Universidad de Sevilla  
Escuela Técnica Superior de Ingeniería Informática



Grado en Ingeniería Informática – Ingeniería del Software  
Diseño y Pruebas II

Curso 2024 – 2025

**Informe de testing**

**Grupo:** C2.030

**Número de estudiante dentro del grupo:** 2

**Repositorio**: <https://github.com/dedece65/Acme-ANS-C2>

**Fecha**: 03/07/2025

|  |  |
| --- | --- |
| **Alumno** | **Correo** |
| del Castillo Piñero, Daniel | dandelpin@alum.us.es |

**Historial de versiones**

|  |  |  |
| --- | --- | --- |
| **Versión** | **Fecha** | **Descripción de los cambios** |
| V1.0 | 02/07/2025 | Creación del informe |
| V1.1 | 03/07/2025 | Elaboración y revisión del informe |
|  |  |  |

**Índice de contenido**

[**1.** **Resumen ejecutivo** 2](#_Toc167735368)

[**2.** **Introducción** 2](#_Toc167735369)

[**3.** **Contenido** 3](#_Toc167735370)

[**3.1.** **Testing funcional** 3](#_Toc167735371)

[**3.1.1.** **Funcionalidad para los managers sobre los proyectos** 3](#_Toc167735372)

[**3.1.1.1.** **List-Mine** 3](#_Toc167735373)

[**3.1.1.2.** **Show** 4](#_Toc167735374)

[**3.1.1.3.** **Create** 4](#_Toc167735375)

[**3.1.1.4.** **Update** 5](#_Toc167735376)

[**3.1.1.5.** **Delete** 5](#_Toc167735377)

[**3.1.1.6.** **Publish** 6](#_Toc167735378)

[**3.1.2.** **Funcionalidad para los managers sobre las historias de usuario** 6](#_Toc167735379)

[**3.1.2.1.** **List-Mine** 6](#_Toc167735380)

[**3.1.2.2.** **List** 7](#_Toc167735381)

[**3.1.2.3.** **Show** 7](#_Toc167735382)

[**3.1.2.4.** **Create** 8](#_Toc167735383)

[**3.1.2.5.** **Update** 8](#_Toc167735384)

[**3.1.2.6.** **Delete** 9](#_Toc167735385)

[**3.1.2.7.** **Publish** 9](#_Toc167735386)

[**3.1.3.** **Funcionalidad para los managers sobre la relación entre proyectos e historias de usuario** 10](#_Toc167735387)

[**3.1.3.1.** **Create** 10](#_Toc167735388)

[**3.1.3.2.** **Delete** 11](#_Toc167735389)

[**3.2.** **Testing de rendimiento** 12](#_Toc167735390)

[**3.2.1.** **Sin índices – Ordenador estudiante 1** 12](#_Toc167735391)

[**3.2.2.** **Con índices – Ordenador estudiante 1** 13](#_Toc167735392)

[**3.2.3.** **Con índices – Ordenador estudiante 4** 15](#_Toc167735393)

[**4.** **Conclusiones** 17](#_Toc167735394)

[**Bibliografía** 18](#_Toc167735395)

# **Resumen ejecutivo**

Este informe presenta los resultados detallados de las pruebas funcionales y de rendimiento realizadas en el proyecto. En el apartado de pruebas funcionales, se describen los casos de prueba implementados. Cada caso de prueba incluye una descripción clara y una evaluación de su efectividad en la detección de errores. En el capítulo de pruebas de rendimiento, se presentan gráficos detallados y un intervalo de confianza del 95% para el tiempo de respuesta del sistema al procesar solicitudes con la aplicación de índices y sin ellos. Además, se incluye un análisis comparativo con un nivel de confianza del 95% para determinar estadísticamente si los índices tienen impacto en el rendimiento de las pruebas. Este informe utiliza un enfoque sistemático y riguroso, asegurando que los resultados sean precisos y útiles para la evaluación del proyecto.

# **Introducción**

El presente documento tiene como objetivo detallar los procedimientos y resultados de las pruebas funcionales y de rendimiento llevadas a cabo en el proyecto. En primer lugar, se aborda el apartado de pruebas funcionales, donde se describen los casos de prueba diseñados para verificar las diversas características del sistema. Cada uno de los casos de prueba incluye una descripción detallada y una evaluación de su efectividad en la detección de errores. Esta sección proporciona una visión clara y estructurada del desempeño del sistema bajo diferentes condiciones de prueba, permitiendo identificar y corregir fallos de manera efectiva.

