

Testing Report



Diseño y Pruebas II

2024/25

Group: C2.009

GitHub Organization: <https://github.com/DP2-2024-2025-C1-009>

Ricardo Carreño Mariño

riccarmar@alum.us.es

Índice

Índice	2
Tabla de versiones	2
Introducción	2
Contenido	3
Testing funcional	3
Claim	3
Tracking Log	5
Coverage	7
Testing de rendimiento	8
Antes	8
Después	10
Comparativa con el ordenador de un compañero	12
Conclusiones	14
Bibliografía	14

Tabla de versiones

Versión	Fecha	Comentarios
1.0	01/07/2025	Primera versión
1.1	03/07/2025	Completar documento

Introducción

El presente documento constituye el **informe de pruebas** del proyecto, estructurado conforme a las directrices establecidas. Su objetivo es ofrecer una visión clara, organizada y basada en datos sobre las fases de pruebas funcionales y pruebas de rendimiento.

En el **primer capítulo**, se describe en detalle la comprobación funcional del sistema. Los casos de prueba se agrupan por funcionalidad e incluyen una descripción concisa, así como una evaluación de su eficacia a la hora de detectar errores. Estas pruebas se han llevado a cabo de forma sistemática utilizando Eclipse.

En el **segundo capítulo**, se presentan los resultados de las pruebas de rendimiento. Se incluyen gráficos representativos y un análisis estadístico del tiempo de respuesta del sistema con un intervalo de confianza del 95%. También se incluye una comparativa de rendimiento entre dos ordenadores

Contenido

Testing funcional

Nota: Para los casos de prueba se ha usado la metodología y los datos recomendados por el profesorado.

Claim

assistance-agent/claim/create

Para validar esta funcionalidad, se inició enviando el formulario vacío para comprobar las restricciones básicas. Luego, se ejecutaron pruebas exhaustivas para escenarios positivos y negativos en cada uno de los campos del formulario, especialmente verificando la asociación adecuada con los Tracking Logs correspondientes. Finalmente, se procedió a crear exitosamente un claim con datos válidos y debidamente vinculado a un Tracking Log válido.

En las pruebas de seguridad, se intentó acceder al formulario con roles no autorizados y también se probaron identificadores inválidos. El sistema devolvió correctamente errores de autorización 500 (Non authorised).

No se detectaron fallos durante las pruebas realizadas en la creación y validaciones del formulario de Claim.

assistance-agent/claim/list

Esta funcionalidad fue verificada accediendo desde un usuario con rol assistance-agent, comprobando que todos los claims aparecen correctamente listados, cumpliendo así con la prueba positiva.

En términos de seguridad, se intentó acceder al listado desde un rol no autorizado, obteniendo correctamente un error 500 (Non authorised).

No se encontraron fallos durante estas pruebas.

assistance-agent/claim/delete

Se realizaron pruebas eliminando distintos tipos de claims: claims no publicados sin Tracking Logs asociados, claims no publicados con Tracking Logs asociados, y claims con Tracking Logs ya publicados. El sistema devolvió adecuadamente mensajes de validación indicando que no es posible eliminar un claim si su Tracking Log está publicado.

En la prueba de hacking, se intentó eliminar claims inexistentes, claims publicados, y claims pertenecientes a otro assistance-agent. También se probó desde un rol no autorizado. En todos los escenarios se obtuvo correctamente un error 500 (Non authorised).

No se identificaron fallos durante las pruebas de eliminación.

assistance-agent/claim/publish

Para esta funcionalidad se realizaron diversos escenarios: publicación exitosa de un claim con Tracking Log válido y publicado, intento de publicación de claims cuyos Tracking Logs no estaban aún publicados, y publicación de claims sin Tracking Logs asignados, obteniendo las advertencias esperadas.

