

26 Mayo de 2025



TESTING REPORT STUDENT 4

REPOSITORIO: <https://github.com/AlbertoValenzuelaMunoz1/DP2-C1.002>

GRUPO: C1.002

Rafael Pineda Pérez (rafpinper@alum.us.es)

Índice

1 Resumen ejecutivo	3
2 Introducción	3
3 Pruebas realizadas	4
4 Rendimiento obtenido	6
5 Comparativa rendimiento tras creación índices	8
6 Comparativa entre distintos ordenadores	9
7 Conclusión	9
8 Bibliografía	9

1. Resumen ejecutivo

Este informe presenta los resultados de las pruebas funcionales y de rendimiento realizadas sobre el sistema desarrollado. En la primera parte, se detalla el procedimiento realizado para probar la aplicación, de forma general para todas las entidades y a continuación se han detallados los detalles concretos para probar algunas entidades. Las pruebas han permitido identificar y corregir diversos fallos críticos, especialmente en áreas como autenticación y validación de datos de entrada.

En la segunda parte, se analizó el rendimiento del sistema ejecutando las pruebas funcionales en dos equipos con diferente capacidad de procesamiento. Se recopilaron los tiempos de respuesta y se generaron intervalos de confianza del 95% para cada conjunto de datos. Asimismo, se realizó una prueba de hipótesis estadística con un 95% de confianza para realizar una comparativa entre el rendimiento obtenido antes y después de la creación de los índices.

Tabla de versiones

Versión	Fecha	Descripción
1.0.0	24/5/2025	Inicio realización de pruebas
1.1.0	25/5/2025	Fin de pruebas. Análisis de datos obtenidos y creación de índices y posterior análisis de los datos con el uso de estos y con el uso de otro PC.
1.2.0	26/5/2025	Creación del Informe definitivo

2. Introducción

Este documento recoge de manera detallada los resultados obtenidos durante el proceso de pruebas del sistema desarrollado, tanto a nivel funcional como de rendimiento. El objetivo principal de este informe es evaluar la calidad del software mediante la verificación del cumplimiento de sus funcionalidades esperadas y el análisis de su en diferentes entornos de ejecución.

Las pruebas funcionales se centraron en verificar que las características clave del sistema operan correctamente en distintos escenarios, incluyendo tanto casos de uso comunes como situaciones de error. Para ello, se diseñaron casos de prueba específicos para cada funcionalidad del sistema, los cuales permitieron detectar errores, validar la lógica de negocio y comprobar la robustez de las validaciones implementadas.

Por otro lado, las pruebas de rendimiento se llevaron a cabo para medir el tiempo de respuesta del sistema durante la ejecución de las pruebas funcionales, utilizando dos ordenadores con distintas capacidades de hardware y en el mismo ordenador antes y después de la creación de índices. Esta evaluación permitió generar intervalos de confianza del 95% y realizar un contraste de hipótesis que permitió determinar en cuál de los equipos el sistema se desempeña mejor.

Este documento se estructura en dos partes. En la primera parte, se presenta el conjunto de pruebas funcionales, organizadas por característica. En la segunda parte, se expone el análisis de las pruebas de rendimiento, incluyendo gráficos representativos, los intervalos de confianza calculados y los resultados del contraste estadístico.

3. Pruebas realizadas

3.1. Procedimiento común realizado en las pruebas

El objetivo de este apartado es detallar el procedimiento que se ha realizado de forma común en todas las entidades. Cabe destacar que además de este procedimiento, en algunas entidades más complejas sí se han llevado a cabo otras pruebas adicionales que se explicarán más adelante.

3.1.1 Procedimiento común para pruebas de listado

Para las pruebas de listado la prueba realizada consiste en acceder al listado de la entidad y comprobar que aparece la información correcta. Para el Hacking se comprobaba que usuarios no autorizados no pudiesen acceder a estas.

3.1.2 Procedimiento común para pruebas de tipo show

Para las pruebas de tipo show el procedimiento ha consistido en acceder a la pantalla de show de una entidad y comprobar que aparece toda la información de forma correcta.

Además, como prueba de hacking se ha probado a modificar la URL para colocar el id de una entidad que no pertenece al usuario y se ha verificado que se obtiene una pantalla de pánico en estos casos. También las mismas que en el listado viendo que usuarios no autorizados pudiesen acceder a estas pantallas. La realización de estas pruebas no ha detectado errores nuevos porque este procedimiento ya se había realizado al hacer el testing informal.