En segundo lugar, el capítulo de pruebas de rendimiento se centra en el análisis del tiempo de respuesta del sistema al procesar solicitudes durante las pruebas funcionales. Se incluyen gráficos detallados que ilustran los tiempos de respuesta con la implementación de índices y sin ellos, junto con un intervalo de confianza del 95% para estos tiempos. Además, se lleva a cabo un contraste de hipótesis con un nivel de confianza del 95% para determinar si la implementación de índices tiene impacto en el rendimiento. El enfoque utilizado en este informe es metódico y sistemático, utilizando herramientas de planificación y Eclipse para garantizar la precisión y efectividad en la detección de errores durante las pruebas funcionales.

# **Contenido**

# **Testing funcional**

En esta sección vamos a explicar los casos de pruebas (positivos, negativos y de hacking) que hemos implementado para cada funcionalidad del sistema. Además, vamos a ofrecer una explicación de como de efectivo es para detectar errores, ayudados por la cobertura que nos aporta la herramienta Eclipse la cual nos indica el porcentaje de instrucciones que ha sido ejecutadas.

# **Funcionalidades de flight crew member sobre flight assignments**

Vamos a comenzar por las pruebas establecidas para las funcionalidades de los flight crew members sobre los flight assignments.

# **List**

Esta funcionalidad consiste en que un flight crew member pueda listar los flight assignments completados y planificados por separado.

Para realizar las pruebas positivas y negativas de esta funcionalidad nos hemos logueado como un member y hemos listado sus flight assignments completados y planificados.

Para realizar las pruebas de hacking, con el rol anónimo hemos introducido al url de los listados comprobando que nos saltaba un error de autorización y hemos repetido el mismo proceso con otro member para que nos saltara de nuevo otro error de autorización.

Tras realizar estas pruebas hemos obtenido en el servicio un 100% de cobertura en ambas listas lo cual nos asegura que la funcionalidad no esconde ningún posible fallo ya que ha ejecutado todas las instrucciones según lo esperado.

# **Show**

Esta funcionalidad consiste en que un customer pueda mostrar los detalles de un flight assignment.

Para realizar las pruebas positivas y negativas de esta funcionalidad nos hemos logueado como member, hemos mostrado el listado de sus flight assignments y hemos ido pulsando en alguno comprobando que aparecía la información correcta.

Para realizar las pruebas de hacking, hemos intentado acceder a la información de un proyecto con un rol distinto al del member (anónimo) y luego hemos intentado acceder como un member distinto al que nos habíamos logueado. En todos los casos han saltado los errores correspondientes.

Tras realizar estas pruebas hemos obtenido un 100% de cobertura asegurándonos de esta manera que no van a existir errores ya que se han ejecutado todas las líneas de código.

# **Create**

Esta funcionalidad consiste en que un member puede crear un flight assignment rellenando un formulario con los detalles necesarios y dentro de las restricciones impuestas.

Para realizar las pruebas positivas y negativas, hemos empezado primero por las

negativas, mandando el formulario en blanco y luego rellenando cada campo del formulario con datos negativos varios, haciendo que salten todas las restricciones posibles. Tras probar todos los casos negativos posibles y probar que los casos positivos no hacen que salte ningún error, hemos creado un flight assignment con valores estándar.

Para realizar las pruebas de hacking, hemos probado a introducir la url para crear proyecto con un rol distinto al de member (anónimo), comprobando que no dejaba acceder, y hemos probado a modificar con el menú de inspección los elementos de sólo lectura y desplegables para posteriormente crear el flight assignment y comprobar que no han sido modificados y han conservado el valor que tenían antes de la modificación.

Tras realizar estas pruebas hemos obtenido un 99,5% de cobertura, tras inspeccionar en eclipse se ha comprobado que el porcentaje de cobertura que faltaba se corresponde con las siguientes líneas marcadas en amarillo:



En este caso “authorised2” se inicializa como true y sólo cambia si una leg no existe en el repositorio, no se ha podido replicar este error en el testeo pero creemos que no presenta ningún fallo en la aplicación.

Logotipo

El contenido generado por IA puede ser incorrecto.

