Universidad de Sevilla  
Escuela Técnica Superior de Ingeniería Informática

**Testing Report**



Grado en Ingeniería Informática – Ingeniería del Software  
Diseño y Pruebas 2.

Curso 2024 – 2025

|  |  |
| --- | --- |
| **Fecha** | **Versión** |
| 02/07/2025 | V2.0.0 |

|  |  |  |
| --- | --- | --- |
| **Grupo de prácticas: C2.005** | | |
| **Autores por orden alfabético** | **Rol** | **Descripción del rol** |
| González Benito, Claudio – clagonben@alum.us.es | Project Manager  Developer  Tester | Persona encargada de tomar decisiones de diseño y vigilar el correcto desarrollo  Persona encargada de desarrollar el código.  Persona encargada de realizar pruebas sobre el código. |
| Ramos Vargas, Alba – albramvar1@alum.us.es | Operador Developer  Tester | Encargado de las tareas de campo, de las instalaciones y del mantenimiento de los sistemas de la empresa.  Persona encargada de desarrollar el código.  Persona encargada de realizar pruebas sobre el código. |

**Control de Versiones**

|  |  |  |
| --- | --- | --- |
| **Fecha** | **Versión** | **Descripción** |
| 22/05/2025 | v1.0.0 | Desarrollo de la primera versión. |
| 02/07/2025 | V2.0.0 | Versión de la convocatoria 2 |
|  |  |  |
|  |  |  |

Repositorio: <https://github.com/Manuelgithuv/Acme-ANS-C2>

Índice de contenido

[**1.** **Resumen ejecutivo** 2](#_Toc198917376)

[**2.** **Introducción** 3](#_Toc198917377)

[**3.** **Testeo funcional** 4](#_Toc198917378)

[**4.** **Reporte sobre las mutaciones** 13](#_Toc198917379)

[**5.** **Cobertura de las operaciones** 14](#_Toc198917380)

[**6.** **Testeo de rendimiento** 16](#_Toc198917381)

[**7.** **Conclusiones** 19](#_Toc198917382)

[**8.** **Bibliografía** 20](#_Toc198917383)

# **Resumen ejecutivo**

Este documento corresponde al informe individual del Estudiante 4, requerido como parte del reporte de pruebas funcionales y de rendimiento relacionadas con las entidades del Assistance Agent desarrolladas por dicho estudiante. En él se detallan los errores detectados durante las pruebas, los tipos de pruebas realizadas sobre las distintas operaciones, así como las variaciones de rendimiento observadas al ejecutar la aplicación en diferentes equipos.  
El propósito principal de este documento es validar formalmente el correcto funcionamiento de la aplicación, mediante la grabación de pruebas que contemplen tanto operaciones válidas como inválidas, aplicadas a las funcionalidades implementadas.

# **Introducción**

Este documento se compone de dos capítulos fundamentales: el primero enfocado en pruebas funcionales, donde se presenta un conjunto de casos de prueba clasificados según la funcionalidad evaluada y su eficacia para identificar fallos; y el segundo dedicado a pruebas de rendimiento, el cual incluye gráficos y análisis estadísticos con intervalos de confianza del 95% sobre los tiempos de respuesta obtenidos en dos equipos diferentes, junto con una comparación que permite determinar cuál de ellos ofrece un mejor rendimiento. Ambos capítulos resultan esenciales para valorar la calidad y el desempeño del software desarrollado.

# **Testeo funcional**

En esta sección se describen detalladamente todas las pruebas realizadas para verificar el correcto funcionamiento del software. Para cada prueba se especificará: una descripción clara de lo que se ha llevado a cabo, el resultado esperado, el resultado real (entendido como el resultado obtenido tras corregir posibles errores en el código), y un apartado dedicado a los bugs encontrados.

Cabe destacar que, en los casos donde se detectaron fallos, estos fueron corregidos, por lo que el resultado real refleja el comportamiento del sistema una vez solucionados dichos errores. Cada tabla refleja un tipo de operación en nuestro sistema

* Entidad: refleja la entidad sobre la que se probo y el tipo de pruebas que se hizo, si safe o de hack
* Descripción: Se documentan los pasos concretos seguidos en la ejecución de cada prueba.
* Resultado esperado: Se indica el comportamiento previsto del sistema ante las condiciones establecidas en la prueba.
* Resultado real: Se expone el resultado obtenido tras ejecutar la prueba y tras correcciones aplicadas de los errores encontrados.
* Bugs encontrados: Se enumeran los errores identificados durante la prueba, junto con una breve explicación de las correcciones implementadas.

