**Testing Report**

## **Grupo C1.048**



## **Integrantes del grupo**

* Rodríguez Muñoz, Rafael: [rafrodmunn@alum.us.es](mailto:rafrodmunn@alum.us.es)

## **Repositorio**

<https://github.com/DP2-IIS-C1048/Acme-ANS-D>

26/05/2025

Índice

[Resumen ejecutivo 3](#_Toc199178759)

[Tabla de versiones 3](#_Toc199178760)

[Functional Testing 4](#_Toc199178761)

[Claims 6](#_Toc199178762)

[AssistanceAgentClaimCreateService 7](#_Toc199178763)

[AssistanceAgentClaimDeleteService 9](#_Toc199178764)

[AssistanceAgentClaimPublishService 10](#_Toc199178765)

[AssistanceAgentClaimShowService 12](#_Toc199178766)

[AssistanceAgentClaimUpdateService 13](#_Toc199178767)

[AssistanceAgentCompletedClaimListService 14](#_Toc199178768)

[AssistanceAgentPendingClaimListService 14](#_Toc199178769)

[TrackingLogs 14](#_Toc199178770)

[AssistanceAgentTrackingLogCreateService 15](#_Toc199178771)

[AssistanceAgentTrackingLogDeleteService 17](#_Toc199178772)

[AssistanceAgentTrackingLogListService 17](#_Toc199178773)

[AssistanceAgentTrackingLogPublishService 18](#_Toc199178774)

[AssistanceAgentTrackingLogShowService 20](#_Toc199178775)

[AssistanceAgentTrackingLogUpdateService 20](#_Toc199178776)

[Performance Testing 22](#_Toc199178777)

[Gráfico de peticiones y respuestas (Antes de índices) 22](#_Toc199178778)

[Comparativa Estadística descriptiva Indices 22](#_Toc199178779)

[Comparativa Estadística descriptiva Ordenadores 23](#_Toc199178780)

[Conclusión 24](#_Toc199178781)

[Bibliografía 24](#_Toc199178782)

[Firmas de los miembros 25](#_Toc199178783)

# Resumen ejecutivo

En este documento se recoge toda la información relevante de los test del student4 en el proyecto DP2-C1048, detallando todo en cuanto a la realización de las pruebas, es decir, que se ha probado, cómo y los análisis de como afectó al proyecto.

# Tabla de versiones

|  |  |  |
| --- | --- | --- |
| **Versión** | **Cambio** | **Fecha** |
| 1.0.0 | Inicio del documento | 26/05/2025 |
| 1.1.0 | Revisión de todos los apartados y preparación para la entrega. | 26/05/2025 |

Introducción

En el presente documento se describe las pruebas realizas al código del dentro el desarrollo del proyecto cuatrimestral de la asignatura DP2 para el grupo de laboratorio C1.048 los martes a las 17:40.

El documento contiene un resumen ejecutivo que proporciona una visión clara de los puntos más importantes del documento, así como la información necesaria para entenderlo. Además, el documento cuenta con una lista de versiones como se puede comprobar en la tabla de versiones.

En cuanto al contenido de este, está dividido en varios apartados que se corresponden con el Functional Testing que describe las pruebas end-to-end realizadas para mejorar el proyecto, Performance Testing para analizar como la optimización del código ha mejorado el rendimiento, por último, una conclusión con las ideas más interesantes obtenidas fruto de discutir acerca del “testing report”.

# Functional Testing

Para realizar los test siempre se ha seguido la siguiente metodología:

Archivos.safe:

* List:
  + Comprobar que todos los datos son mostrados correctamente
* Show:
  + Comprobar que todos los detalles se muestran correctamente
* Create:
  + Enviar formulario sin datos
  + Dejar todos los atributos en blanco excepto el que se esté probando; probar todas las variaciones de datos inválidas posibles y comprobar que el sistema devuelve los errores adecuados
  + Dejar todos los atributos en blanco excepto el que se esté probando; probar todas las variaciones de datos válidas posibles y comprobar que el sistema no devuelve ningún error.
  + Enviar un formulario completo con datos válidos y verificar que la operación se realiza correctamente (sin errores).
* Update:
  + Enviar formulario sin datos
  + Dejar todos los atributos en blanco excepto el que se esté probando; probar todas las variaciones de datos inválidas posibles y comprobar que el sistema devuelve los errores adecuados
  + Dejar todos los atributos en blanco excepto el que se esté probando; probar todas las variaciones de datos válidas posibles y comprobar que el sistema no devuelve ningún error.
  + Enviar un formulario completo con datos válidos y verificar que la operación se realiza correctamente (sin errores).
* Publish:
  + Enviar formulario sin datos
  + Dejar todos los atributos en blanco excepto el que se esté probando; probar todas las variaciones de datos inválidas posibles y comprobar que el sistema devuelve los errores adecuados
  + Dejar todos los atributos en blanco excepto el que se esté probando; probar todas las variaciones de datos válidas posibles y comprobar que el sistema no devuelve ningún error.
  + Enviar un formulario completo con datos válidos y verificar que la operación se realiza correctamente (sin errores).
* Delete:
  + Comprobar que el objeto se borra al darle al botón