En este caso tenemos una línea que se usa como método privado para comprobar que ninguna leg que se seleccione es incompatible con las legs asociadas al flight crew member, la validación para la que se utiliza esta línea está marcada en verde y se desconoce el motivo de la línea en amarillo.



Se desconoce el motivo de esta línea en amarillo ya que durante el testeo podemos ver el mensaje de validación asociado a la misma.



Se piensa que el motivo de esta última línea es alguna parte del paréntesis que no se llega a acceder, de nuevo el funcionamiento de la validación se ha comprobado y es correcto.

.

# **Update**

Esta funcionalidad consiste en que un member pueda actualizar uno de sus flight assignments siempre y cuando este flight assignment no haya sido publicado.

Para realizar las pruebas positivas y negativas, hemos llevado una metodología muy parecida a la llevada en el create solo que en el caso de las positivas en vez de crear un flight assignment hemos ido actualizándolo.

Para realizar las pruebas de hacking, hemos intentado actualizar un assignment con un rol diferente al de member (anónimo), después nos hemos logueado con un member pero con un usuario distinto al del assignment que hemos intentado actualizar. Por último, nos hemos logueado con un member y con el usuario adecuado, pero hemos intentado actualizar un assignment que ya está publicado y publicar cambiando los desplegables y campos de solo lectura mencionados anteriormente. En todos los casos el sistema ha funcionado como esperábamos, rechazando tales peticiones.

Tras realizar estas pruebas hemos obtenido un 99,4% de cobertura donde las líneas amarillas son o las mismas que en el método anterior u otras que tienen la misma explicación.

# **Delete**

Esta funcionalidad consiste en que un member pueda borrar sus assignments siempre y cuando estos no hayan sido publicados.

Para realizar las pruebas positivas y negativas, hemos primero probado a borrar un assignment no publicado. Es decir, el único caso positivo, ya que los casos negativos no están permitidos porque no existe autorización para borrar un assignment que este publicado (no damos la opción del botón de borrar si esta publicado).

Para realizar las pruebas de hacking han seguido la misma línea que el Update, intentar borrar assignments con el rol inadecuado, intentar borrarlo con el mismo rol pero usuario inadecuado (tanto publicados como no publicados) e intentar borrar un assignment publicado con el rol y usuario adecuado. En todos los casos el acceso ha sido denegado.

En este caso, tras realizar las pruebas hemos obtenido un 97,5% de cobertura donde se repite el mismo caso de líneas amarillas y aparece una línea marcada en roja:



Que se debe a que al borrar no se accede al método unbind ya que no hace falta guardar ningún dato.

# **Publish**

Esta funcionalidad consiste en que un member pueda publicar uno de sus assignments.

Para llevar a cabo las pruebas positivas y negativas, primero hemos intentado publicar assignment probando casos inválidos (es decir, pruebas negativas de la misma forma que en el Update). Tras comprobar que en todos estos casos nos han dado los errores correspondientes, hemos procedido a publicar un assignment correcto.

Para llevar realizar las pruebas de hacking, hemos intentado realizarlo con un rol distinto al de member, luego con el rol de member pero con el usuario inadecuado y, por último, con el rol y usuario correcto hemos intentado publicar assignments que ya estaban publicados, comprobando también la modificación de los desplegables y campos de solo lectura mencionados anteriormente. En todos los casos se nos ha denegado el acceso, por tanto, la aplicación ha funcionado como esperábamos.

Tras realizar estas pruebas hemos obtenido un 99,6% de cobertura repitiendo las mismas líneas amarillas.

# **Funcionalidad de flight crew member sobre activity logs**

Vamos a continuar por las pruebas establecidas para las funcionalidades de los members sobre los activity logs.

# **List**

Esta funcionalidad consiste en que un member pueda ver todos los activity logs asociados a un flight assignment completado. Para realizar los casos de prueba positivos y negativos, nos hemos logueado como un member y hemos accedido al listado de sus assignments completados, allí hemos comprobado que se listan todos sus activity logs. Además, hemos accedido a un assignment planeado para comprobar que no se da la opción de listar activity logs.

Para realizar los casos de prueba de hacking, nos hemos logueado con un rol inadecuado intentado obtener un listado de activity logs. A parte de esto se ha probado a acceder al listado de pasajeros de un assignment cuando se estaba logueado con otro rol y con un usuario que no era propietario de este y cuando el identificador del flight assignment asociado se correspondía con uno planeado, respondiendo la aplicación como esperábamos.