**List**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Entidad** | **Descripción** | **Resultado Esperado** | **Resultado Real** | **Bugs detectados** |
| **Claim safe** | Verificar que un assistance agent puede listar sus reclamaciones, tanto las que están en estado pending como las que no. | El sistema deberá mostrar las reclamaciones asociadas al assistance agent aplicando el filtro correspondiente marcado por interfaz. | El sistema ha mostrado correctamente las reclamaciones del assistance agent en ambos casos. | No se han detectado errores. |
| **Claim hack** | Verificar que un rol distinto al de assistance agent no puede listar las reclamaciones, y que el valor de url ongoing no acepte parámetros no validos. | El sistema no debe permitir listar las reclamaciones usando un rol no autorizado, debiendo generar un error de acceso no autorizado. El sistema solo debe aceptar o la url sin el parámetro ongoing o con ongoing=true | El sistema devolvía un 500 cuando no se accedía con el rol adecuado, y solo ecepta como valor valido ongoing=true o la ausencia del mismo | Se encontró un bug donde ongoing=false se consideraba valido a pesar de no ser accesible por interfaz, se arreglo en el validador |
| **Claim Tracking Log safe** | Comprobar que un assistance agent puede ver los logs de reclamaciones asociadas a sus propias reclamaciones, además de los logs asociados a una claim concreta | El sistema deberá mostrar los logs de reclamaciones correspondientes a las reclamaciones del assistance agent, además de los logs de una claim concreta | El sistema ha mostrado correctamente todos los logs de reclamaciones asociados a las reclamaciones del assistance agent, tanto de forma general como concreta | No se han detectado errores. |
| **Claim Tracking Log hack** | Comprobar que un rol no autorizado no pueda acceder a los logs de reclamaciones, y que un assistance agent no pueda acceder a logs de una reclamación ajena o inexistente. | El sistema no debe permitir el acceso a los logs si el usuario no tiene rol de assistance agent, si intenta acceder a reclamaciones de otro agente o si la reclamación no existe, debiendo devolver un error de autorización. | El sistema no permite el acceso | No se han detectado errores. |

**Show**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Entidad** | **Descripción** | **Resultado Esperado** | **Resultado Real** | **Bugs detectados** |
| **Claim safe** | Verificar que un assistance agent puede ver toda la información de su reclamacion. | El sistema deberá mostrar la información de las reclamacion asociada al assistance agent. | El sistema ha mostrado correctamente la información de las reclamacion del assistance agent. | No se han detectado errores. |
| **Claim hack** | Verificar que un rol no assistance agent no puede ver la información de una reclamacion, que un assistance agent no puede acceder a la información de una reclamación de otro agent, ni visualizar una reclamación inexistente. | El sistema no debe mostrar la información de una reclamación cuando se accede con un rol no autorizado, a reclamaciones de otro assistance agent o a reclamaciones que no existen, debiendo devolver un error de acceso no autorizado. | El sistema ha devuelto correctamente un error 500 de acceso no autorizado en los tres casos: acceso con un rol no assistance agent, intento de visualizar reclamaciones de otro agent, y consultas a reclamaciones inexistentes. | No se han detectado errores. |
| **Claim Tracking Log safe** | Verificar que un assistance agent puede ver toda la información de su log de reclamacion. | El sistema deberá mostrar la información del log de reclamacion asociado al assistance agent. | El sistema ha mostrado correctamente la información del log de reclamacion del assistance agent. | No se han detectado errores. |
| **Claim Tracking Log hack** | Verificar que un rol no assistance agent no puede acceder a la información de los logs de reclamaciones, que un assistance agent no puede visualizar logs de una reclamación de otro agent, ni acceder a los logs de una reclamación inexistente. | El sistema no debe permitir el acceso a los logs de reclamaciones si se usa un rol no autorizado, si se intenta acceder a logs de otra reclamación ajena o si la reclamación no existe, generando en todos los casos un error de acceso no autorizado. | El sistema ha respondido con un error 500 de acceso no autorizado en todos los casos: acceso desde un rol no assistance agent, reclamaciones ajenas y reclamaciones inexistentes. | No se han detectado errores. |