En las pruebas de seguridad, se intentó publicar claims inexistentes, ya publicados, o pertenecientes a otros assistance-agents, obteniendo correctamente un error 500 (Non authorised). Durante estas pruebas se detectó un fallo que provocaba un error inesperado al modificar valores antes de publicar el claim. Este fallo fue corregido mediante ajustes en los métodos correspondientes del servicio ClaimPublishService.

assistance-agent/claim/show

Para validar esta funcionalidad, se accedió desde el listado a los detalles de claims publicados y no publicados, verificando que se mostraban correctamente junto a sus Tracking Logs relacionados.

Durante las pruebas de hacking, se intentó acceder a claims inexistentes, claims ajenos al assistance-agent autenticado, y desde roles no autorizados. Todas estas pruebas generaron correctamente errores 500 (Non authorised).

No se encontraron fallos en esta funcionalidad.

assistance-agent/claim/update

En la actualización de claims, se verificaron validaciones básicas enviando formularios vacíos y se realizaron pruebas exhaustivas con valores válidos e inválidos en cada campo, logrando actualizar satisfactoriamente un claim vinculado correctamente a un Tracking Log.

Las pruebas de hacking incluyeron intentos de modificar claims inexistentes, publicados, o pertenecientes a otros assistance-agents, además de intentos desde roles no autorizados. En todas estas pruebas se obtuvieron correctamente errores 500 (Non authorised).

No se detectaron fallos durante las pruebas.

Tracking Log

assistance-agent/tracking-log/create

La validación de esta funcionalidad comenzó enviando formularios vacíos, para luego ejecutar pruebas exhaustivas en todos los escenarios posibles, incluyendo solapamientos y fechas inválidas en relación con claims existentes. Finalmente, se creó exitosamente un Tracking Log válido.

En las pruebas de seguridad se verificaron los siguientes escenarios:

- Creación de Tracking Logs con claims publicados.
- Uso de identificadores incorrectos o inexistentes.
- Intentos desde roles no autorizados.
- Modificación directa desde consola de desarrolladores de campos críticos.

Todos estos escenarios devolvieron correctamente errores 500 (Non authorised). Las pruebas ayudaron a detectar errores de validación en el método authorise del servicio, especialmente en la lógica relacionada con el draftMode. Estos errores fueron corregidos satisfactoriamente.

assistance-agent/tracking-log/list

La funcionalidad fue validada accediendo desde un rol autorizado, verificando que todos los Tracking Logs se mostraban correctamente vinculados a sus respectivos claims.

En pruebas de seguridad se intentó acceder desde un rol no autorizado, obteniendo correctamente un error 500 (Non authorised).

No se hallaron errores en estas pruebas.

assistance-agent/tracking-log/delete

Se realizaron pruebas eliminando Tracking Logs no publicados asociados a claims.

En las pruebas de seguridad, se intentó eliminar Tracking Logs inexistentes, publicados, pertenecientes a otro assistance-agent y desde roles no autorizados, obteniendo en todos los casos el error 500 (Non authorised).

No se encontraron fallos durante estas pruebas.

assistance-agent/tracking-log/show

Se verificó esta funcionalidad accediendo a los detalles de Tracking Logs publicados y no publicados desde un rol autorizado, mostrando la información correctamente.

En pruebas de seguridad, se intentó acceder a Tracking Logs inexistentes, de otros assistance-agents o desde roles no autorizados, obteniendo en todos los casos el error 500 (Non authorised).

No se detectaron fallos en esta funcionalidad.

assistance-agent/tracking-log/publish

La validación incluyó formularios vacíos, pruebas exhaustivas con datos válidos e inválidos, y validaciones específicas de fechas futuras relacionadas con el reloj del sistema.

En pruebas de seguridad, se verificaron intentos de publicar Tracking Logs ya publicados, con identificadores inválidos, desde roles no autorizados y con modificaciones directas desde la consola. Todas estas situaciones devolvieron correctamente errores 500 (Non authorised).