3.1.3 Procedimiento común para pruebas de edición/creación

Para las pruebas de creación/edición/publicado de entidades se ha llevado a cabo el procedimiento que se ha explicado en la asignatura. Dicho procedimiento consiste en enviar primero el formulario en blanco y luego ir probando variaciones de los distintos atributos dejando siempre todos en blanco menos el que se está probando.

Finalmente, una vez probado suficientes variaciones de los atributos utilizando como ayuda los valores proporcionados en la hoja de Excel Sample Data se ha enviado el formulario con valores válidos de cada uno de los atributos y se ha comprobado que se realiza correctamente el proceso sin producirse ningún error. Este procedimiento de

prueba no ha detectado demasiados errores, puesto que ya se había probado mientras se iba haciendo la propia funcionalidad

3.1.4 Procedimiento común para pruebas de tipo borrado

Para las pruebas de borrado el procedimiento realizado ha consistido en acceder a una entidad posible de borrar y eliminarla y comprobar que el proceso finaliza correctamente.

Además, como prueba de hacking se ha probado a eliminar una entidad que no pertenece al usuario o que está publicada modificando el input oculto del id mencionado anteriormente y se ha comprobado que en estos casos se ha obtenido una pantalla de pánico. Este procedimiento, ha permitido detectar este problema de hacking que quedaba en la aplicación.

3.2.1 Procedimiento común para pruebas de hacking en listado y show.

Estas se han centrado en que solo pudiesen acceder a las das diferentes claims o tracking logs solo con los que tengan el acceso autorizado y sean los asignados a sus correspondientes claims.

3.2.2 Procedimiento común para pruebas de hacking en edición/publicado/borrado

Para las pruebas de edición y publicado se ha llevado a cabo la siguiente prueba en todas las entidades. Al entrar en un formulario si se inspecciona el HTML haciendo uso de las herramientas de desarrollador del navegador se observa que aparece un input oculto que genera el framework de forma automática. Este id es el que se utiliza a la hora de editar publicar o borrar la entidad y si no se valida que se corresponda con una entidad que le corresponda al usuario y este publicada, podría hacer modificar o eliminar entidades que no le corresponde. Por tanto, la prueba realizada consiste en modificar ese id poniendo uno de una entidad que no le corresponda a ese usuario o que este publicada y verificar que se obtiene una pantalla de pánico. La realización de estas pruebas ha sido de gran utilidad porque se ha detectado casos de hacking de los que no estaba protegida la aplicación durante la realización de las mismas. También me ayudo a encontrar un bug que una vez ocurriese un error en el formulario de actualización o publicación no se podía seguir trabajando ya que se me había olvidado añadir la propiedad "draftMode" al unbind.

3.2.3 Procedimiento común para pruebas de hacking en creación

La prueba realizada en este caso es muy similar a la del apartado anterior y consiste en el mismo problema que ocurría con el input oculto con el id. Se ha observado que este input también está presente en el formulario de creación y que además se hacía bind de este atributo de forma automática. Esto habría las puertas a casos de hacking contra la aplicación ya que bastaba con editarlo y poner un id de otra entidad que ya existía y se podía editar una entidad que ya existe desde el formulario de creación, incluso aunque esta este publicada o no pertenezca al usuario. La prueba realizada consiste en editar dicho id y verificar que si era distinto de 0 (que es el valor que debe

tener el id para que al usar el método save del repositorio no se cree una entidad nueva) debía aparecer una pantalla de pánico. La realización de esta prueba ha sido de gran utilidad, ya que es un caso que no se había planteado y que habría las puertas a casos de hacking.

3.2 Pruebas realizadas entidad Claim

En esta entidad se han realizado las pruebas comentadas anteriormente. Destaca la relación con Leg ya que es un tanto compleja lo que ha hecho que se tengan que probar varios casos para los diferentes atributos de estas y comprobar que todo funcionase correctamente. Esto ha conseguido una cobertura del 99,7% que es prácticamente un 100%.

3.3 Pruebas realizadas entidad TrackingLog

En el caso de las TrackingLog se ha tenido que hacer un par de comprobaciones adicionales además de todas las mencionadas anteriormente también por su relación con las Claims.

