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** |
| 22/05/2025 | v1.0.0 |

|  |  |  |
| --- | --- | --- |
| **Grupo de prácticas: C1.005** | | |
| **Autores por orden alfabético** | **Rol** | **Descripción del rol** |
| Artero Bellido Manuel – manartbel@alum.us.es | Developer | Persona encargada de desarrollar el código. |
| Calderón Rodríguez, Manuel María -mancalrod@alum.us.es | Operador | Encargado de las tareas de campo, de las instalaciones y del mantenimiento de los sistemas de la empresa. |
| González Benito, Claudio – clagonben@alum.us.es | Project Manager | Persona encargada de tomar decisiones de diseño y vigilar el correcto desarrollo |
| Márquez Gutiérrez, José Manuel – josmargut@alum.us.es | Tester | Persona encargada de realizar pruebas sobre el código. |
| Ramos Vargas, Alba – albramvar1@alum.us.es | Developer | Persona encargada de desarrollar el código. |

**Control de Versiones**

|  |  |  |
| --- | --- | --- |
| **Fecha** | **Versión** | **Descripción** |
| 22/05/2025 | v1.0.0 | Desarrollo de la primera versión. |
|  |  |  |
|  |  |  |
|  |  |  |

Repositorio: <https://github.com/Manuelgithuv/Acme-ANS-D04>

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 esta entrega 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 en la entrega anterior.

# **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.

* Descripción de la prueba: 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, considerando las correcciones aplicadas a los errores detectados.
* Bugs encontrados: Se enumeran los errores identificados durante la prueba, junto con una breve explicación de las correcciones implementadas. Una vez resueltos, se actualiza el resultado real para reflejar el correcto funcionamiento del software.

**List**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Entidad** | **Descripción** | **Resultado Esperado** | **Resultado Real** | **Bugs detectados** |
| **Claim safe** | Verificar que un assistance agent puede listar sus reclamaciones. | El sistema deberá mostrar las reclamaciones asociadas al assistance agent. | El sistema ha mostrado correctamente las reclamaciones del assistance agent. | Ninguno |
| **Claim hack** | Verificar que un rol distinto al de assistance agent no puede listar las reclamaciones. | El sistema no debe permitir listar las reclamaciones usando un rol no autorizado, debiendo generar un error de acceso no autorizado. | El sistema devolvía un 500 y la aplicacon crasheaba | No se verificaba el rol, y a la hora de buscar las reclamaciones encontraba null |
| **Claim Tracking Log safe** | Comprobar que un assistance agent puede ver los logs de reclamaciones asociadas a sus propias reclamaciones. | El sistema deberá mostrar los logs de reclamaciones correspondientes a las reclamaciones del assistance agent. | El sistema ha mostrado correctamente todos los logs de reclamaciones asociados a las reclamaciones del assistance agent. | 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. | Ninguno |
| **Claim hack** | Verificar que un rol no assistance agent no puede ver la información de las reclamaciones, 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 reclamaciones 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. | Ninguno |
| **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 de el log de reclamacion asociado al assistance agent. | El sistema ha mostrado correctamente la información del log de reclamacion del assistance agent. | Ninguno |
| **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. | Ninguno |

**Create**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Entidad** | **Descripción** | **Resultado Esperado** | **Resultado Real** | **Bugs detectados** |
| **Claim safe** | Verificar que un assistance agent puede crear reclamaciones de forma correcta. | El sistema deberá permitir la creación de reclamaciones válidas y rechazar aquellas que no cumplan con las restricciones establecidas. | El sistema ha permitido crear reclamaciones con datos válidos y ha rechazado aquellas con campos nulos o con tipos de datos incorrectos. | Ninguno |
| **Claim hack** | Verificar que un rol no assistance agent no puede acceder a la creación de reclamaciones, que un assistance agent no puede crear reclamaciones para otro usuario, y que no se puedan forzar campos de solo lectura mediante herramientas de inspección. | 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 (evitando ataques de tipo *Get Hacking*) e ignorar los intentos de modificar campos de solo lectura mediante herramientas del navegador. | El sistema ha generado correctamente un error 500 al acceder con un rol no assistance agent, ha permitido la creación por parte del assistance agent autenticado y ha ignorado los campos de solo lectura forzados desde herramientas de inspección. | Ninguno |
| **Claim Tracking Log safe** | El sistema deberá permitir la creación de logs válidos y rechazar aquellos que incumplan las reglas de negocio, como fechas incorrectas, superposición de logs. | El sistema deberá permitir crear logs válidos y rechazar aquellos datos que no cumplan con las restricciones de negocio. | El no ha dejado crear algunos tracking logs con datos validos | Habia restricciones en el validador que nunca se cumplían y otras que necesitaban revisar que algunos valores eran null |
| **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 o ya publicadas, y que no sea posible manipular atributos navegables con valores no válidos. | El sistema debe devolver un error 500 al acceder con un rol no autorizado, al intentar crear logs sobre reclamaciones ajenas o ya publicadas. También debe rechazar intentos de modificar atributos navegables con IDs inexistentes. | El sistema ha generado un error 500 en todos los intentos de acceso indebido: desde roles no assistance agent, sobre reclamaciones de otro agente, y sobre reclamaciones ya publicadas. También devolvió un error 500 cuando se forzaron valores no válidos en los atributos navegables, excepto en el caso de un recurso existente pero en mantenimiento, donde el sistema respondió con un error controlado. | Ninguno |