Durante estas pruebas se detectaron y corrigieron errores en la validación del método authorise.

assistance-agent/tracking-log/update

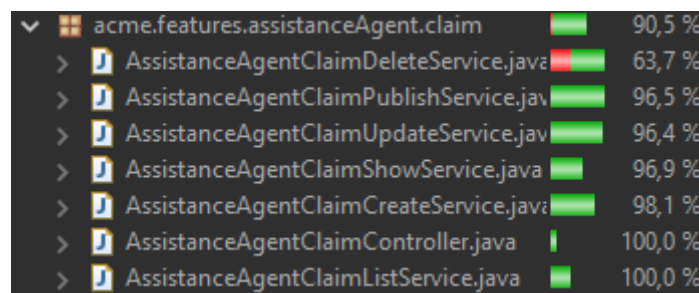
La funcionalidad fue validada con formularios vacíos y exhaustivas pruebas con valores válidos e inválidos, comprobando especialmente las restricciones relacionadas con fechas futuras.

En las pruebas de hacking se intentó actualizar Tracking Logs publicados, inexistentes, desde roles no autorizados y con modificaciones directas desde la consola. Todas estas acciones generaron correctamente errores 500 (Non authorised).

Durante estas pruebas se detectaron errores en la validación del método authorise, que fueron corregidos exitosamente.

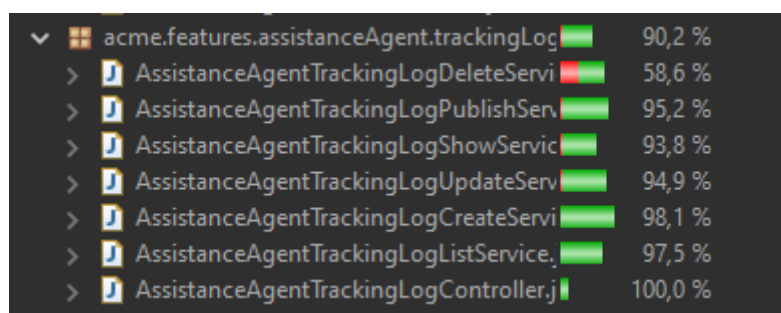
Coverage

A continuación se incluyen capturas de pantalla y explicaciones relacionadas con la cobertura obtenida al realizar las pruebas.







acme.features.assistanceAgent.claim	90,5 %
> AssistanceAgentClaimDeleteService.java	63,7 %
> AssistanceAgentClaimPublishService.java	96,5 %
> AssistanceAgentClaimUpdateService.java	96,4 %
> AssistanceAgentClaimShowService.java	96,9 %
> AssistanceAgentClaimCreateService.java	98,1 %
> AssistanceAgentClaimController.java	100,0 %
> AssistanceAgentClaimListService.java	100,0 %

He obtenido un **90,5%** en la cobertura de las pruebas relacionadas con Claim por lo que he ejecutado todas las instrucciones disponibles.



acme.features.assistanceAgent.trackingLog	90,2 %
> AssistanceAgentTrackingLogDeleteService.java	58,6 %
> AssistanceAgentTrackingLogPublishService.java	95,2 %
> AssistanceAgentTrackingLogShowService.java	93,8 %
> AssistanceAgentTrackingLogUpdateService.java	94,9 %
> AssistanceAgentTrackingLogCreateService.java	98,1 %
> AssistanceAgentTrackingLogListService.java	97,5 %
> AssistanceAgentTrackingLogController.java	100,0 %

He obtenido un 90,2 % de cobertura en las pruebas, tanto en Claim como en Tracking Log. No se alcanza un porcentaje mayor porque el método `unbiend` del `DeleteService` de ambas entidades no llega a ejecutarse durante el flujo normal, por lo que sus instrucciones no se cubren.

>	 <code>ClaimValidator.java</code>		70,1 %
>	 <code>TrackingLogValidator.java</code>		90,6 %

Aquí podemos comprobar la cobertura en las validaciones a nivel de entidad de Claimy de Tracking Log (`ClaimValidator` y `TrackingLogValidator`).

No se ha llegado al **100%** de cobertura en estos casos ya que existen validaciones que no se pueden ejecutar probando la aplicación. Estas validaciones están incluidas para comprobar los datos de prueba.