En este caso se ha conseguido una cobertura del 97,2% muy cercano también al 100%, ya que hay supuestos que en ningún caso ocurren, pero si están contemplados por si acaso.

3.3.1 Pruebas extras realizadas

Algunas de las pruebas extras realizadas:

- Para las TrackingLogs se hicieron pruebas extras tanto normales como de Hacking ya que estas tienen diversas restricciones a tener en cuenta. Por lo que hubo que comprobar tanto mediante el formulario como mediante el Hacking que los atributos de las tracking logs como el Status o el porcentaje de resolución funcionase correctamente, probando todos los supuestos descubriendo dos errores que fueron solucionados gracias esto.
- Modificación del desplegable de legs: en el formulario de Claim aparecía un desplegable donde podíamos seleccionar la leg asociada, este debía cumplir unos requisitos. La prueba realizada consistió en modificar los valores cambiando el HTML haciendo uso de las herramientas del navegador y comprobar que se obtuviera una pantalla de pánico al introducir una leg que no cumpliese los requisitos, además saltase el aviso de NotNull en caso de introducir una leg no existente lo que también evitaría casos de Hacking. Estas pruebas encontraron un error y es que saltaba el error en legs no existentes, pero las que no cumplían si se podían asignar y crear o updatear sin problemas lo que resultaba además en otros problemas a la hora de volver a operar con esa Claim

4. Rendimiento obtenido

El objetivo de este apartado es realizar un análisis del rendimiento obtenido en las pruebas descritas anteriormente.

4.1. Gráficos rendimiento

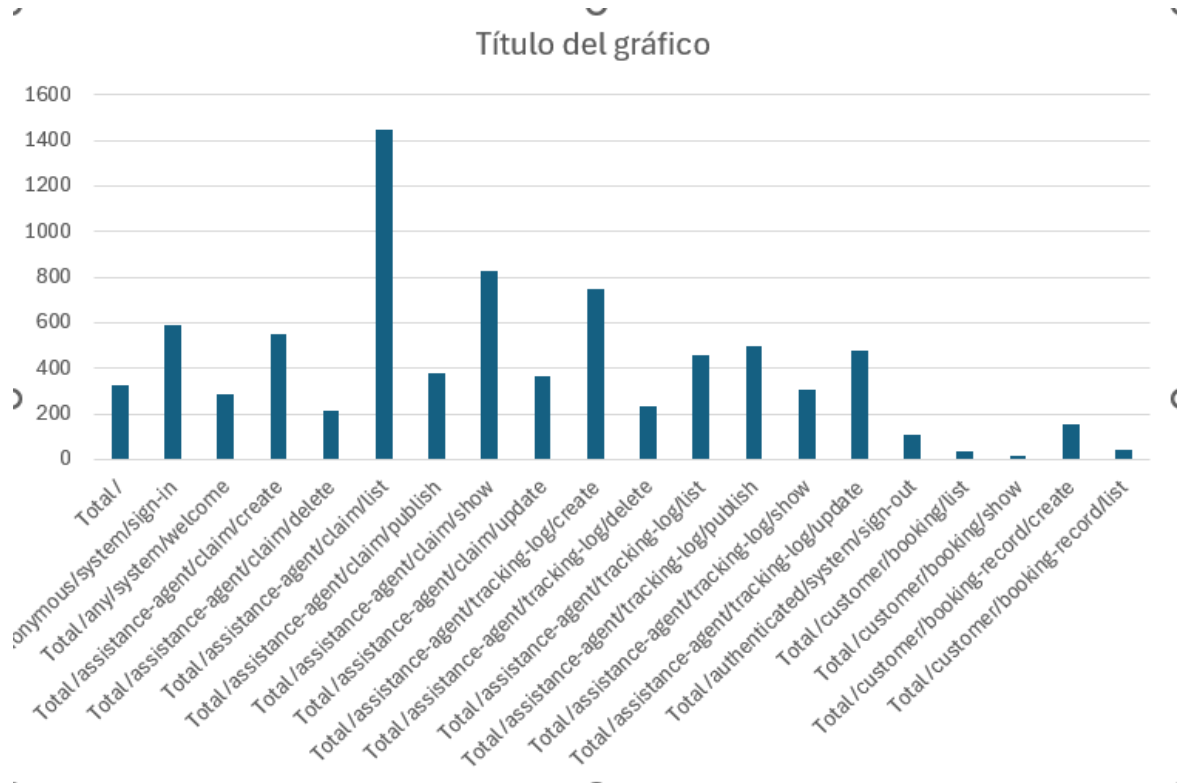


Figura 1: Promedio de tiempo de cada tipo de petición

Como se puede observar las peticiones más cargadas son las relacionadas con claim, esto se debe a que a la hora de consultar las Tracking Logs primero se tiene que “pasar” por la Claim a la que está asociada, por eso destacan tanto el list y después el show de esta. De TrackingLogs destacan update y publish ya que son los que mas llaman al repositorio para comprobar que las Legs sean correctas.