**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. | Ninguno |
| **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 | El sistema deberá devolver un error 500 al intentar editar desde un rol no autorizado 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 ha ignorado los campos de solo lectura manipulados desde herramientas de inspección. | Ninguno |
| **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. | Ninguno |
| **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, y que no se puedan manipular atributos navegables con IDs no válidos. | El sistema deberá devolver un error 500 para cualquiera de las peticiones anteriores | El sistema ha dado un 500 donde se esperba salvo cuando se ha editado el campo claimId con el navegador que ha dejado actualizarlo al claim de otro asistente. | La validaciones de propiedad de la claim (que si se hacían correctamente en el create) no se hacían correctamente ni en el update ni en el publish |

**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. | Ninguno |
| **Claim hack** | Verificar que un assistance agent puede publicar logs de reclamaciones de forma correcta. | El sistema deberá permitir la publicación de logs válidos y rechazar aquellos que incumplan las reglas de negocio. Ademas, solo publicara un log si su reclamación esta publicada | El sistema ha permitido publicar logs con datos válidos y ha rechazado correctamente los que contenían datos inválidos. | No se han detectado errores. |
| **Claim Tracking Log safe** | Verificar que un mánager puede publicar tramos de vuelos de forma correcta. | El sistema deberá publicar tramos válidos. Y rechazar la publicación de tramos que incumplan las reglas de negocio. | El sistema ha dejado publicar tramos con datos válidos. Y ha rechazado los de los datos inválidos. | Ninguno |
| **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, y que no se puedan manipular atributos navegables con IDs no válidos. | El sistema deberá devolver un error 500 si se realiza alguna de las acciones anteriores | El sistema ha devuelto un error 500 al intentar replicar los casos anteriores | Ninguno |

**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 es reclamación no tiene logs creados | 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. | Ninguno |
| **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 | El sistema deberá dar un 500 al intentar hacer la peticiones mencionadas anteriormente | El sistema ha generado un error 500 al intentar acceder a la eliminación desde un rol no assistance agent. | Ninguno |
| **Claim Tracking Log safe** | Verificar que un assistance agent puede eliminar logs de reclamaciones de forma correcta. | El sistema deberá permitir eliminar logs válidos y rechazar la eliminación de aquellos que incumplan las reglas de negocio. | El sistema ha permitido eliminar logs con datos válidos y ha rechazado los que contenían datos inválidos. | Ninguno |
| **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 no estén publicados, y que no pueda eliminar logs manipulando atributos navegables con IDs no válidos. | 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. | Ninguno |

# **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. 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

Tabla

El contenido generado por IA puede ser incorrecto.

# **Testeo de rendimiento**

En este capítulo se analizará el desempeño de nuestro proyecto mediante pruebas exhaustivas 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 con especificaciones algo superiores al primero.

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, Excel

El contenido generado por IA puede ser incorrecto.**

En nuestras pruebas de rendimiento, hemos obtenido un valor crítico de z de 0.552769703 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 (α, 1.00], esto indica que los cambios no produjeron mejoras relevantes; es decir, aunque haya variaciones en los tiempos observados, en términos generales se consideran equivalentes. Dado que el valor crítico de z obtenido cae dentro de este rango, concluimos que no existe una mejora significativa en el rendimiento al comparar los tiempos de respuesta entre las pruebas.

En resumen, las diferencias detectadas no son lo suficientemente notables como para considerarlas estadísticamente significativas con un nivel de confianza del 95%. Por lo tanto, no tiene sentido realizar una comparación entre las medias de tiempo de ejecución de las baterías de pruebas realizadas en un ordenador frente al 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.