Archivos.hack:

* List:
  + Realizar una petición a las funcionalidades usando un rol equivocado y comprobar que se deniega el acceso.
* Show:
  + Realizar una petición a las funcionalidades usando un rol equivocado y comprobar que se deniega el acceso.
  + Realizar una petición usando el rol correcto pero autenticado con un usuario que no tiene permisos y verificar que se deniega el acceso.
* Create:
  + Realizar una petición a las funcionalidades usando un rol equivocado y comprobar que se deniega el acceso.
  + Realizar una petición usando el rol correcto pero autenticado con un usuario que no tiene permisos y verificar que se deniega el acceso.
  + Realizar una petición con rol y usuario correctos, pero intentando una acción ilegal (p.ej., actualizar un recurso ya publicado) y comprobar la denegación.
  + Con el inspector de elementos (DevTools) modificar valores de campos marcados como solo lectura e intentar enviar el formulario; verificar que dichos valores no se modifican.
* Update:
  + Realizar una petición a las funcionalidades usando un rol equivocado y comprobar que se deniega el acceso.
  + Realizar una petición usando el rol correcto pero autenticado con un usuario que no tiene permisos y verificar que se deniega el acceso.
  + Realizar una petición con rol y usuario correctos, pero intentando una acción ilegal (p.ej., actualizar un recurso ya publicado) y comprobar la denegación.
  + Con el inspector de elementos (DevTools) modificar valores de campos marcados como solo lectura e intentar enviar el formulario; verificar que dichos valores no se modifican.
* Publish:
  + Realizar una petición a las funcionalidades usando un rol equivocado y comprobar que se deniega el acceso.
  + Realizar una petición usando el rol correcto pero autenticado con un usuario que no tiene permisos y verificar que se deniega el acceso.
  + Realizar una petición con rol y usuario correctos, pero intentando una acción ilegal (p.ej., actualizar un recurso ya publicado) y comprobar la denegación.
  + Con el inspector de elementos (DevTools) modificar valores de campos marcados como solo lectura e intentar enviar el formulario; verificar que dichos valores no se modifican.
* Delete:
  + Realizar una petición a las funcionalidades usando un rol equivocado y comprobar que se deniega el acceso.
  + Realizar una petición usando el rol correcto pero autenticado con un usuario que no tiene permisos y verificar que se deniega el acceso.
  + Realizar una petición con rol y usuario correctos, pero intentando una acción ilegal (p.ej., actualizar un recurso ya publicado) y comprobar la denegación.
  + Con el inspector de elementos (DevTools) modificar valores de campos marcados como solo lectura e intentar enviar el formulario; verificar que dichos valores no se modifican.

#### Conclusiones:

Todas las pruebas han sido de gran utilidad para poder verificar que todo funcionaba correctamente pero a mi parecer las que mayor me han aportado han sido todas las pruebas de hacking que me han permitido mejorar mi control del authorise y las pruebas de validaciones de datos erróneas ya que me han permitido mejorar el validate.

## Claims

Tras la ejecución de los test vemos que se llega casi al 100%. No se ha llegado al 100% total ya que en algunos servicios existen ramas inalcanzables que iremos comentando.

Texto

El contenido generado por IA puede ser incorrecto.

### AssistanceAgentClaimCreateService

Texto

El contenido generado por IA puede ser incorrecto.

Texto

El contenido generado por IA puede ser incorrecto.

Como podemos observar en este servicio todo esta en verde menos en algunos sitios debido a que algunas ramas inalcanzables son añadir una leg null o que el status sea falso.

### AssistanceAgentClaimDeleteService

Texto

El contenido generado por IA puede ser incorrecto.

Interfaz de usuario gráfica, Texto, Aplicación

El contenido generado por IA puede ser incorrecto.

Como podemos observar en este servicio todo esta en verde menos en algunos sitios debido a que algunas ramas inalcanzables son añadir una claim o una leg sean null.

### AssistanceAgentClaimPublishService

Texto

El contenido generado por IA puede ser incorrecto.

Interfaz de usuario gráfica, Texto, Aplicación

El contenido generado por IA puede ser incorrecto.

Como podemos observar en este servicio todo esta en verde menos en algunos sitios debido a que algunas ramas inalcanzables son añadir una claim o el tiempo de registro sean null.

### AssistanceAgentClaimShowService

Texto

El contenido generado por IA puede ser incorrecto.

Como podemos observar en este servicio todo esta en verde menos en algunos sitios debido a que algunas ramas inalcanzables son añadir una claim o una leg sean null.

### AssistanceAgentClaimUpdateService

Texto

El contenido generado por IA puede ser incorrecto.

Interfaz de usuario gráfica, Texto, Aplicación

El contenido generado por IA puede ser incorrecto.

