Testing Report – Student #4

* **Group Number**: C1.034
* **Repository**: https://github.com/CarlosCerdaMorales/dp2-c1-034
* **Workgroup Members** **and Corporate Emails:**

Loubna Founoun Elaoud: [loufouela@alum.us.es](mailto:loufouela@alum.us.es)

Carlos Cerdá Morales: [carcermor@alum.us.es](mailto:carcermor@alum.us.es)

José Luis Moraza Vergara: [josmorver@alum.us.es](mailto:josmorver@alum.us.es)

Claudia Oviedo Govantes: [claovigov@alum.us.es](mailto:claovigov@alum.us.es)

Sergio Cantillo Blanco: [sercanbla@alum.us.es](mailto:sercanbla@alum.us.es)

* **Date**: Monday 26th, May 2025

**Table Of Contents**

[1. Executive Summary 3](#_451ewtno8qvm)

[2. Revision Table 3](#_ov3mb6vwut6t)

[3. Introduction 3](#_r2bem3ub6oc7)

[4. Contents 3](#_nh08ol2sdh4r)

[4.1 Functional Testing 3](#_h69on2nqubkm)

[4.2 Performance Testing 3](#_xn4c25wc84mn)

[5. Conclusions 3](#_2l2wgk3amr06)

[6. Bibliography 3](#_a255vkbpqjez)

# Executive Summary

This document outlines the testing process I conducted to evaluate both the functionality and performance of my student's system. The primary goal was to confirm that the application's features operate correctly and to measure how efficiently it performs under typical usage conditions.

The functional testing phase involved systematically verifying that all test cases associated with each feature outlined in my requirements were executed and produced the expected results. To carry out these tests, I utilized the testing tools and methodologies introduced during the lessons, which provided a structured approach for validating the application's functionality in a reliable and consistent manner.

The performance evaluation was carried out in two distinct setups: one where the database was left without optimization, and another where relevant indexes were introduced. Using the traces generated after doing a replay , I extracted execution time data and analyzed it through Excel.

# Revision Table

|  |  |  |
| --- | --- | --- |
| **Revision Number** | **Date** | **Description** |
| 1.0 | 25/05/2025 | Initial Version |
| 1.1 | 26/05/2025 | Final Version |

# Introduction

Throughout this report, we will walk through the step-by-step guidelines and methodologies followed to carry out both performance testing and functional testing in a structured and effective manner. The objective was to gain meaningful insights into how efficiently the system can handle requests under various conditions.

The process began with functional testing, where each test case was executed and evaluated to ensure that the system's features behaved as expected. Once functionality was validated, the focus shifted to analyzing the system’s performance. This involved assessing response times and identifying potential inefficiencies, first under a baseline configuration and later after implementing optimizations using database indexing. The comparison between pre- and post-optimization performance allowed for a clearer understanding of the impact indexing has on system efficiency.

# Contents

## 4.1 Functional testing

For the functional testing, I used the recording tool to capture execution traces corresponding to each test scenario. This included both valid cases, which were saved as .safe files, and invalid cases, saved as .hack files. By systematically running all defined test cases across the different features, I was able to validate the application's behavior in both expected and edge-case situations. As a result of this thorough approach, I achieved 100% test coverage across both services, ensuring that every functional requirement was properly verified.

Tabla

El contenido generado por IA puede ser incorrecto.

Tabla

El contenido generado por IA puede ser incorrecto.

