I Universidad de Sevilla

Escuela Técnica Superior de Ingeniería Informática

**Testing Report**

****

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

**Curso 2024 – 2025**

|  |  |  |
| --- | --- | --- |
| **Grupo: C1.052** | |  |
| **Autores por orden alfabético** | **Roles** | **Correo Corporativo** |
| **Galea Magro, Carlos** | **Operator, Developer, Tester** | **cargalmag@alum.us.es** |

**Fecha: 26/05/2025(month/day/year)**

**Student#3**

**Introducción**

En este informe se recopilan los datos de pruebas funcionales y pruebas de rendimiento realizadas por parte de el Student 3 en el proyecto ACME\_ANS.

Las pruebas de funcionales recogen los casos de prueba realizados y comprueba su capacidad para detectar los fallos.

Las pruebas de rendimiento evalúan el tiempo de respuesta de los casos de prueba en distintas computadoras.

**1.Pruebas Funcionales**

**1.Casos de prueba de FlightAssignment**

**1-FlightAssignmentListCompletedService**

**/FlightAssignmentListPlannedService**

**-Descripción**: Prueba la funcionalidad de que un flightCrewMember puede listar sus FlightAssignment completadas y planeadas.

Para el hacking se prueba que un usuario no autenticado, con otro rol o un flightCrewMember distinto no pueda acceder a la lista de FlightAssignment que pertenece a un flightCrewMember concreto.

**-Resultado**: La cobertura del .hack y .safe son del 100% consiguiendo una cobertura total de 100% en esta feature. Todas las lineas de ambas clases estan en verde por lo que se ejecutó todo correctamente.

Texto

El contenido generado por IA puede ser incorrecto.

Texto

El contenido generado por IA puede ser incorrecto.

**2- FlightAssignmentShowService**

**-Descripción**: Prueba la funcionalidad de que un flightCrewMember pueda acceder a ver los datos de un FlightAssignment. Además para el hacking se prueba que un usuario no autenticado, con otro rol o un flightCrewMember distinto no pueda acceder a los datos de un FlightAssignment que pertenece a otro flightCrewMember.

**-Resultado**: La cobertura total es de 100% en esta feature. Todas las líneas estan en verde por lo que se ejecutó todo correctamente.

Texto

El contenido generado por IA puede ser incorrecto.

**3- FlightAssignmentCreateService**

**-Descripción**: Prueba la funcionalidad de que un flightCrewMember pueda crear un FlightAssignment correctamente con todos los posibles datos, correctos e incorrectos. Probando inicialmente todo vacío y después cada uno con el resto vacío. Además se prueba el hacking, un usuario no autenticado no pueda crear un FlightAssignment. También se comprueba que no se actualicen los readonly de la clase (el lastUpdateMoment y flightCrewMember). Las leg deben ser futuras para crear un assignment, no pueden ser nulas, no estar en modo borrador, es decir, un id de una feature en borrador mediante POST hacking y tampoco tener un id inexistente o un id perteneciente a otra feature.

**-Resultado**: La cobertura total es de 98% en esta feature. Las líneas verdes que son la mayoría se debe a que funciona bien pero encontramos dos líneas en amarillo:



Esta es porque no necesita aplicar en el test el método POST.



Esta otra es un assert innecesario que podría omitirse, ya que la condición de que el assignment es distinto de null siempre se cumple.

Texto

El contenido generado por IA puede ser incorrecto.Texto

El contenido generado por IA puede ser incorrecto.

**4- FlightAssignmentUpdateService**

**-Descripción**: Prueba la funcionalidad de que un flightCrewMember pueda actualizar un FlightAssignment usando todos los posibles datos, correctos e incorrectos. Probando inicialmente todo vacío y después cada uno con el resto vacío.

En el hacking se prueba que un usuario no autenticado, con otro rol o un flightCrewMember distinto no pueda actualizar un FlightAssignment. También se comprueba que no se actualicen los readonly de la clase (el lastUpdateMoment y flightCrewMember)

Con respecto al atributo de las leg, deben ser futuras para crear un assignment, no pueden ser nulas, no pueden estar en modo borrador, es decir, tener un id de una leg en borrador(buscado en la base de datos) y tampoco tener un id inexistente o un id perteneciente a otra feature. Todos se ha probado uno a uno.

