## **DP2 – Testing Report**

**Group:** C3.027  
**Repository:** <https://github.com/DP2-C1-027/AirNav-Logistics-C3>

**Group Members:** Niza Cobo, Manuel Jesús; Gomez Claraco, Nicolas.

**Corporate Emails:** [mannizcob@alum.us.es](mailto:mannizcob@alum.us.es) , [nicgomcla@alum.us.es](mailto:nicgomcla@alum.us.es)   
**Date:** 10/16/2025

**Table of Contents**

1. Executive Summary
2. Revision Table
3. Introduction
4. Contents
5. Conclusions
6. Bibliography

## **Executive Summary**

This report provides an overview of the testing phase conducted to assess the functionalities developed by the group C3.027. The objective of the testing process was to ensure that the implemented features meet the defined quality standards in terms of functionality and performance.

Functional testing confirmed that the developed requirements generally behave as expected, fulfilling the specified criteria. Performance testing was also conducted to evaluate the system’s efficiency under different load conditions. The results indicate that the software performs within acceptable limits, with some areas identified for potential optimization.

Overall, the tests demonstrate that the implemented functionalities are stable and meet the project’s requirements, although minor improvements could further enhance performance and maintainability.

## **Revision Table**

|  |  |  |
| --- | --- | --- |
| **Revision Number** | **Date** | **Description** |
| 1.0 | 10/14/2025 | Initial version of the document |

## **Introduction**

The purpose of this document is to analyze and evaluate the operation and performance of the requirements developed by the group C3.027. Through various functional and performance tests, the aim is to ensure that the implemented functionalities meet the expected quality criteria, both in terms of correct behavior and efficiency.

First, there is a section dedicated to functional testing, which collects the results obtained after running various tests on thedeveloped functionalities. Secondly, there is a section focused on performance testing, which studies the efficiency of the software under certain conditions.

## **Content**

## **Functional Testing**

Functional testing was carried out to verify that #11 requirement behave according to the expected specifications. The tests were executed using .safe and .hack files that represent different input scenarios, including typical and boundary cases.

All test cases executed successfully, and no functional defects were detected during this phase:

11) Operations by **administrators** on **airports**:

* List the airports and show their details.

|  |  |  |  |
| --- | --- | --- | --- |
| Feature | Description | File | Result |
| List airports | - Access the list of airports with administrator account. | list.safe | High (100%) |
| - Attempting to access the list of airports with technician member account. | list.hack |

|  |  |  |  |
| --- | --- | --- | --- |
| Feature | Description | File | Result |
| Show the details of an airport | - Access airport data with administrator account. | show.safe | High (100%) |
| - Attempting to access airport data with technician member account.  - Attempting to access airport data without an id provided.  - Attempting to access airport data with an incorrect id. | show.hack |

* Create and update an airport. Both operations require confirmation.

|  |  |  |  |
| --- | --- | --- | --- |
| Feature | Description | File | Result |
| Create an airport | - Create a valid airport.  - Attempting to create an invalid airport, testing all the fields with incorrect data. | create.safe | High (100%) |
| - Attempting to access the create airport view with technician member account.  - Attempting to create an airport without and id provided.  - Attempting to create an airport with and id provided different from 0. | create.hack |
| Update an airport | - Update a valid airport.  - Attempting to update an invalid airport, testing all the fields with incorrect data. | update.safe | High (99.5%) |
| - Attempting to access the update airport view with technician member account.  - Attempting to update an airport without and id provided.  - Attempting to update an airport with an invalid id provided. | update.hack |

We will analyze the code coverage we have achieved thanks to the functional tests performed. Code coverage indicates what percentage of the source code has been executed during testing. This allows us to identify the areas that have been verified and those that have not yet been analyzed (which may contain undetected errors):

Gráfico

Descripción generada automáticamente con confianza media

We can see that classes related with requirement #11 reach close to 100%. This implies that much of the code implemented to perform these functionalities is tested in case there is an error that has not been contemplated.

To justify the remaining 0.2%, I will explain which concepts have not been analyzed or taken into account:

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

Although exhaustive tests are performed on both .safe and .hack to test the code, Eclipse has not been able to detect the different tests.

## **Performance Testing**

Here we present the results obtained after running the functional tests, developed for requirement #11 of Group C3.027. The main objective is to evaluate the impact of the tests on response times.

The results obtained by analyzing the log trace when running the launcher replay. During this process, the average response times of the feature was recorded, allowing statistics to be generated.

Tabla

El contenido generado por IA puede ser incorrecto.

Gráfico

El contenido generado por IA puede ser incorrecto.

The performance tests without indexes meet some basic response requirements, so we decided to add indexes and improve some line codes on the airport features to check if the improvement is worth.

Tabla

El contenido generado por IA puede ser incorrecto.

Gráfico

El contenido generado por IA puede ser incorrecto.

Based on the data obtained, the average response times showed an increase, going from 15.4 ms before optimization to 10.7 ms after the implementation of indexes in average.

Interfaz de usuario gráfica, Aplicación, Tabla

El contenido generado por IA puede ser incorrecto.

Tabla

El contenido generado por IA puede ser incorrecto.

The two-tailed p-value is close to 0.033, which reinforces the claim that there may be sufficient evidence to assert that response times improved, as this value is below the alpha threshold of 0.05 in device 1. However, it is still close to 0.05 alpha threshold, so we cannot garantee the improvement.

**The 95% confidence intervals overlap:**

Before the change: [10.36 ms, 20.61 ms]

After the change: [7.65 ms, 13.89 ms]

The performance of the system will now be compared on two different computers, PC1 and PC2 (group partner). Will be the actual project with indexes in both computers.

Tabla

El contenido generado por IA puede ser incorrecto.

Tabla, Excel

El contenido generado por IA puede ser incorrecto.

The two-tailed p-value is 0.998, which confirms that there is not improvement between PC1 and PC2, since this value is above the alpha threshold of 0.05. This makes sense, as both computers have similar computational characteristics.

**The 95% confidence intervals overlap:**

PC1: [7.65 ms, 13.89 ms]

PC2: [5.17 ms, 8.69 ms]

## **Conclusions**

Since we want to evaluate the impact of functional tests and system performance, multiple functional tests were carried out, satisfying 99.8% coverage for the Airport entity and their functionalities.

In performance testing, it can be stated that neither device showed statistically significant differences in response times between them. However, with regard to changes made to the code (including the addition of indexes), response times have improved substantially. Nevertheless, the two-tailed p-value is close to the 0.05 threshold, so it may be advisable to conduct further testing or obtain more data to confirm the improvement.

## 

## **Bibliography**

Intentionally blank.