Here are the test cases tested for each feature:

|  |  |
| --- | --- |
| ASSISTANCE AGENT CLAIM | |
| CREATE SERVICE SAFE | |
| TC-01: all values null | Create a claim with all values null, a must “not be null” message must appear for every attribute. |
| TC-02: VALID CLAIM | A claim is created with all the values in the proper range, whether they are text-type or numeric-type attributes. The associated leg is the one that does allow the select, so it will be the valid one |
| CREATE SERVICE HACK | |
| TC-03: ID FIELD IS NOT 0 | There was an attempt to alter the id field of the claim during its creation, with the intention of replacing an existing booking in the database rather than generating a new entry. This action triggered an "Access Unauthorised" response from the system |
| TC-04: INVALID LEG | When a claim is created modifying with the dev tool the leg id to a one that isn`t valid, an “Access Unauthorised” must be shown. |
| TC-05: INVALID ENUM | When a claim is created modifying with the dev tool the enum value to a one that isn`t valid, an “Access Unauthorised” must be shown. |
| DELETE SERVICE SAFE | |
| TC-06: DELETE claim | A claim is deleted properly without changing any field of the claim that is going to be deleted |
| TC-07: DELETE CLAIM ALL VALUES NULL | A claim is deleted with all fields set to null in order to observe whether the system exposes any unexpected behavior or underlying bugs during the operation. |
| DELETE SERVICE HACK | |
| TC-08: DELETE a claim that not exists | An operation was performed to delete a claim using an ID that does not exist in the database. The system appropriately handled the invalid input and responded with an "Access Unauthorised" error, indicating proper enforcement of access control and validation mechanisms. |
| TC-09: DELETE a claim that was not created by me | An operation was performed to delete a claim using an ID that does not belongs to me. The system appropriately handled the invalid input and responded with an "Access Unauthorised" error, indicating proper enforcement of access control and validation mechanisms. |
| TC-10: DELETE a published claim | An operation was performed to delete a claim using an ID of a published claim. The system appropriately handled the invalid input and responded with an "Access Unauthorised" error, indicating proper enforcement of access control and validation mechanisms. |
| LIST SERVICE SAFE | |
| TC-11: LIST MY CLAIMS | I list all of the claims associated to me. |
| LIST SERVICE HACK | |
| THERE IS NO hacking attempt applicable to this specific FEATURE. | |
| PUBLISH SERVICE SAFE | |
| TC-12: all values null | Create a claim with all values null, a must “not be null” message must appear for every attribute. |
| TC-13: VALID claim | A claim is created with all the values in the proper range, whether they are text-type or numeric-type attributes. The associated leg is the one that does allow the select, so it will be the valid one |
| PUBLISH SERVICE hack | |
| TC-14: pUBLISH a claim that not exists | An operation was performed to publish a claim using an ID that does not exist in the database. The system appropriately handled the invalid input and responded with an "Access Unauthorised" error, indicating proper enforcement of access control and validation mechanisms. |
| TC-15: PUBLISH a claim that was not created by me | An operation was performed to publish a claim using an ID that does not belongs to me. The system appropriately handled the invalid input and responded with an "Access Unauthorised" error, indicating proper enforcement of access control and validation mechanisms. |
| TC-16: publish a published claim | An operation was performed to publish a claim using an ID of a published claim. The system appropriately handled the invalid input and responded with an "Access Unauthorised" error, indicating proper enforcement of access control and validation mechanisms. |
| TC-17: INVALID LEG | When a claim is published modifying with the dev tool the leg id to a one that isn`t valid, an “Access Unauthorised” must be shown. |
| TC-18: INVALID ENUM | When a claim is published modifying with the dev tool the enum value to a one that isn`t valid, an “Access Unauthorised” must be shown. |
| SHOW SERVICE SAFE | |
| TC-19: valid show | Show a claim created by me. |
| show service hack | |
| tc-20: show non-existent CLAIM | Tried to show a claim that doesn`t exist, an “Access Unauthorised” must be shown. |
| tc-21: show A CLAIM THAT IS NOT MINE | Tried to show a claim that wasn’t created by me, an “Access Unauthorised” must be shown. |
| UPDATE SERVICE SAFE | |
| TC-22: all values null | Update a claim with all values null, a must “not be null” message must appear for every attribute. |
| TC-23: VALID claim | A claim is updated with all the values in the proper range, whether they are text-type or numeric-type attributes. The associated leg is the one that does allow the select, so it will be the valid one |
| UPDATE SERVICE hack | |
| TC-24: UPDATE a claim that not exists | An operation was performed to update a claim using an ID that does not exist in the database. The system appropriately handled the invalid input and responded with an "Access Unauthorised" error, indicating proper enforcement of access control and validation mechanisms. |
| TC-25: UPDATE a claim that was not created by me | An operation was performed to update a claim using an ID that does not belongs to me. The system appropriately handled the invalid input and responded with an "Access Unauthorised" error, indicating proper enforcement of access control and validation mechanisms. |
| TC-26: update a published claim | An operation was performed to update a claim using an ID of a published claim. The system appropriately handled the invalid input and responded with an "Access Unauthorised" error, indicating proper enforcement of access control and validation mechanisms. |
| TC-27: INVALID LEG | When a claim is updated modifying with the dev tool the leg id to a one that isn`t valid, an “Access Unauthorised” must be shown. |
| TC-28: INVALID ENUM | When a claim is updated modifying with the dev tool the enum value to a one that isn`t valid, an “Access Unauthorised” must be shown. |