**Create**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Entidad** | **Descripción** | **Resultado Esperado** | **Resultado Real** | **Bugs detectados** |
| **Claim safe** | Verificar que un assistance agent puede crear reclamaciones de forma correcta, y probar que todas las validaciones de cumplen. | El sistema deberá permitir la creación de reclamaciones válidas y rechazar aquellas que no cumplan con las restricciones establecidas, devolviendo los mensajes de error esperados | El sistema ha permitido crear reclamaciones con datos válidos y ha rechazado aquellas con campos nulos o con tipos de datos incorrectos y devuelto los mensajes de error correctos | No se han detectado errores. |
| **Claim hack** | Verificar que un rol no assistance agent no puede acceder a la creación de reclamaciones, que no se puedan forzar campos de solo lectura mediante herramientas de inspección, y que el sistema no de por valido cambiar el valor de las choices de legs a un valor de id invalido | El sistema deberá devolver un error 500 al intentar acceder a la creación desde un rol no autorizado. Solo debe permitir la creación si se trata del assistance agent autenticado e ignorar los intentos de modificar campos de solo lectura mediante herramientas del navegador, y devolver error si se hackea un legId a un valor invalido que no este en el choices | El sistema ha generado correctamente un error 500 al acceder con un rol no assistance agent, ha ignorado los campos de solo lectura forzados desde herramientas de inspección y cuando se ha modificado el campo leg id a uno invalido también a devuelnto el error 500 esperado | No se han detectado errores. |
| **Claim Tracking Log safe** | El sistema deberá permitir la creación de logs válidos y rechazar aquellos que incumplan las reglas de negocio, superposición de logs. | El sistema deberá permitir crear logs validos y rechazar aquellos datos que no cumplan con las restricciones de negocio. | El sistema ha permitido crear logs de reclamaciones con datos válidos y ha rechazado aquellos con campos nulos obligatorios o con tipos de datos incorrectos y devuelto los mensajes de error correctos | Se ha detectado un mensaje de restricción incorrecto en atributo “resolutionPercentage” que decía que tenia ser un valor entre 1 y 0, se ha implementado un mensaje custom que devuelve que debe ser un valor entre 0.00 y 100.00 |
| **Claim Tracking Log hack** | Verificar que un rol no assistance agent no puede acceder a la creación de logs de reclamaciones, que un assistance agent no pueda crear logs para reclamaciones de otros usuarios hackeando el choices a otro id, y que los valores de solo lectura se ignoran | El sistema debe devolver un error 500 al acceder con un rol no autorizado, al intentar crear logs sobre reclamaciones ajenas o no existente hackeando el valor del choices y que ignore los atributos readonly modificados | El sistema ha generado un error 500 en todos los intentos de acceso indebido con roles incorrectos. También devolvió un error 500 cuando se forzaron valores no válidos en los atributos de el claimId en choices, y se han ignorado los atributos read-only. | No se han detectado errores. |

**Update**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Entidad** | **Descripción** | **Resultado Esperado** | **Resultado Real** | **Bugs detectados** |
| **Claim safe** | Verificar que un assistance agent puede editar sus reclamaciones de forma correcta. | El sistema deberá permitir actualizar reclamaciones con datos válidos y rechazar aquellas que incumplan las reglas de negocio. | El sistema ha permitido la edición de reclamaciones con datos válidos y ha rechazado correctamente aquellas con datos inválidos. | Se detectaron algunos datos de prueba en el sample que no cumplían las restricciones temporales y no dejaba actualizarlos. Se ha modificado esos valores para que tengan los vaolres correctos. |
| **Claim hack** | Verificar que un rol no assistance agent no pueda acceder a la edición de reclamaciones, que un assistance agent no pueda editar reclamaciones que estén publicadas, que los valores read only son ignorados, y que los vaolores inválidos hackeando el legId o poniendo en la url un claimId no existente no sean permitidos | El sistema deberá devolver un error 500 al intentar editar desde un rol no autorizado, o cuando se modifica legId y claimId a valores inválidos o de claims ya publicadas o de otros usuarios. Además, deberá ignorar cualquier intento de modificar campos de solo lectura desde el navegador. | El sistema ha generado correctamente un error 500 al intentar editar desde un rol no assistance agent y al cambiar los valores de las ids a valores no validos o de otros agentes o de claims ya publicadas, y ha ignorado los campos de solo lectura manipulados desde herramientas de inspección. | No se han detectado errores. |
| **Claim Tracking Log safe** | Verificar que un assistance agent puede editar logs de reclamaciones de forma correcta. | El sistema deberá permitir la edición de logs con datos válidos y rechazar aquellos que incumplan las restricciones definidas por las reglas de negocio. | El sistema ha permitido actualizar logs con datos válidos y ha rechazado correctamente los que contenían datos inválidos. | Se ha detectado errores he incongruencias en las validaciones de los datos que son correctas en el créate pero incorrectas en el update, por ende se han modificado para el update para que sean distintas y cumplan con la lógica de negocio  Se ha encontrado un bug en el validador a nivel de entidad que devolvía un error de validación en un caso en el no debía, y se ha corregido. |
| **Claim Tracking Log hack** | Verificar que un rol no assistance agent no puede acceder a la edición de logs de reclamaciones, que un assistance agent no pueda editar logs que estén publicados, que no se puedan manipular atributos navegables con IDs no válidos o de otros usuarios, y que los atributos readonly sean ignorados | El sistema deberá devolver un error 500 cuando el rol no sea valido, el id de navegación sea de un log publicado o que no pertenece al usuario o de un log invalido, y que al forzar valores readonly estos se ignoran. | El sistema ha dado un 500 donde se esperba y ignorado los campos readonly modificados | No se han detectado errores. |