Testing de rendimiento

Se presentan los resultados de las pruebas de rendimiento. Se incluyen gráficos representativos y un análisis estadístico del tiempo de respuesta del sistema con un intervalo de confianza del **95%**

En primer lugar se ejecutaron los tests sobre el repositorio sin ningún tipo de cambio relacionado con optimizar el rendimiento. Después se corrieron las mismas pruebas pero haciendo uso de `@Index` para tratar de optimizar las consultas del repositorio lo máximo posible (según lo aprendido en el **S02 de la lección 4**)

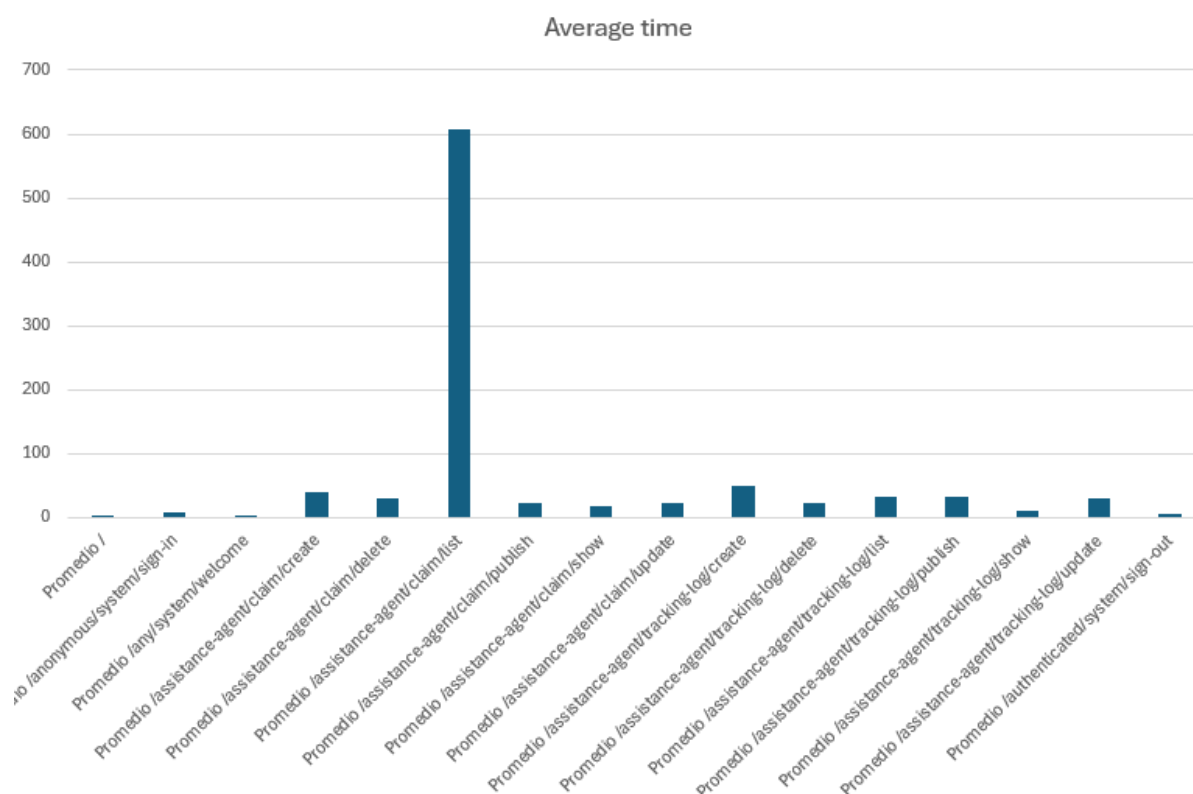
Primero analizaremos los datos de la primera ejecución y luego los compararemos con la segunda

Antes

Los resultados obtenidos se pueden visualizar en las siguientes imágenes:

