# Testing report

Grupo C1.054 | Diseño y Pruebas II | 26/05/2025

Versiones

|  |  |  |
| --- | --- | --- |
| Fecha | Versión | Descripción |
| 26/05/2025 | 1.0 | Creación plantilla |
| 26/05/2025 | 2.0 | Testing Report Completado |

Miembros:

Adrián Díaz Vázquez (adrdiavaz@alum.us.es)

Esteban López Pérez (estlopper@alum.us.es)

Jesús Pons Morís ([jesponmor@alum.us.es](mailto:jesponmor@alum.us.es)) - Autor

Kevin Amador Calzadilla (kevamacal@alum.us.es)

Manuel Zurita Fernández (manzurfer@alum.us.es)

Repositorio: <https://github.com/DP2-C1-054/Acme-ANS-D04>

# Índice:

Resumen ejecutivo………………………………………………………………………1

Introducción……………………………………………………………………………..2

Contenido………………………………………………………………………………..5

Pruebas funcionales……………………………………………………………...5

Análisis de rendimiento………………………………………………………...12

Conclusiones…………………………………………………………………………....15

Bibliografía……………………………………………………………………………..16

# Resumen ejecutivo

A lo largo de este informe se dará información detallada sobre las pruebas funcionales realizadas tanto positivas, negativas como de hacking. Además, su respectivo análisis y comparativa de rendimiento del código anterior y posterior a la refactorización y del uso de los índices para ver su impacto en el rendimiento de Acme Ans.

# Introducción

En este informe se pretende explicar como se ha realizado las pruebas funcionales, explicando para cada archivo de test que se ha realizado resumidamente y aportando la cantidad de código cubierto en este caso con respecto a los requisitos funcionales 8 y 9 del Estudiante 2.  
  
También se pretende explicar el análisis realizado al código anteriormente cubierto por las pruebas funcionales y explicar cómo se ha optimizado en caso de ser necesario para obtener un mejor rendimiento. Para ello es importante la realización y comparación de un segundo análisis con respecto al primer análisis realizado sin cambios y determinar si los cambios han sido significativos.

# Contenido

## Pruebas funcionales

Pruebas de Booking:

* create.safe: Pruebas positivas y negativas del servicio CustomerBookingCreateService desde la vista de un cliente que le da un uso normal a la aplicación usando datos de prueba positivos y negativos para comprobar validadores.
* create.hack: Pruebas de hacking del servicio CustomerBookingCreateService, accediendo desde urls que no nos proporciona de forma natural Acme Ans como por ejemplo intentar entrar con otro real e intentando introducir datos maliciosos con el modo inspeccionar del navegador.
* delete.safe: Pruebas positivas y negativas del servicio CustomerBookingDeleteService, probar a borrar una reserva que se nos permita (modo borrador).
* delete.hack: Pruebas de hacking del servicio CustomerBookingDeleteService, get hacking, intent de borrar una reserva ya publicada mediante url y pruebas de acceso no autorizados con otro tipo de usuario o clientes distintos al propietario de la reserva.
* list&show.safe: Pruebas positivas y negativas de los servicios CustomerBookingListService y CustomerBookingShowService, visualización de la lista de reservas y de reservas con distintos valores, publicadas, sin publicar, con pasajeros sin pasajeros, etc.
* list&show.hack: Pruebas de hacking de los servicios CustomerBookingListService y CustomerBookingShowService, intent de acceder a listas de reservas con otro tipo de usuario e intento de visualización de reservas pertenecientes a un cliente con un customer distinto o con otro tipo de usuario
* publish.safe: Pruebas positivas y negativas del servicio CustomerBookingPublishService, intento de publicar una reserva con datos positivos y negativos para ver el correcto funcionamiento de validadores y restricciones.
* publish.hack: Pruebas de hacking del servicio CustomerBookingPublishService intentos de publicación a una reserva ya publicada o sin pasajeros, iniciando sesión con otros usuarios que no deberán tener acceso y publicar con datos inválidos introducidos desde el modo inspeccionar.
* update.safe: Pruebas positivas y negativas del servicio CustomerBookingUpdateService, intento de actualizar una reserva no publicada con datos válidos e inválidos para probar los validadores y el correcto funcionamiento de la aplicación.
* update.hack: Pruebas de hacking del servicio CustomerBookingUpdateService, Intentos de actualizar una booking publicada, de actualizar con otro usuario distinto al customer propietario, introduciendo valores con el modo inspeccionar del navegador que de forma normal no podríamos.