|  |  |
| --- | --- |
| ASSISTANCE AGENT TRACKING LOG features | |
| CREATE SERVICE SAFE | |
| TC-29: all values null | Create a tracking log with all values null, a must “not be null” message must appear for every attribute. |
| TC-30: VALID tracking log | A tracking log is created with all the values in the proper range, whether they are text-type or numeric-type attributes. The associated leg is the one that does allow the select, so it will be the valid one |
| TC-31: INVALID STATUS | When attempting to create a tracking log with 100 of resolution percentage but a ‘PENDING’ status, or a tracking log with a resolution percentage lower than 100 and a ‘ACCEPTED’ or ‘REJECTED’ status a valid error must be shown. |
| TC-32: INVALID RESOLUTION | When attempting to create a tracking log with 100 of resolution percentage, resolution field can’t be null. |
| TC-33: all values null | Create a tracking log with a resolution percentage that not follow the “must increasing” restriction. |
| CREATE SERVICE HACK | |
| TC-34: ID FIELD IS NOT 0 | There was an attempt to alter the id field of the tracking log during its creation, with the intention of replacing an existing booking in the database rather than generating a new entry. This action triggered an "Access Unauthorised" response from the system |
| TC-35: CREATE a TRACKING LOG IN AN INVALID CLAIM | If a creation operation of a tracking log is made in a claim that isn`t published, that doesn’t exists or doesn’t belong to me, a “Access Unauthorised” response is shown. |
| TC-36: INVALID EXCEPTIONAL TRACKING LOG NUMBER | When attempting to create a more than two tracking logs with 100 of resolution percentage a “Access Unauthorised” must be shown. |
| TC-37: INVALID ENUM | When a tracking log is updated modifying with the dev tool the enum value to a one that isn`t valid, an “Access Unauthorised” must be shown. |
| DELETE SERVICE SAFE | |
| TC-38: DELETE TRACKING LOG | A tracking log is deleted properly without changing any field of the claim that is going to be deleted |
| TC-39: DELETE TRACKING LOG VALUES NULL | A tracking log is deleted with all fields set to null in order to observe whether the system exposes any unexpected behavior or underlying bugs during the operation. |
|  |  |
| DELETE SERVICE HACK | |
| TC-40: DELETE a tracking log that not exists | An operation was performed to delete a tracking log using an ID that does not exist in the database. The system appropriately handled the invalid input and responded with an "Access Unauthorised" error, indicating proper enforcement of access control and validation mechanisms. |
| TC-41: DELETE a TRACKING LOG that was not created by me | An operation was performed to delete a tracking log using an ID that does not belongs to me. The system appropriately handled the invalid input and responded with an "Access Unauthorised" error, indicating proper enforcement of access control and validation mechanisms. |
| TC-42: DELETE a published TRACKING LOG | An operation was performed to delete a claim using an ID of a published claim. The system appropriately handled the invalid input and responded with an "Access Unauthorised" error, indicating proper enforcement of access control and validation mechanisms. |
| LIST SERVICE SAFE | |
| TC-43: LIST MY TRACKING LOGS | I list all of the tracking logs associated with a claim created by me. |
| LIST SERVICE HACK | |
| TC-44: LIST TRACKING LOGS IN AN INVALID CLAIM | If a list operation of a tracking log is made in a claim that isn`t published, that doesn’t exist or doesn’t belong to me, a “Access Unauthorised” response is shown. |
| PUBLISH SERVICE SAFE | |
| TC-45: all values null | Publish a tracking log with all values null, a must “not be null” message must appear for every attribute. |
| TC-46: VALID claim | A tracking log is published with all the values in the proper range, whether they are text-type or numeric-type attributes. The associated leg is the one that does allow the select, so it will be the valid one |
| TC-47: VALID RESOLUTION PERCENTAGE | The resolution percentage must follow the rules of “must be increasing”. |
| TC-48: INVALID FINAL TRACKING LOG | When attempting to publish the final tracking log (the one with ID 100), a validation error is displayed if all previous logs have not been published. |