Otro hack es en un FlightAssignment publicado no se pueda actualizar mediante la url.

**-Resultado**: La cobertura total es de 100% en esta feature. Todas las líneas estan en verde por lo que se ejecutó todo correctamente.

Texto

El contenido generado por IA puede ser incorrecto.

Texto

El contenido generado por IA puede ser incorrecto.

**5- FlightAssignmentPublishService**

**-Descripción**: Prueba la funcionalidad de que un flightCrewMember pueda actualizar un FlightAssignment. Además se prueba el hacking, un usuario no autenticado, con otro rol o un flightCrewMember distinto no pueda actualizar un FlightAssignment. También se comprueba que no se actualicen los readonly de la clase (el lastUpdateMoment y flightCrewMember).

Con respecto a las leg, deben ser futuras para crear un assignment, no pueden ser nulas, no pueden estar en modo borrador, es decir, tener un id de una leg en borrador(buscado en la base de datos) y tampoco tener un id inexistente o un id perteneciente a otra featureOtro hack es en un FlightAssignment publicado no se pueda publicar mediante la url. Las validaciones también son comprobadas una a una.

**-Resultado**: La cobertura total es de 100% en esta feature. Todas las líneas estan en verde por lo que se ejecutó todo correctamente. Texto

El contenido generado por IA puede ser incorrecto.Texto

El contenido generado por IA puede ser incorrecto.Texto

El contenido generado por IA puede ser incorrecto.Texto

El contenido generado por IA puede ser incorrecto.

**2.Casos de prueba de ActivityLog**

**1-ActivityLogListService**

**-Descripción**: Prueba la funcionalidad de que un flightCrewMember puede listar los activityLog pertenecientes a sus correspondientes flightAssignments. Además se prueba que un usuario no autenticado, con otro rol o un flightCrewMember distinto no pueda acceder a la lista de activityLogs los cuales pertenecen a un flightAssignment de otro flightCrewMember.

**-Resultado**: La cobertura total es de 100% en esta feature. Todas las líneas estan en verde por lo que se ejecutó todo correctamente.

Texto

El contenido generado por IA puede ser incorrecto.

**2-ActivityLogShowService**

**-Descripción**: Prueba la funcionalidad de que un flightCrewMember pueda acceder a ver los datos de un activityLog de un flightAssignment de este flightCrewMember. Además se prueba que un usuario no autenticado, con otro rol o un flightCrewMember distinto no pueda acceder a los datos de un activityLog que pertenece a otro flightCrewMember.

**-Resultado**: La cobertura total es de 100% en esta feature. Todas las líneas estan en verde por lo que se ejecutó todo correctamente.

Texto

El contenido generado por IA puede ser incorrecto.

**3-ActivityLogCreateService**

**Descripción**: Prueba la funcionalidad de que un flightCrewMember pueda crear un activityLog correctamente con todos los posibles datos, probando inicialmente todo vacío y después cada uno con el resto vacío. Además se prueba el hacking, un usuario no autenticado, un usuario con otro rol, un member que no sea el que contiene el FlightAssignment del activityLog no pueda crear un FlightAssignment. También se comprueba que no se actualicen los readonly de la clase (el registrationMoment).

**-Resultado**: La cobertura total es de 100% en esta feature. Todas las líneas estan en verde por lo que se ejecutó todo correctamente. Texto

El contenido generado por IA puede ser incorrecto.

**4-ActivityLogUpdateService**

**Descripción**: Prueba la funcionalidad de que un flightCrewMember pueda actualizar un activityLog correctamente con todos los posibles datos, probando inicialmente todo vacío y después cada uno con el resto vacío. Además se prueba el hacking, un usuario no autenticado, un usuario con otro rol o un member que no sea el que contiene el FlightAssignment del activityLog no pueda actualizar un FlightAssignment. También se comprueba que no se actualicen los readonly de la clase (el registrationMoment) y que en un activityLog publicado no se pueda actualizar con la url.