Columna1	
Media	41,94517893
Error típico	4,759150316
Mediana	9,0564
Moda	1,9593
Desviación estándar	127,8786435
Varianza de la muestra	16352,94747
Curtosis	17,51569914
Coeficiente de asimetría	4,311170611
Rango	862,8439
Mínimo	1,4387
Máximo	864,2826
Suma	30284,41919
Cuenta	722
Nivel de confianza(95,0%)	9,343447889

Estos son los datos estadísticos resumidos obtenidos para la primera prueba sin realizar los índices



En esta gráfica podemos observar el tiempo medio de respuesta por tipo de petición. Podemos comprobar que el **MIR** es la creación de legs "assistance-agent/claim/list". Tras obtener estos datos consulte el servicio de creación de legs para comprobar posibles mejoras a simple vista.

Antes de hacer cambios en el código quería realizar las pruebas habiendo incluidos los índices porque podrían estar muy relacionados con las peticiones que más tardan en ejecutarse.

Promedio /	3,64709085
Promedio /anonymous/system/sign-in	7,0718349
Promedio /any/system/welcome	2,34323463
Promedio /assistance-agent/claim/create	38,6197152
Promedio /assistance-agent/claim/delete	30,845325
Promedio /assistance-agent/claim/list	607,060406
Promedio /assistance-agent/claim/publish	21,8330143
Promedio /assistance-agent/claim/show	16,6762536
Promedio /assistance-agent/claim/update	22,13433
Promedio /assistance-agent/tracking-log/create	48,895839
Promedio /assistance-agent/tracking-log/delete	22,7832
Promedio /assistance-agent/tracking-log/list	32,1239821
Promedio /assistance-agent/tracking-log/publish	32,5041696
Promedio /assistance-agent/tracking-log/show	10,2839154
Promedio /assistance-agent/tracking-log/update	29,0245333
Promedio /authenticated/system/sign-out	5,31394359

Interval (ms)	32,601731	51,2886268
Interval (s)	0,03260173	0,05128863

Después

Tras agregar los índices en las entidades de **Claim** y **Tracking Log** obtuve las siguientes estadísticas. Se muestran en comparación con las pruebas anteriores.

<i>before</i>			<i>after</i>	
Media	41,9451789		Media	41,0502197
Error típico	4,75915032		Error típico	4,21652547
Mediana	9,0564		Mediana	6,12595
Moda	1,9593		Moda	2,9292
Desviación estándar	127,878644		Desviación estándar	139,336869
Varianza de la muestra	16352,9475		Varianza de la muestra	19414,763
Curtosis	17,5156991		Curtosis	29,0596334
Coeficiente de asimetría	4,31117061		Coeficiente de asimetría	5,23731987
Rango	862,8439		Rango	1277,3919
Mínimo	1,4387		Mínimo	1,3966
Máximo	864,2826		Máximo	1278,7885
Suma	30284,4192		Suma	44826,8399
Cuenta	722		Cuenta	1092
Nivel de confianza(95,0%)	9,34344789		Nivel de confianza(95,0%)	8,27341648

Como podemos observar, ha habido un cambio significativo en los datos estadísticos proporcionados por la columna *time*.

Para comparar adecuadamente los intervalos de confianza calculados tras ejecutar las pruebas, se ha realizado la prueba **Z-Test** sobre las columnas de tiempos generados.

Los datos obtenidos son los siguientes:

Prueba z para medias de dos muestras		
	<i>before</i>	<i>after</i>
Media	34,19227888	35,55631334
Varianza (conocida)	16352,94747	19414,76303
Observaciones	1146	1717
Diferencia hipotética de las medias	0	
z	-0,269712363	
P(Z<=z) una cola	0,393690774	
Valor crítico de z (una cola)	1,644853627	
Valor crítico de z (dos colas)	0,787381548	
Valor crítico de z (dos colas)	1,959963985	

