**INFORME DE PRUEBAS**



**Grado en Ingeniería Informática – Ingeniería del Software**

**Diseño y Pruebas 2**

**Curso 2023-2024**

Lucía Campos Díez

### Índice

1. [Executive summary 4](#_Toc28482)

[2. Table Review 5](#_Toc28483)

[3. Introduction 6](#_Toc28484)

[4. Contents 7](#_Toc28485)

[5.1 Funcionalm Testing 7](#_Toc28486)

[5.2 Performance Testing 14](#_Toc28487)

[6. Conclusions 17](#_Toc28488)

[**7. Bibliography** 18](#_Toc28489)

# 

# Executive summary

Testing is a fundamental element in software development and maintenance, helping development team members verify the proper functioning of the system and detect errors.  
This document presents a report on the testing carried out by Student 5, organized into sections on functional testing and performance testing; it also includes an analysis of the test coverage.

# Review Table

|  |  |  |
| --- | --- | --- |
| **Version** | **Description** | **Date** |
| v1.0 | Initial Version | 02/07/2025 |
| v2.0 | Initial Version | 03/07/2025 |

# Introduction

In the first section, focused on functional testing, the implemented test cases will be discussed grouped by implementation, including their effectiveness in detecting errors.  
The second section addresses the system’s performance through graphs and statistical analysis of the request response times, using 95% confidence intervals before and after refactoring with corresponding metrics, along with a hypothesis test based on the analysis.

# Contents

## Funcional Testing

Below is a brief summary of the implemented test cases, with additional explanations provided for those tests that uncovered errors in the functionality implementation. Please note that in cases where a test failed due to a detected error or produced an unexpected result, the error was fixed and the test was repeated to ensure proper operation.

### InvolvedInNuevo

* **create1Nuevo.safe y create2Nuevo.safe**: ambos test fueron creados para poder completar cada una de las peticiones posibles que podrían hacer los técnicos, cometiendo así posibles errores de validación. La primera prueba que consideremos es que dejen todos los espacios en blanco, a partir de ahí vamos rellenando cada uno de los campos con valores que no son válidos para el formulario y cuyas restricciones son comunicadas con pequeños mensajes de advertencia. En este caso, no podemos dejar ninguno de los dos desplegables sin seleccionar nada y se nos avisa si la entrada que hemos metido existía previamente o no.
* **createNuevo.**hack: este test fue creado para los casos en los que creamos una relación entre un registro y una tarea de manera ilegal, es decir, los casos que se han contemplado son cambios en la id que llega en el formulario, ya sea tanto de la relación que se quiere crear como del record o de la task elegida, con valores incorrectos (letras o símbolos) o en situaciones que no se pueden dar (el record escogido esta publicado o la task escogida es de otro técnico pero no está publicada)
* **deleteNuevo.**hack: este test fue creado para los casos en los que borramos una relación entre un registro y una tarea de manera ilegal, es decir, los casos que se han contemplado son cambios en la id que llega en el formulario, ya sea tanto de la relación que se quiere crear como del record o de la task elegida, con valores incorrectos (letras o símbolos) o en situaciones que no se pueden dar (el record escogido esta publicado)
* **deleteNuevo.**safe: el test fue creado para poder completar cada una de las peticiones posibles que podrían hacer los técnicos, cometiendo así posibles errores de validación. La primera prueba que consideremos es que dejen todos los espacios en blanco, a partir de ahí vamos rellenando cada uno de los campos con valores que no son válidos para el formulario y cuyas restricciones son comunicadas con pequeños mensajes de advertencia. En este caso, no podemos dejar ninguno de los dos desplegables sin seleccionar nada.
* **final.**safe y **final2.**safe: ambos test se han hecho con el objetivo de poder cubrir el coverage de las tres entidades debido a cambios de última hora que hubo de otros compañeros en los csv.
* **showNuevo.**hack: este test se hizo contemplando la posibilidad de que alguien intente a través de la url de la barra de búsqueda intentar acceder a una relación que no le pertenece (el record no es de su propiedad).
* **updateNuevo.**hack: este test fue creado para los casos en los que actualizamos una relación entre un registro y una tarea de manera ilegal, es decir, los casos que se han contemplado son cambios en la id que llega en el formulario, ya sea tanto de la relación que se quiere crear como del record o de la task elegida, con valores incorrectos (letras o símbolos) o en situaciones que no se pueden dar (el record escogido esta publicado o no le pertenece, la task no le pertenece y no esta publicada o la relación ya ha sido previamente guardada en la base de datos)
* **updateNuevo.**safe: el test fue creado para poder completar cada una de las peticiones posibles que podrían hacer los técnicos, cometiendo así posibles errores de validación. La primera prueba que consideremos es que dejen todos los espacios en blanco, a partir de ahí vamos rellenando cada uno de los campos con valores que no son válidos para el formulario y cuyas restricciones son comunicadas con pequeños mensajes de advertencia. En este caso, no podemos dejar ninguno de los dos desplegables sin seleccionar nada.

