.

* **Show**

The test of the show functionality consisted of viewing the details of an airport. For the hacking, we tried to access this list without having sufficient permissions, In this case trying to log in with a user of another role and make a show of an airport, as well as trying to make a show of an airport with an id that does not exist. The coverage obtained was 100%.

* **Create**

The test of the create functionality consisted in creating airports with all possible variations in their attributes including not allowed values to check that the appropriate error message is triggered. For the hacking we tried to create with a non-logged user, create above an id that already exists, among others. The coverage obtained was 100.0%.

* **Update**

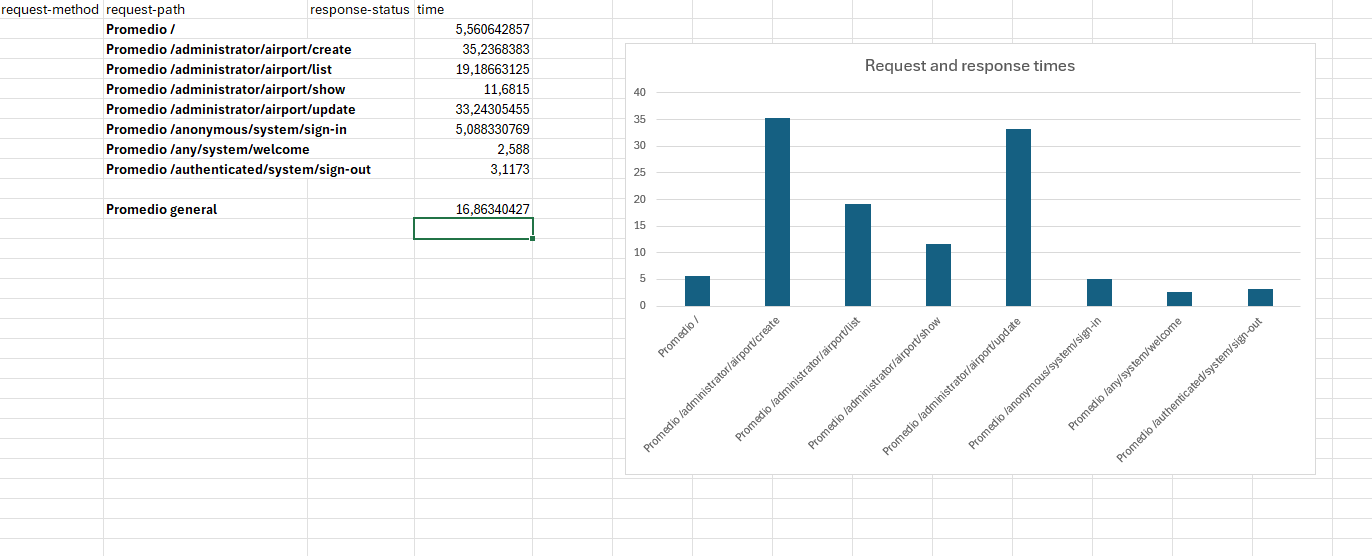
The test of the update functionality consisted in updating airports with all possible variations in their attributes including not allowed values to check that the appropriate error message is triggered. For the hacking we tried to update with a non-logged user, also a non-existing airport and it was also tested to try to update not as an administrator. The coverage obtained was 100%.