**-Resultado**: La cobertura total es de 100% en esta feature. Todas las líneas estan en verde por lo que se ejecutó todo correctamente.

Texto

El contenido generado por IA puede ser incorrecto.

**5-ActivityLogPublishService**

**Descripción**: Prueba la funcionalidad de que un flightCrewMember pueda publicar un activityLog correctamente con todos los posibles datos, probando inicialmente todo vacío y después cada uno con el resto vacío. Además se prueba el hacking, un usuario no autenticado, un usuario con otro rol o un member que no sea el que contiene el FlightAssignment del activityLog no pueda publicar un FlightAssignment. También se comprueba que no se actualicen los readonly de la clase (el registrationMoment) y que en un activityLog publicado no se pueda publicar con la url. Las validaciones también se comprueban, solo hay una y es que no pueda publicarse el activityLog si no se ha publicado su flightAssignment.

**-Resultado**: La cobertura total es de 100% en esta feature. Todas las líneas estan en verde excepto una, la cual no es verde porque no ejecutaría alguna condición aún así esta línea si que esta verde en otras features y teniendo un 100% alomejor fue un fallo que no debería ocurrir.

Texto

El contenido generado por IA puede ser incorrecto.

Aquí finalmente proporciono las capturas de la cobertura de todas las pruebas de las features:

Interfaz de usuario gráfica, Texto

El contenido generado por IA puede ser incorrecto.

Imagen de la pantalla de un computador

El contenido generado por IA puede ser incorrecto.

**Conclusión de Pruebas Funcionales**

Para concluir podemos aclarar que el coverage está muy completo en general para las features implementadas, aclarando que las pruebas realizadas fueron muy completas tras probar todos los atributos con todos sus posibles valores y validaciones de las features, además de todos los hacking vistos en la asignatura .

**2.Pruebas de Rendimiento**

Ahora mediante las pruebas de rendimiento voy a analizar los tiempos de respuesta y comparar las distintas pruebas realizadas comprobando así si al hacer cambios para mejorar han resultado con una mejora real o no.

La metodología para extraer los datos ha sido usando el replayer de eclipse a los casos de prueba lo cual proporciona una traza general que puede ser analizada usando excel mediante métodos detallados en clase.

**Resultados de las Pruebas**

**Prueba 1:** La primera prueba se realizó sin índices.

**Gráfico, Gráfico de barras

El contenido generado por IA puede ser incorrecto.**

**Tabla

El contenido generado por IA puede ser incorrecto.** ****

Se puede observar en el gráfico que las pruebas en general no sobrepasan los 50ms lo que está muy bien, sin embargo el flightAssignmentPublishService si que es la más lenta y problemática, esto se debe a que tiene mayor cantidad de código y no está todo optimizado de la forma óptima. Esta feature contiene gran cantidad de validaciones y por esto se refleja así en el gráfico.

**Prueba 2:** La primera prueba se realizó con índices.

Gráfico, Gráfico en cascada

El contenido generado por IA puede ser incorrecto.

Interfaz de usuario gráfica, Tabla

El contenido generado por IA puede ser incorrecto. 

En este gráfico se puede ver que es muy similar al anterior, simplemente los tiempos bajaron un poco.

Con los datos recopilados se realizó el siguiente Z-alpha-test:

Interfaz de usuario gráfica, Aplicación, Tabla

El contenido generado por IA puede ser incorrecto.

Tabla

El contenido generado por IA puede ser incorrecto.

Sabiendo que el alpha es 0.05, los cambios se consideran significativos si el p-value obtenido es menor al alpha. Como nuestro p-value es 0.41399 se puede decir que los cambios no son significativos ya que pertenece al intervalor (0.05 , 1.00].

Los cambios son visibles pero no significativos.

**Conclusión de Pruebas de Rendimiento**

Finalmente podemos decir que no se ha tenido una mejora notable en el rendimiento tras la optimización aplicando los índices. Si es verdad que los tiempos de respuesta son menores tras aplicar índices y hay una mejora, pero tras el análisis Z podemos decir que no es una mejora significativa.