### MaintanenceRecord

* **create1Nuevo.hack y create2Nuevo.hack:** ambos tests fueron creados para los casos en los que creamos un registro de manera ilegal, es decir, los casos que se han contemplado son cambios en la id que llega en el formulario, ya sea tanto del record como del aircraft seleccionado, con valores incorrectos (letras o símbolos). También se ha probado a cambiar los valores del enumerado y a poner valores que no deben llegar en el formulario como una fecha que debe estar en el futuro ponerla en el pasado, las notas como es opcional se puede dejar en blanco, y se comprueba que no se puede cambiar el valor de la fecha a la que se le asigna el momento de la creación del record.
* **deleteNuevo.hack:** este test fue creado para los casos en los que borremos un record de manera ilegal, es decir, los casos que se han contemplado son cambios en la id que llega en el formulario (no existe, no le pertenece o está previamente publicado).
* **deleteNuevo.safe:** el test fue creado para poder completar cada una de las peticiones posibles que podrían hacer los técnicos, cometiendo así posibles errores de validación. La primera prueba que consideremos es que dejen todos los espacios en blanco, a partir de ahí vamos rellenando cada uno de los campos con valores que no son válidos para el formulario y cuyas restricciones son comunicadas con pequeños mensajes de advertencia. En este caso, no podemos dejar ninguno de los dos desplegables sin seleccionar nada, la fecha tiene que ser posterior a la fecha actual y que el dinero tenga formato del mismo.
* **nuevoCreate.safe:** el test fue creado para poder completar cada una de las peticiones posibles que podrían hacer los técnicos, cometiendo así posibles errores de validación. La primera prueba que consideremos es que dejen todos los espacios en blanco, a partir de ahí vamos rellenando cada uno de los campos con valores que no son válidos para el formulario y cuyas restricciones son comunicadas con pequeños mensajes de advertencia. En este caso, no podemos dejar ninguno de los dos desplegables sin seleccionar nada, la fecha tiene que ser posterior a la fecha actual y que el dinero tenga formato del mismo.
* **publishNuevo.hack:** este test fue creado para los casos en los que publiquemos un record de manera ilegal, es decir, los casos que se han contemplado son cambios en la id que llega en el formulario (no existe, no le pertenece o está previamente publicado), un intento de cambio de fecha, o intentos maliciosos con las ids de los aircraft. También se contempla que no se puede publicar un record que no tenga al menos una tarea publicada o que todas las tareas que tenga lo estén.
* publishNuevo.safe: el test fue creado para poder completar cada una de las peticiones posibles que podrían hacer los técnicos, cometiendo así posibles errores de validación. La primera prueba que consideremos es que dejen todos los espacios en blanco, a partir de ahí vamos rellenando cada uno de los campos con valores que no son válidos para el formulario y cuyas restricciones son comunicadas con pequeños mensajes de advertencia. En este caso, no podemos dejar ninguno de los dos desplegables sin seleccionar nada, la fecha tiene que ser posterior a la fecha actual y que el dinero tenga formato del mismo.
* **showNuevo.hack:** este test se hizo contemplando la posibilidad de que alguien intente a través de la url de la barra de búsqueda intentar acceder a un record que no le pertenece.
* **updateNuevo.hack:** este test fue creado para los casos en los que actualicemos un record de manera ilegal, es decir, los casos que se han contemplado son cambios en la id que llega en el formulario (no existe, no le pertenece o está previamente publicado), un intento de cambio de fecha, o intentos maliciosos con las ids de los aircraft.
* **updateNuevo.safe:** el test fue creado para poder completar cada una de las peticiones posibles que podrían hacer los técnicos, cometiendo así posibles errores de validación. La primera prueba que consideremos es que dejen todos los espacios en blanco, a partir de ahí vamos rellenando cada uno de los campos con valores que no son válidos para el formulario y cuyas restricciones son comunicadas con pequeños mensajes de advertencia. En este caso, no podemos dejar ninguno de los dos desplegables sin seleccionar nada, la fecha tiene que ser posterior a la fecha actual y que el dinero tenga formato del mismo.