| TC-49: INVALID EXCEPTIONAL TRACKING LOG | When attempting to publish the exceptional tracking log (the one after the claim is completed), a validation error is displayed if all previous logs have not been published. |
| TC-50: INVALID STATUS | When attempting to publish a tracking log with 100 of resolution percentage but a ‘PENDING’ status, or a tracking log with a resolution percentage lower than 100 and a ‘ACCEPTED’ or ‘REJECTED’ status a valid error must be shown. |
| TC-51: INVALID RESOLUTION | When attempting to publish a tracking log with 100 of resolution percentage, resolution field can’t be null. |
| PUBLISH SERVICE hack | |
| TC-52: PUBLISH a tracking log that not exists | An operation was performed to publish a tracking log using an ID that does not exist in the database. The system appropriately handled the invalid input and responded with an "Access Unauthorised" error, indicating proper enforcement of access control and validation mechanisms. |
| TC-53: PUBLISH a TRACKING LOG that was not created by me | An operation was performed to publish a tracking log using an ID that does not belongs to me. The system appropriately handled the invalid input and responded with an "Access Unauthorised" error, indicating proper enforcement of access control and validation mechanisms. |
| TC-54: PUBLISH a published TRACKING LOG | An operation was performed to publish a tracking log using an ID of a tracking log claim. The system appropriately handled the invalid input and responded with an "Access Unauthorised" error, indicating proper enforcement of access control and validation mechanisms. |
| TC-55: INVALID ENUM | When a tracking log is published modifying with the dev tool the enum value to a one that isn`t valid, an “Access Unauthorised” must be shown. |
| SHOW SERVICE SAFE | |
| TC-56: show valid TRACKING LOG | Showing a tracking log that belongs to a claim created by me. |
| show service hack | |
| tc-57: show non-existent TRACKING LOG | Tried to show a tracking log that doesn`t exist, an “Access Unauthorised” must be shown. |
| tc-58: show A TRACKING LOG THAT IS NOT MINE | Tried to show a tracking that wasn’t created by me or show a tracking log that belongs to a claim note created by me, an “Access Unauthorised” must be shown. |
| UPDATE SERVICE SAFE | |
| TC-59: all values null | Update a tracking log with all values null, a must “not be null” message must appear for every attribute. |
| TC-60: VALID claim | A tracking log is updated with all the values in the proper range, whether they are text-type or numeric-type attributes. The associated leg is the one that does allow the select, so it will be the valid one |
| TC-61: VALID RESOLUTION PERCENTAGE | The resolution percentage must follow the rules of “must be increasing”. |
| TC-62: INVALID STATUS | When attempting to update a tracking log with 100 of resolution percentage but a ‘PENDING’ status, or a tracking log with a resolution percentage lower than 100 and a ‘ACCEPTED’ or ‘REJECTED’ status a valid error must be shown. |
| TC-63: INVALID RESOLUTION | When attempting to update a tracking log with 100 of resolution percentage, resolution field can’t be null. |
| UPDATE SERVICE hack | |
| TC-64: update a tracking log that not exists | An operation was performed to update a tracking log using an ID that does not exist in the database. The system appropriately handled the invalid input and responded with an "Access Unauthorised" error, indicating proper enforcement of access control and validation mechanisms. |
| TC-65: update a TRACKING LOG that was not created by me | An operation was performed to update a tracking log using an ID that does not belongs to me. The system appropriately handled the invalid input and responded with an "Access Unauthorised" error, indicating proper enforcement of access control and validation mechanisms. |
| TC-66: update a published TRACKING LOG | An operation was performed to update a tracking log using an ID of a published tracking log. The system appropriately handled the invalid input and responded with an "Access Unauthorised" error, indicating proper enforcement of access control and validation mechanisms. |
| TC-67: INVALID ENUM | When a tracking log is updated modifying with the dev tool the enum value to a one that isn`t valid, an “Access Unauthorised” must be shown. |