4.2. Datos estadísticos

Columna1			
Media	11,09232743		
Error típico	0,515774984		
Mediana	6,2263		
Moda	1,6517		
Desviación estándar	13,92592456		
Varianza de la muestra	193,9313747		
Curtosis	8,713236065		
Coeficiente de asimetría	2,559178559		
Rango	101,8727		
Mínimo	1,1018		
Máximo	102,9745		
Suma	8086,3067		
Cuenta	729		
Nivel de confianza(95,0%)	1,012583854		
Intervalo (m)s	10,07974358	12,1049113	

Figura 2: datos estadísticos obtenidos

Estos son los resultados estadísticos obtenidos en el análisis. El intervalo de confianza se sitúa entre 10,07 y 12,1 ms, que se puede considerar como válido, ya que en el proyecto no se impone ningún requisito relacionado con el rendimiento.

5. Comparativa rendimiento tras creación índices

En este apartado se realizará una comparativa de rendimiento tras la creación de los índices para optimizar las consultas de la base de datos, utilizando la herramienta del z-analysis de Excel.

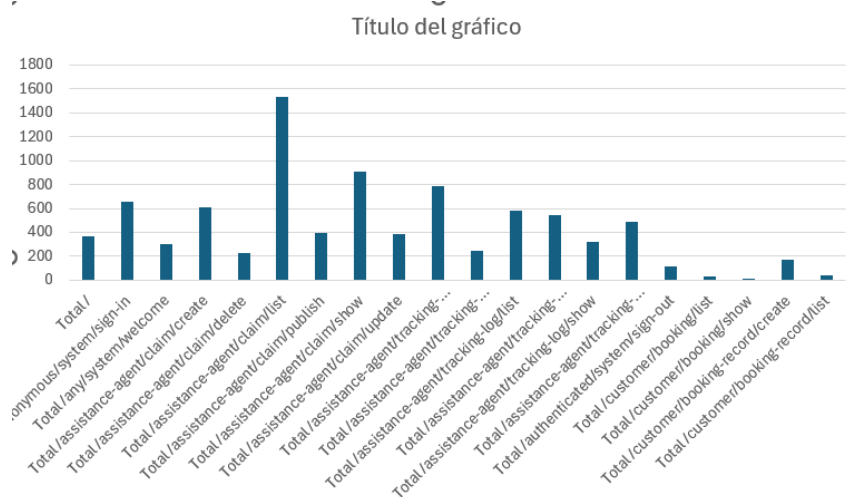


Figura 3: Promedio de tiempos obtenido tras creación de índices

Columna1				
Media	11,7788669			
Error típico	0,54885518			
Mediana	6,5346			
Moda	2,1601			
Desviación es	14,7170991			
Varianza de la	216,593005			
Curtosis	11,1443417			
Coefficiente de	2,74900411			
Rango	126,7952			
Mínimo	1,2416			
Máximo	128,0368			
Suma	8469,0053			
Cuenta	719			
Nivel de confia	1,07755281			
Intervalo(ms)	10,7013141	12,8564197		

Figura 4: Datos estadísticos obtenidos tras creación de índices

Sorprendentemente podemos observar como el rendimiento sin índices es mejor que con índices, aun así, esto también se puede deber a que en los diferentes momentos el pc estuviese más exigido o menos, debido a otras aplicaciones. Aun así y por ello la diferencia es menor a 1 ms lo que aunque es significativo aun así, no puede ser definitivo, también, por lo mencionado anteriormente.

6. Comparativa entre distintos ordenadores

A continuación, se realizará una comparativa entre el rendimiento obtenido en mi ordenador y el ordenador de otro integrante del grupo (después de la creación de índices).