### Task

* **create1Nuevo.hack y create2Nuevo.hack:** ambos tests fueron creados para los casos en los que creamos una task de manera ilegal, es decir, los casos que se han contemplado son cambios en la id que llega en el formulario con valores incorrectos (letras o símbolos). También se ha probado a cambiar los valores del enumerado.
* **createNuevo.safe:** el test fue creado para poder completar cada una de las peticiones posibles que podrían hacer los técnicos, cometiendo así posibles errores de validación. La primera prueba que consideremos es que dejen todos los espacios en blanco, a partir de ahí vamos rellenando cada uno de los campos con valores que no son válidos para el formulario y cuyas restricciones son comunicadas con pequeños mensajes de advertencia.
* **deleteNuevo.hack:** este test fue creado para los casos en los que borremos una task de manera ilegal, es decir, los casos que se han contemplado son cambios en la id que llega en el formulario (no existe, no le pertenece o está previamente publicado).
* **deleteNuevo.safe:** el test fue creado para poder completar cada una de las peticiones posibles que podrían hacer los técnicos, cometiendo así posibles errores de validación. La primera prueba que consideremos es que dejen todos los espacios en blanco, a partir de ahí vamos rellenando cada uno de los campos con valores que no son válidos para el formulario y cuyas restricciones son comunicadas con pequeños mensajes de advertencia.
* **publishNuevo.hack:** este test fue creado para los casos en los que publiquemos una task de manera ilegal, es decir, los casos que se han contemplado son cambios en la id que llega en el formulario (no existe, no le pertenece o está previamente publicado).
* **publishNuevo.safe:** el test fue creado para poder completar cada una de las peticiones posibles que podrían hacer los técnicos, cometiendo así posibles errores de validación. La primera prueba que consideremos es que dejen todos los espacios en blanco, a partir de ahí vamos rellenando cada uno de los campos con valores que no son válidos para el formulario y cuyas restricciones son comunicadas con pequeños mensajes de advertencia.
* **showNuevo.hack:** este test se hizo contemplando la posibilidad de que alguien intente a través de la url de la barra de búsqueda intentar acceder a una task que no le pertenece.
* **updateNuevo.hack:** este test fue creado para los casos en los que publiquemos una task de manera ilegal, es decir, los casos que se han contemplado son cambios en la id que llega en el formulario (no existe, no le pertenece o está previamente publicado).
* **updateNuevo.safe:** el test fue creado para poder completar cada una de las peticiones posibles que podrían hacer los técnicos, cometiendo así posibles errores de validación. La primera prueba que consideremos es que dejen todos los espacios en blanco, a partir de ahí vamos rellenando cada uno de los campos con valores que no son válidos para el formulario y cuyas restricciones son comunicadas con pequeños mensajes de advertencia. En este caso, no podemos dejar el desplegable sin nada y hay que tener cuidado con el rango de cada uno de los atributos.