**Publish**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Entidad** | **Descripción** | **Resultado Esperado** | **Resultado Real** | **Bugs detectados** |
| **Claim safe** | Verificar que un assistance agent puede publicar sus reclamaciones de forma correcta. | El sistema deberá permitir publicar reclamaciones válidas y deberá rechazar la publicación de aquellas que incumplan las reglas de negocio. | El sistema ha permitido la publicación de reclamaciones con datos válidos y ha rechazado las que contenían datos inválidos. | No se han detectado errores. |
| **Claim hack** | Verificar que un rol no assistance agent no pueda acceder a la publicación de reclamaciones, que un assistance agent no pueda publicar reclamaciones que ya están publicadas o de otros usuarios, y que no se puedan manipular atributos navegables con IDs no válidos y que se ignoren atributos readonly | El sistema deberá devolver un error 500 si se realiza alguna de las acciones anteriores salvo la de modificar atributos readonly | El sistema ha devuelto un error 500 al intentar replicar los casos anteriores salvo en el de modificar atributos readonly | No se han detectado errores. |
| **Claim Tracking Log safe** | Verificar que un assistance agent puede publicar sus logs de reclamaciones de forma correcta. | El sistema deberá permitir publicar los logs de reclamaciones válidos y deberá rechazar la publicación de aquellos que incumplan las reglas de negocio. | El sistema ha permitido la publicación de logs con datos válidos y ha rechazado los que contenían datos inválidos. | No se han detectado errores. |
| **Claim Tracking Log hack** | Verificar que un rol no assistance agent no pueda acceder a la publicación de logs de reclamaciones, que un assistance agent no pueda publicar logs que ya están publicados o de otros usuarios, y que no se puedan manipular atributos navegables con IDs no válidos. Debera ignorar la modificación de atriibutos readonly | El sistema deberá devolver un error 500 si se realiza alguna de las acciones anteriores salvo la de modificar atributos readonly | El sistema ha devuelto un error 500 al intentar replicar los casos anteriores salvo en el de modificar atributos readonly | No se han detectado errores. |

**Delete**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Entidad** | **Descripción** | **Resultado Esperado** | **Resultado Real** | **Bugs detectados** |
| **Claim safe** | Verificar que un assistance agent puede eliminar sus reclamaciones de forma correcta solo si esa reclamación no tiene logs asociados | El sistema deberá permitir eliminar reclamaciones válidas y deberá rechazar la eliminación de aquellas que incumplan las reglas de negocio, como aquellas que tiene logs creados | El sistema ha permitido eliminar reclamaciones con datos válidos y ha rechazado las que contenían datos inválidos. | Se ha detectado un erros donde tras hacer un delete invalido, los choices de legId que se devolvían algunos no eran ids validos y si seleccionaban de manera natural daba un error de autorización. Se ha corregido para que devuelva leg choices coherentes |
| **Claim hack** | Verificar que un rol no assistance agent no pueda acceder a la eliminación de reclamaciones, que un assistance agent no pueda eliminar reclamaciones que estén publicadas o que no sean suyas, que no se aceptan atributos de navegación o de choices inválidos | El sistema deberá dar un 500 al intentar hacer la peticiones mencionadas anteriormente | El sistema ha generado un error 500 cuando se esperaba | No se han detectado errores. |
| **Claim Tracking Log safe** | Verificar que un assistance agent puede eliminar logs de reclamaciones de forma correcta. | El sistema deberá permitir eliminar logs válidos. | El sistema ha permitido eliminar logs. | No se han detectado errores. |
| **Claim Tracking Log hack** | Verificar que un rol no assistance agent no pueda acceder a la eliminación de logs de reclamaciones, que un assistance agent no pueda eliminar logs que estan publicados o que no sean suyos, que no se aceptan atributos de navegación invalidos | El sistema deberá lanzar un error 500 si se intenta hacer alguna de las peticiones anteriores | El sistema ha generado un error 500 al intentar eliminar logs desde un rol no assistance agent y al intentar hacerlo sobre un log que estaba publicado. También ha generado un error 500 al manipular atributos navegables con IDs inexistentes o de logs que no le pertenecian. | No se han detectado errores. |