### PERFORMANCE TESTING

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

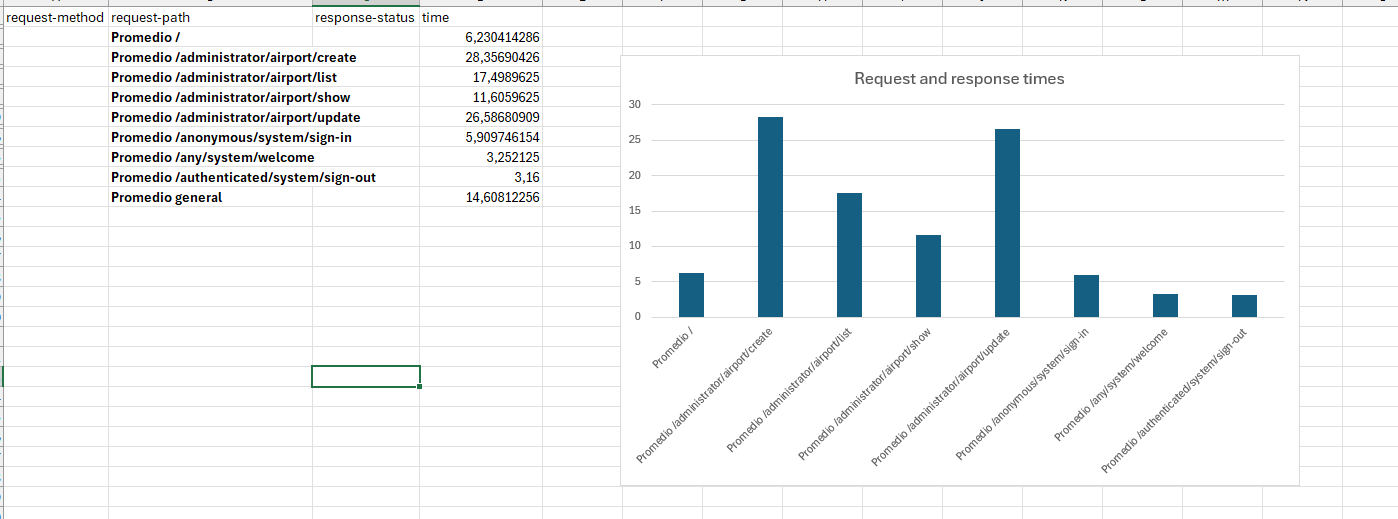
* Analysing performance

Average of the times obtained before the implementation of indexes:

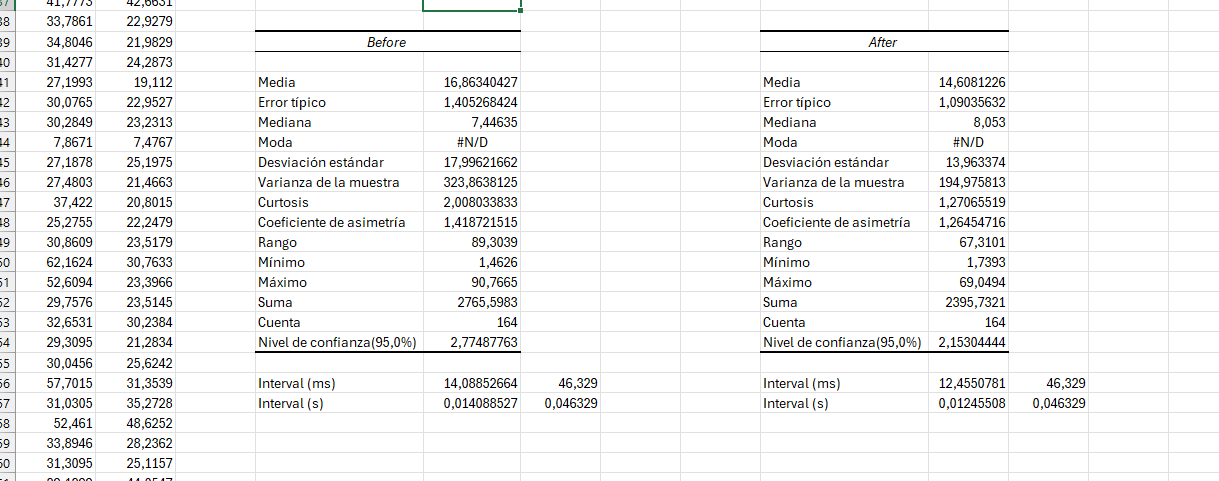


As can be seen, the grand average is 16.86 milliseconds before the implementation of indexes. It can also be seen that the methods with the highest average milliseconds are administrator/airrport/create and administrator/airport/update.

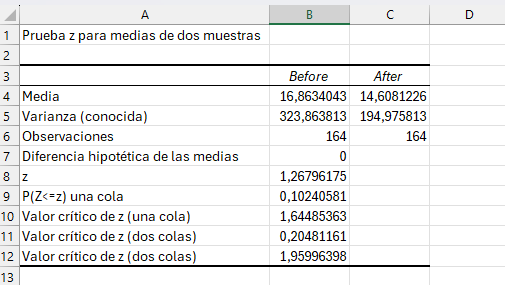
Average of the times obtained after the implementation of indexes:

As can be seen, the grand average is 14.60 milliseconds after the implementation of indexes. After index addition, the methods with the highest average milliseconds are still administrator/airrport/create and administrator/airport/update.

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



Z-test analysis:



## CONCLUSIONS

# Everything went as expected.

## BIBLIOGRAPHY

Intentionally blank.