## Performance Testing

Pre-refactoring tests with metrics

Interfaz de usuario gráfica, Aplicación, Tabla, Excel

El contenido generado por IA puede ser incorrecto.

Post-refactoring tests with metrics

Interfaz de usuario gráfica, Aplicación, Tabla, Excel

El contenido generado por IA puede ser incorrecto.

It can be observed by comparing both graphs that the requests requiring the most time are the update of the entity maintenanteRecord.

Additionally, it is evident that, overall, before refactoring, the requests took longer to process compared to after the refactoring, where the average response time is significantly lower. However, the operations that previously took the most time now take slightly longer.

The confidence intervals with a 95% confidence level are (23, 47) before refactoring and (29, 34) after refactoring.

**Hipótesis de contraste**

Tabla

El contenido generado por IA puede ser incorrecto.

With this results we can say that the use of indixes makes the performance worse because

# Conclusions

The report includes all the tests carried out by Student 5, which have been useful for detecting some errors in the code and verifying its correct functionality, as well as for measuring performance. This has allowed us to draw conclusions about the efficiency of our system in handling requests. These were the latest results obtained in the most recent replay carried out:

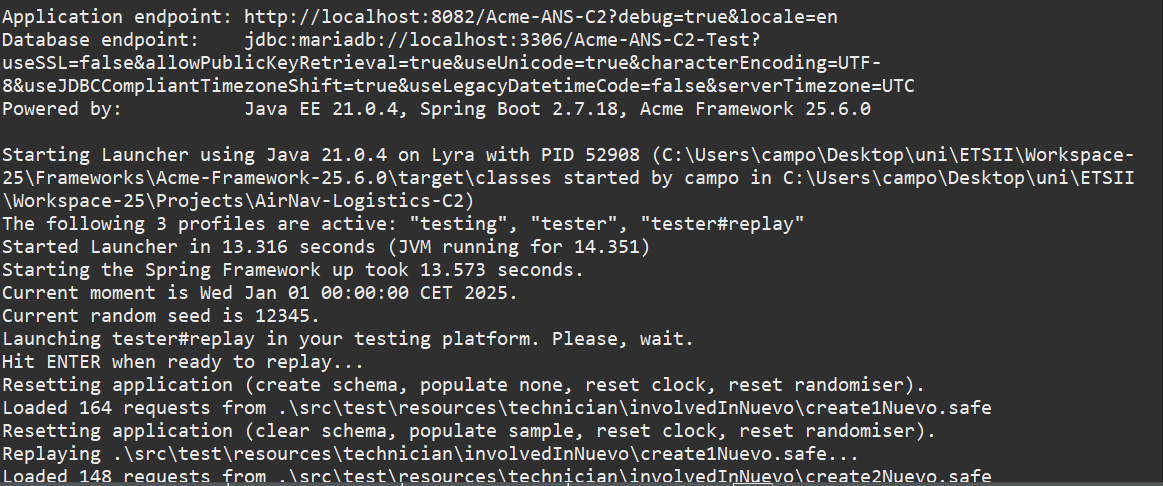


Imagen que contiene texto

El contenido generado por IA puede ser incorrecto.

Texto

El contenido generado por IA puede ser incorrecto.

Imagen que contiene texto, sostener, hombre

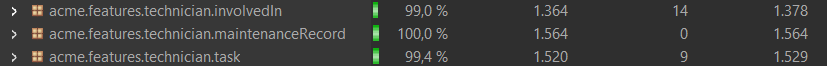
El contenido generado por IA puede ser incorrecto.

Texto

El contenido generado por IA puede ser incorrecto.

Texto

El contenido generado por IA puede ser incorrecto.

And the final coverage percentage obtained:

# **Bibliography**

Intentionally in blank.