TESTING REPORT

Imagen en blanco y negro

Descripción generada automáticamente con confianza media

DELIVERABLE 4

DESING AND TESTING 2

2023-2024

|  |  |
| --- | --- |
| Date | Version |
| 07/07/2024 | V1.0 |

|  |  |
| --- | --- |
| Group: 21 | |
| Members | Email |
| Fernández Rodríguez, Jesús | [jesferrod1@alum.us.es](mailto:jesferrod1@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/JesusFern/Acme-SF-D04>

TABLE OF CONTENTS

[Executive Summary 3](#_Toc171247712)

[Revision Table 3](#_Toc171247713)

[Introduction 3](#_Toc171247714)

[Contents 3](#_Toc171247715)

[Functional Tests 3](#_Toc171247716)

[Banner 4](#_Toc171247717)

[Performance Testing 4](#_Toc171247718)

[Conclusion 6](#_Toc171247719)

[Bibliography 7](#_Toc171247720)

# Executive Summary

This report provides detailed information obtained through the execution of functional and performance tests for deliverable D04 of the project. In this way, we can gain a thorough understanding of the methodology to be followed for conducting these tests and the conclusions we can draw from them.

# Revision Table

|  |  |  |
| --- | --- | --- |
| Revision Number | Date | Description |
| v1r0 | 07/07/2024 | - |
|  |  |  |
|  |  |  |

# Introduction

The report consists of two main sections. The first section focuses on functional testing, verifying that the system's functionalities meet the specified requirements. The second section concentrates on performance testing, ensuring that the system operates within the established performance parameters.

# Contents

## Functional Tests

The development of these tests was carried out following the methodology proposed in the course slides. The highest possible coverage was achieved by discarding cases where our natural intelligence indicated that attempting to cover certain code instructions would be pointless.

## Banner

* Show: The tests for this functionality are quite simple. The process followed involved registering as an administrator and accessing each already created banner. We achieved a coverage of 95.5%.
* Create: The test for this functionality are more complex, for each attribute we had to check for limits and all possible combinations in the creation form were covered to ensure that our validation methods were indeed correct. We achieved a coverage of 92.9%.
* Update: For this test we had to create a new object and check all the restrictions already proven on the create test, all possible combinations in the creation form were covered to ensure that our validation methods were indeed correct. We achieved a coverage of 93.6%.
* Delete: The delete test are quite simple, trying to delete a created banner. We achieved a coverage of 89,0%.

# 

## Performance Testing

These graphs show the results of profiling hardware

Gráfico, Gráfico en cascada

Descripción generada automáticamenteTabla

Descripción generada automáticamente

Tabla

Descripción generada automáticamente

Since the p-value is between 0 and alpha, the means obtained are relevant. Therefore, we can conclude that the use of indexes has worsened the performance of banner functionalities.

This is the result of profiling software.

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente

Gráfico, Gráfico de líneas

Descripción generada automáticamente

# Conclusion

In summary, we have been able to evaluate our code rigorously and understand its functioning in detail, identifying possible dead code and bugs. Additionally, thanks to the performance tests, we have been able to identify which functionalities we may need to improve if a response time requirement is presented by the client.

# Bibliography

Intentionally blank