Como podemos observar en este servicio todo esta en verde menos en algunos sitios debido a que algunas ramas inalcanzables son añadir una claim o una leg sean null.

### AssistanceAgentCompletedClaimListService

Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

El contenido generado por IA puede ser incorrecto.

### AssistanceAgentPendingClaimListService

Interfaz de usuario gráfica, Texto, Aplicación

El contenido generado por IA puede ser incorrecto.

## TrackingLogs

Tras la ejecución de los test vemos que se llega casi al 100%. No se ha llegado al 100% total ya que en algunos servicios existen ramas inalcanzables que iremos comentando.

Texto

El contenido generado por IA puede ser incorrecto.

### AssistanceAgentTrackingLogCreateService

Texto, Aplicación

El contenido generado por IA puede ser incorrecto.

Texto

El contenido generado por IA puede ser incorrecto.

Como podemos observar en este servicio todo esta en verde menos en algunos sitios debido a que algunas ramas inalcanzables son añadir una claim null o en draftMode.

### AssistanceAgentTrackingLogDeleteService

Interfaz de usuario gráfica, Texto, Aplicación

El contenido generado por IA puede ser incorrecto.

Como podemos observar en este servicio todo esta en verde menos en algunos sitios debido a que algunas ramas inalcanzables son añadir una claim o un trackingLog que sean null.

### AssistanceAgentTrackingLogListService

Texto

El contenido generado por IA puede ser incorrecto.

Como podemos observar en este servicio todo esta en verde menos en algunos sitios debido a que algunas ramas inalcanzables son añadir una claim null o en draftMode.

### AssistanceAgentTrackingLogPublishService

Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

El contenido generado por IA puede ser incorrecto.

Texto

El contenido generado por IA puede ser incorrecto.

Como podemos observar en este servicio todo esta en verde menos en algunos sitios debido a que algunas ramas inalcanzables son añadir una claim o un trackingLog que sean null.

### AssistanceAgentTrackingLogShowService

Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

El contenido generado por IA puede ser incorrecto.

Como podemos observar en este servicio todo esta en verde menos en algunos sitios debido a que algunas ramas inalcanzables son añadir una claim null o en draftMode.

### AssistanceAgentTrackingLogUpdateService

Interfaz de usuario gráfica, Texto, Aplicación

El contenido generado por IA puede ser incorrecto.

Texto

El contenido generado por IA puede ser incorrecto.

Como podemos observar en este servicio todo esta en verde menos en algunos sitios debido a que algunas ramas inalcanzables son añadir una claim o un trackingLog que sean null.

# Performance Testing

## Gráfico de peticiones y respuestas (Antes de índices)

Como podemos observar la mayoría de los valores no superan los 50 ms salvo dos. Podemos concluir que la mayoría de las búsquedas se realizan de manera rápida pero habría que mejorar aquellas que superen los 50 ms que deberían de ser analizadas para su optimización

## Comparativa Estadística descriptiva Indices

Interfaz de usuario gráfica, Tabla

El contenido generado por IA puede ser incorrecto.

Tabla

El contenido generado por IA puede ser incorrecto.

Tras analizar los resultados de podemos observar que Alpha es 0.05, y que el p-value es 0.00014118, por lo que podemos decir que los cambios sí dieron como resultado mejoras significativas. Por lo que concluimos que mejora el sistema con índices.

## Comparativa Estadística descriptiva Ordenadores

Interfaz de usuario gráfica, Aplicación, Tabla, Excel

El contenido generado por IA puede ser incorrecto.

Tabla

El contenido generado por IA puede ser incorrecto.

Tras analizar los resultados de podemos observar que Alpha es 0.05, y que el p-value es 0.001649841.

Por tanto existe una diferencia estadísticamente significativa entre los tiempos de respuesta en los ordenadores de David y Rafa.

Como el tiempo medio en el equipo de Rafa es mayor que en el de David, podemos afirmar con un 95 % de confianza que el rendimiento del sistema fue mejor en el ordenador de Rafa.

# Conclusión

Tras discutir los anteriores apartados concienzudamente, se llegó a varias conclusiones.

Es fundamental definir una batería de casos de prueba bien diseñados que cubran todas las rutas de ejecución posibles, especialmente en aquellas funcionalidades que involucran lógica condicional y múltiples combinaciones de valores. Esto no solo garantiza un funcionamiento correcto del sistema ante distintos escenarios, sino que permite detectar errores ocultos que podrían pasar desapercibidos.

Cabe destacar también la utilidad de los test de rendimiento, estas pruebas permiten detectar cuellos de botella, operaciones ineficientes o consultas mal optimizadas, como aquellas que no aprovechan índices, lo que puede traducirse en una experiencia de usuario pobre o en fallos en producción.

# Bibliografía

* 06 – Anexxes.docx
* Diapositivas de teoría

# Firmas de los miembros

* Rafael Rodríguez Muñoz