# **Reporte sobre las mutaciones**

No se han realizado mutaciones

# **Cobertura de las operaciones**

El porcentaje de cobertura obtenido con las pruebas formales supera el umbral mínimo del 90 % antes de iniciar las pruebas de rendimiento. En el delete de claimTrackingLog se alacnza solo el 88,5% porque es una clase con pocas líneas y hay una sola línea que nunca se ejecuta, que tras un análisis se ha decidido dejar. De este modo, queda documentado que todas las operaciones críticas han sido validadas y cumple el requisito de cobertura establecido.

1. Cobertura de Claim y Claim Tracking Log

Interfaz de usuario gráfica, Aplicación, Tabla

El contenido generado por IA puede ser incorrecto.

# **Testeo de rendimiento**

En este capítulo se analizará el desempeño de nuestro proyecto mediante pruebas enfocadas en el tiempo de respuesta al procesar las solicitudes durante las pruebas funcionales. Se emplearán dos archivos de testeo: uno correspondiente a un replay de las pruebas realizadas en un portátil, y otro correspondiente a un replay de pruebas ejecutadas en un equipo de escritorio.

Para llevar a cabo este análisis, se presentarán gráficos que representen visualmente los resultados obtenidos. Asimismo, se calcularán intervalos de confianza del 95% para los tiempos de respuesta registrados y se efectuará un contraste de hipótesis utilizando una prueba z para comparar dos medias, también con un nivel de confianza del 95%, con el objetivo de evaluar si existe una diferencia significativa al ejecutar el replay de las pruebas en un equipo frente al otro.

**Gráfico, Gráfico de barras

El contenido generado por IA puede ser incorrecto.**

**Gráfico, Gráfico de barras

El contenido generado por IA puede ser incorrecto.**

**Comparación de resultados:**

**Tabla

El contenido generado por IA puede ser incorrecto.**

En nuestras pruebas de rendimiento, hemos obtenido un valor crítico de z de **0** para un nivel de significancia (α) de 0.95. Este valor se utiliza para determinar si las diferencias en los tiempos de respuesta entre ambas pruebas son estadísticamente significativas.

Según nuestra metodología, si el p-valor se encuentra dentro del intervalo [0, α) esto indica que podemos comparar los valores. Al comparar las medias de tiempo de ejecución de las pruebas ejecutadas, ha habido cambios significantes de un ordenador a otro, siendo el ordenador 2 bastante mas lento que el ordenador 1, exactamente 2,5798 veces más lento que primero. Esto seguramente se deba al hardware desactualizado del segundo ordenador.

En resumen, las diferencias detectadas son bastante grandes, y un ordenador tiene mejores resultados de rendimiento que el otro

# **Conclusiones**

Durante el proceso de pruebas, quedó claro que una definición detallada de los casos de prueba para cada funcionalidad solicitada por el cliente es esencial para asegurar que los resultados obtenidos se correspondan con los esperados. Contar con un entorno de pruebas independiente resulta clave, ya que permite replicar fácilmente los fallos encontrados y agiliza su corrección cuando se identifican errores durante el testing.

Además, el estudio de los indicadores de rendimiento nos brinda la posibilidad de examinar aspectos no funcionales del sistema, identificar posibles cuellos de botella y determinar si estos problemas provienen del software o del hardware. En caso de que el origen sea el código, esta información es útil para orientar acciones que mejoren su eficiencia.

# **Bibliografía**

Intencionalmente en blanco.