Tras realizar estas pruebas hemos obtenido un 99,3% de cobertura repitiéndose de nuevo el mismo patrón de líneas amarillas.

# **Show**

Esta funcionalidad consiste en que un member pueda mostrar los detalles de un activity log.

Para realizar las pruebas positivas y negativas de esta funcionalidad nos hemos logueado como member, hemos mostrado el listado de activity logs de un flight assignment completado y hemos ido pulsando en algunos comprobando que aparecía la información correcta.

Para realizar las pruebas de hacking, hemos intentado acceder a la información de un activity log con un rol distinto al del member (anónimo) y luego hemos intentado acceder como un member pero con el usuario incorrecto, es decir, al activity log de un member distinto al que nos hemos logueado. En todos los casos han saltado los errores correspondientes.

Tras realizar estas pruebas hemos obtenido un 99,4% de cobertura con las mismas líneas amarillas.

# **Create**

Esta funcionalidad consiste en que un member pueda crear un activity log. Para realizar las pruebas positivas y negativas, hemos empezado primero por las negativas, mandando el formulario en blanco y luego probando todos los fallos distintos posibles que podían darse para cada atributo. Tras probar todos los casos negativos posibles, hemos creado un activity log con valores posibles como se ha explicado en la metodología de clase (probando los extremos de las restricciones, los diferentes valores que puede tomar…), siempre siendo valores permitidos.

Para realizar las pruebas de hacking, hemos probado a introducir la url para crear un pasajero con un rol distinto al de member (anónimo), comprobando que no dejaba acceder y con un member distinto para comprobar que se asociaba al usuario correcto y nos hemos logueado con el member correcto para intentar hackear sin éxito el campo de solo lectura.

Tras realizar estas pruebas hemos obtenido un 98,6% de cobertura, esta vez con una línea nueva en amarillo en el método unbind:



Como se ha mencionado, se ha creado un activity log con valores posibles y comprobado posteriormente, entre otras cosas esta línea, y funciona todo correctamente.

# **Update**

Esta funcionalidad consiste en que un member pueda actualizar un activity log.

Para realizar las pruebas positivas y negativas, hemos llevado una metodología muy parecida a la llevada en el Create solo que en el caso de las positivas en vez de crear hemos actualizado el activity log.

Para realizar las pruebas de hacking, hemos intentado actualizar un pasajero con un rol diferente al de member (anónimo), después nos hemos logueado con un member pero con un usuario distinto al del activity log que hemos intentado actualizar. Por último, nos hemos logueado con un member y con el usuario adecuado, pero hemos intentado actualizar un activity log que ya estaba publicado además de probar a modificar la restricción de solo lectura. En todos los casos han funcionado el sistema como esperábamos, rechazando tales peticiones.

Tras realizar estas pruebas hemos obtenido un 97,5% de cobertura con la aparición de las siguientes líneas en amarillo y rojo:

Captura de pantalla de un celular

El contenido generado por IA puede ser incorrecto.

Este caso se da ya que no se ha conseguido replicar todos los comportamientos contemplados y no se ha accedido a ningún return, en cualquier caso y como ocurría en el punto 3.1.1.3 (Create de flight assignment) no se cree que presente ningún problema ni fallo en la aplicación.

# **Delete**

Esta funcionalidad consiste en que un member pueda borrar un activity log.

Para realizar las pruebas positivas y negativas, hemos borrado un activity log no publicados ya que no se da la opción de borrar uno que sí lo está.

Para realizar las pruebas de hacking han seguido la misma línea que el Update, intentar borrar activity logs con el rol inadecuado, intentar borrarlo con el mismo rol pero usuario inadecuado (tanto publicados como no publicados) e intentar borrar un activity log publicado con el rol y usuario adecuado. En todos los casos el acceso ha sido denegado.

Tras realizar estas pruebas hemos obtenido un 94,5% de cobertura debido a la aparición de 2 líneas rojas, una de ellas igual a la del punto 3.1.1.5 (Delete de flight assignment) y la siguiente que se debe a no acceder nunca al método GET:

Texto, Carta

El contenido generado por IA puede ser incorrecto.