## 4.2 Performance testing

For the performance testing phase, I conducted a replay of all the traces generated during the earlier functional testing. This was done in two separate runs: the first using the original database configuration without any indexing, and the second after applying appropriate indexes to the entities. This approach allowed for a direct comparison of system performance before and after optimization, helping to assess the impact of indexing on response times.

**Conclusion on the Impact of Indexes on Performance**

As observed in the performance results, the introduction of indexes had a modest but measurable impact on overall system responsiveness. While the improvement was not dramatic, there was a slight enhancement in execution times, particularly noticeable in the update, publish, and create operations related to the tracking log entity—where the indexes were specifically applied. These operations benefited the most due to their frequent interaction with the indexed fields, confirming that even minimal indexing can contribute to more efficient data handling in scenarios involving frequent write or update actions.

In the following images we can see some statistical information of the previous performance test:

**Interfaz de usuario gráfica, Aplicación, Tabla

El contenido generado por IA puede ser incorrecto.**

**Tabla

El contenido generado por IA puede ser incorrecto.**

We can observe from this statistical analysis that the execution times after applying indexing are consistently lower. This indicates a measurable improvement in performance. Furthermore, the confidence intervals for both scenarios are very close to each other, which suggests a high degree of reliability in the comparison. Therefore, I consider that the requirement for both confidence levels to fall within an acceptable range has been satisfactorily met.

In the following image we can see the z-test analysis:

Tabla, Excel

El contenido generado por IA puede ser incorrecto.

Given that the p-value falls within a range greater than zero, we can reasonably proceed to compare the performance based on the average. As previously confirmed, the average time is lower immediately after the introduction of indexing. This supports the conclusion that indexing has had a positive effect on performance. The statistical significance indicated by the p-value, combined with the observed reduction in average execution time, provides further evidence of this improvement.

# Conclusions

Based on the statistical analysis conducted, it is evident that the implementation of indexing has led to a noticeable improvement in system performance. This is supported by the reduction in average execution times observed after the introduction of indexes. Furthermore, the confidence intervals before and after indexing are very close, indicating consistency and reliability in the results.

Additionally, the p-value, which lies above zero, allows for a valid comparison of performance metrics through average values. Given that the post-indexing averages are demonstrably lower, we can confidently assert that indexing has contributed to performance optimization.

Therefore, the combination of statistically significant evidence and consistent confidence intervals satisfies the necessary criteria for concluding that the performance gains observed are both real and meaningful.

# Bibliography

If there’s no relevant bibliography, write “intentionally blank”.