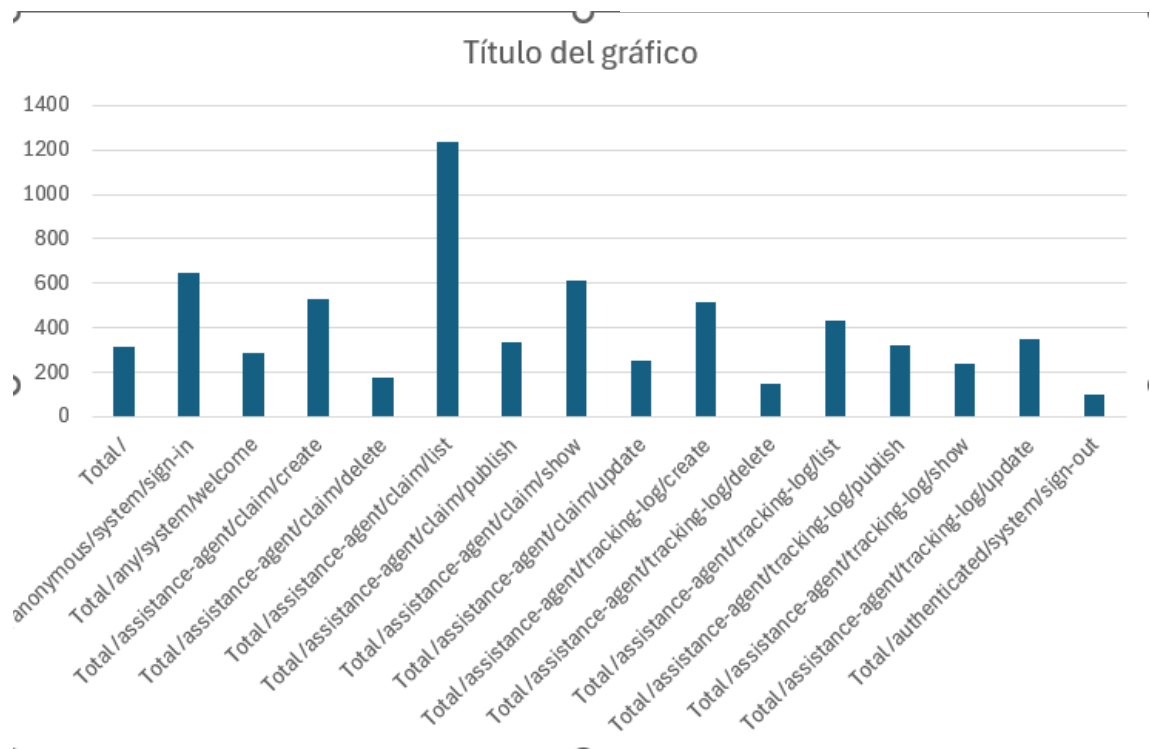


Figura 5: Promedio de tiempos obtenidos en otro ordenador

Columna1			
Media	9,12140238		
Error típico	0,42291204		
Mediana	6,3072		
Moda	1,8213		
Desviación es	11,2926226		
Varianza de la	127,523326		
Curtosis	17,0580771		
Coeficiente de	3,28006464		
Rango	104,8117		
Mínimo	0,8567		
Máximo	105,6684		
Suma	6503,5599		
Cuenta	713		
Nivel de confi	0,8303038		
Intervalo	8,29109858	9,95170619	

Figura 6: Datos estadísticos obtenidos en otro ordenador

Como en mi ordenador la media obtenida es de 11,77 ms y en el otro es de 9,21 ms, se llega a la conclusión de que el rendimiento en mi ordenador es bastante peor ya que disminuye hasta en 2.5 ms de media.

7. Conclusión

La elaboración del conjunto completo de pruebas ha resultado de gran utilidad para identificar errores residuales en la aplicación, así como para detectar posibles vectores de ataque que no se habían considerado inicialmente. Este proceso ha permitido reforzar tanto la robustez funcional como la seguridad del sistema.

En lo que respecta a la comparativa de rendimiento, las diferencias observadas tras la incorporación de índices fueron mínimas, lo cual era previsible dado el reducido volumen de datos utilizado durante las pruebas. No obstante, al comparar los resultados con los de otro integrante del grupo, se constató que el rendimiento en mi equipo fue significativamente inferior, lo que evidencia las limitaciones del hardware empleado en mi caso.

8. Bibliografía

Intencionalmente en blanco.