# **Publish**

Esta funcionalidad consiste en que un member puede publicar un activity log. Para realizar las pruebas positivas y negativas, hemos primero probado a publicar activity logs modificando campos para que dieran los errores correspondientes de forma similar al Publish de flight assignment.

Para llevar realizar las pruebas de hacking, hemos intentado realizar la acción con roles que no eran el correcto, luego con el rol de member pero con el usuario inadecuado y, por último, con el rol y usuario correcto hemos intentado publicar activity logs que ya estaban publicados además de modificar la restricción de solo lectura. En todos los casos se nos ha denegado el acceso, por tanto, la aplicación ha funcionado como esperábamos.

Tras realizar estas pruebas hemos obtenido un 98,1% de cobertura apareciendo las mismas línea amarillas que se vienen explicando y las líneas rojas mencionadas en el punto 3.1.2.4 (Update de activity log).

# **Testing de rendimiento**

Nos centramos ahora en el análisis del tiempo de respuesta del sistema al procesar solicitudes durante las pruebas funcionales. Se llevarán a cabo comparaciones entre nuestro portátil sin y con los índices para ver si estos han mejorado el rendimiento.

# **Sin índices**

Tras aplicar los procedimientos explicados en las diapositivas hemos obtenido los siguientes tiempos para las diversas funcionalidades. Además, mostramos un gráfico donde podemos ver cuáles son las funcionalidades que tardan más en ejecutarse.

Gráfico, Gráfico en cascada

El contenido generado por IA puede ser incorrecto.

Y hemos obtenido de los datos los siguientes resultados después del análisis estadístico:

Tabla

El contenido generado por IA puede ser incorrecto.

A continuación, en el siguiente apartado, veremos si los resultados con los índices ya incluidos han mejorado o no las prestaciones.

# **Con índices**

Estos son los tiempos obtenidos para las distintas funcionalidades además de un gráfico para resaltar cuales son las funcionalidades que tardan más en ejecutarse.

Gráfico, Gráfico en cascada

El contenido generado por IA puede ser incorrecto.

Como podemos observar el promedio general ha sido algo mayor que sin utilizar los índices lo cual ya puede darnos un indicio de que no han mejorado las prestaciones.

Y hemos obtenido de los datos los siguientes resultados después del análisis estadístico:

Tabla

El contenido generado por IA puede ser incorrecto.

**Comparación entre utilizar y no utilizar índices**

Nos disponemos a comparar los siguientes resultados con una prueba z para medias de dos muestras y llegamos a la siguiente tabla:

Interfaz de usuario gráfica, Aplicación, Tabla

El contenido generado por IA puede ser incorrecto.

Como se observa en *Valor crítico de z (dos colas)***,** el valor está en el intervalo [0.00, α], lo que indica que las diferencias en los tiempos medios entre las dos ejecuciones se pueden comparar. Se entiende mirando la media de tiempos después de los cambios que no se ha mejorado el rendimiento de la aplicación.

# **Conclusiones**

Este informe presenta los resultados de las pruebas funcionales y de rendimiento realizadas en el proyecto. Las pruebas funcionales detallan diversos casos implementados para verificar las funcionalidades del sistema, evaluando su efectividad en la detección de errores. Las pruebas de rendimiento se enfocan en el análisis del tiempo de respuesta del sistema al procesar solicitudes en dos ordenadores diferentes, utilizando gráficos detallados e intervalos de confianza del 95% para asegurar la precisión de los resultados.

La sección de pruebas funcionales describe casos de prueba positivos, negativos y de hacking para cada funcionalidad del sistema, utilizando herramientas como Eclipse para asegurar una alta cobertura de código. En ambos conjuntos de operaciones, sobre flight assignment y activity log, se ha conseguido una cobertura muy próxima al 100%, 99,5% y 98,2% respectivamente. Este análisis nos indica que nuestro código es robusto frente a ataques y que se ha evaluado la funcionalidad correcta de toda la aplicación.

Las pruebas de rendimiento compararon el tiempo de respuesta del sistema con y sin índices. Los resultados mostraron que el uso de índices afectó negativamente al rendimiento de la aplicación.

En resumen, el informe confirma la efectividad del sistema en términos de funcionalidad y rendimiento. Estos resultados proporcionan una base sólida para la evaluación y mejora continua del proyecto.

# **Bibliografía**

Intencionalmente en blanco.