El valor que nos interesa es **$P(Z \leq z)$ two-tail**

Dicho valor nos indica que nos encontramos en el intervalo $(0.00 - \alpha]$, donde, **$\alpha = 1 -$**
Intervalo de Confianza $= 1 - 0.95 = \mathbf{0.05}$

Al ser $z = 0,787381548$ tenemos que: $z > \alpha$. El *p-value* está fuera del intervalo $[0.00, \alpha)$

El valor de **z** obtenido durante la ejecución de las pruebas ha resultado ser relativamente alto. Esto se debe a las limitaciones del equipo utilizado para el desarrollo y la ejecución del proyecto. En concreto, el ordenador cuenta con un procesador **Intel i7 de cuarta generación**, lo que implica un rendimiento inferior en comparación con dispositivos más actuales y recomendados para esta asignatura. Debido a la antigüedad del hardware, los tiempos de ejecución se ven afectados negativamente, repercutiendo directamente en el cálculo de zeta. Lamentablemente, **no dispongo de acceso a un equipo más potente**, por lo que esta ha sido la única opción disponible para la realización del trabajo.

Comparativa con el ordenador de un compañero

Este apartado consiste en comparar y analizar las diferencias entre el rendimiento obtenido desde dos máquinas distintas. En esta ocasión haré la comparativa con el ordenador de mi compañero de grupo cargalrod2.

Para la comparativa he usado la versión con índices.

Ordenador personal: Intel i7-4710, 16GB RAM, 512 SSD

Ordenador de Carlos: Ryzen 7 4700U, 16GB RAM, 512 SSD

Ricardo		Carlos	
Media	35,55631334	Mean	11,19811227
Error típico	2,801635167	Standard Error	0,702353444
Mediana	8,6851	Median	7,1111
Moda	1,8788	Mode	0,4037
Desviación estándar	116,0905123	Standard Deviation	15,53138359
Varianza de la muestra	13477,00704	Sample Variance	241,2238761
Curtosis	40,46164363	Kurtosis	13,54015326
Coefficiente de asimetría	6,006851208	Skewness	3,190203381
Rango	1277,3919	Range	121,141
Mínimo	1,3966	Minimum	0,3254
Máximo	1278,7885	Maximum	121,4664
Suma	61050,19	Sum	5475,8769
Cuenta	1717	Count	489
Nivel de confianza(95,0%)	5,49497981	Confidence Level(95,0%)	1,380010083

Prueba z para medias de dos muestras		
	time	Carlos
Media	35,55631334	11,19811227
Varianza (conocida)	13477,00704	2412238761
Observaciones	1717	489
Diferencia hipotética de las	0	
z	0,010967033	
P(Z<=z) una cola	0,495624875	
Valor crítico de z (una cola)	1,644853627	
Valor crítico de z (dos colas)	0,991249749	
Valor crítico de z (dos colas)	1,959963985	

Tras analizar los datos podemos concluir que el ordenador de mi compañero Carlos tiene mucho mejor rendimiento para este tipo de trabajos.

El *p-value* obtenido es 0,991249749, valor muy aproximado a 1 y no perteneciente al rango [0.00, alpha), siendo alpha 0.05

Esto nos resalta el cambio tan grande de rendimiento entre una máquina y otra.

Conclusiones

Tras realizar un exhaustivo conjunto de pruebas funcionales, de validación de campos y de hacking sobre todas las funcionalidades relacionadas con la entidad **Claim** y **Tracking Log**, se ha verificado que el sistema se comporta conforme a los requisitos establecidos, tanto en escenarios válidos como ante intentos de acceso o manipulación no autorizada. Las pruebas han permitido identificar y corregir algunos errores y bugs importantes que podrían haber causado errores de funcionalidad y seguridad importantes

En cuanto al apartado de rendimiento, se ha conseguido aumentar el rendimiento de las consultas haciendo uso de índices y analizando los resultados obtenidos. También hemos podido comprobar que el rendimiento depende mucho del ordenador que se utilice.

Bibliografía

- S02 - Performance testing (L04)
- Documento de anexos de la asignatura, "06- Anexes"