A screenshot of a graph

AI-generated content may be incorrect.

Pruebas de Passenger:

* create.safe: Pruebas positivas y negativas del servicio CustomerPassengerCreateService, desde la vista de un cliente que le da un uso normal a la aplicación usando datos de prueba positivos y negativos para comprobar validadores de la entidad pasajero.
* create.hack: Pruebas de hacking del servicio CustomerPassengerCreateService, intento de acceder a crear un pasajero sin tener la sesión iniciada o iniciando sesión con otro tipo de usuario.
* delete.safe: Pruebas positivas y negativas del servicio CustomerPassengerDeleteService, intentar borrar un pasajero no publicado.
* delete.hack: Pruebas de hacking del servicio CustomerPassengerDeleteService, intentar borrar un pasajero publicado, borrar un pasajero de otro customer, borrar un pasajero sin tener la sesión iniciada o iniciada con otro tipo de usuario.
* list&show.safe: Pruebas positivas y negativas de los servicios CustomerPassengerListService y CustomerPassengerShowService, visualizar distintos pasajeros (publicados y sin publicar) desde distintas reservas.
* list&show.hack: Pruebas de hacking de los servicios CustomerPassengerListService y CustomerPassengerShowService, intento de visualización sin estar autorizado como con distintos tipos de usuarios, con un cliente distinto al que creo el pasajero o sin tener la sesión iniciada.
* publish.safe: Pruebas positivas y negativas del servicio CustomerPassengerPublishService, intentos de publicar entidad pasajero con datos válidos e inválidos para comprobar los validadores.
* publish.hack: Pruebas de hacking del servicio CustomerPassengerPublishService, intentando publicar un pasajero publicado, publicar un pasajero que no pertenece al cliente que ha iniciado sesión, publicar un pasajero sin tener la sesión iniciada o con otro tipo de usuario.
* update.safe: Pruebas positivas y negativas del servicio CustomerPassengerUpdateService, intentos de actualizar entidad pasajero con datos válidos e inválidos para comprobar los validadores.
* update.hack: Pruebas de hacking del servicio CustomerPassengerUpdateService, intentando actualizar un pasajero publicado, actualizar un pasajero que no pertenece al cliente que ha iniciado sesión, actualizar un pasajero sin tener la sesión iniciada o con otro tipo de usuario.

A screen shot of a graph

AI-generated content may be incorrect.

Pruebas de Takes:

* create.safe: Pruebas positivas y negativas del servicio CustomerPassengerCreateService crear todas las relaciones entre Pasajero y cliente posible.
* create.hack: Pruebas de hacking del servicio CustomerPassengerCreateService, se intento crear una relación desde un usuario no autorizado, usando pasajeros que no pertenecen al customer.
* delete.safe: Pruebas positivas y negativas del servicio CustomerPassengerDeleteService, intentando eliminar relaciones correctamente.
* delete.hack: Pruebas de hacking del servicio CustomerPassengerDeleteService, intentando eliminar relaciones de una booking ya publicada, intentando eliminar una relación inexistente entre una reserva y un pasajero, intentando eliminar relaciones existentes pertenecientes a otro usuario, con la sesión no iniciada o inicando sesión con otro tipo de usuario.

A screenshot of a graph

AI-generated content may be incorrect.

## Análisis de rendimiento

Análisis 1:

Tras realizar los test obtenemos la siguiente gráfica donde podemos ver que los MIR estarían relacionados con el créate,publish,el show y el update de las reservas:

Obteniendo los siguientes intervalos y nivel de confianza:

A screenshot of a spreadsheet

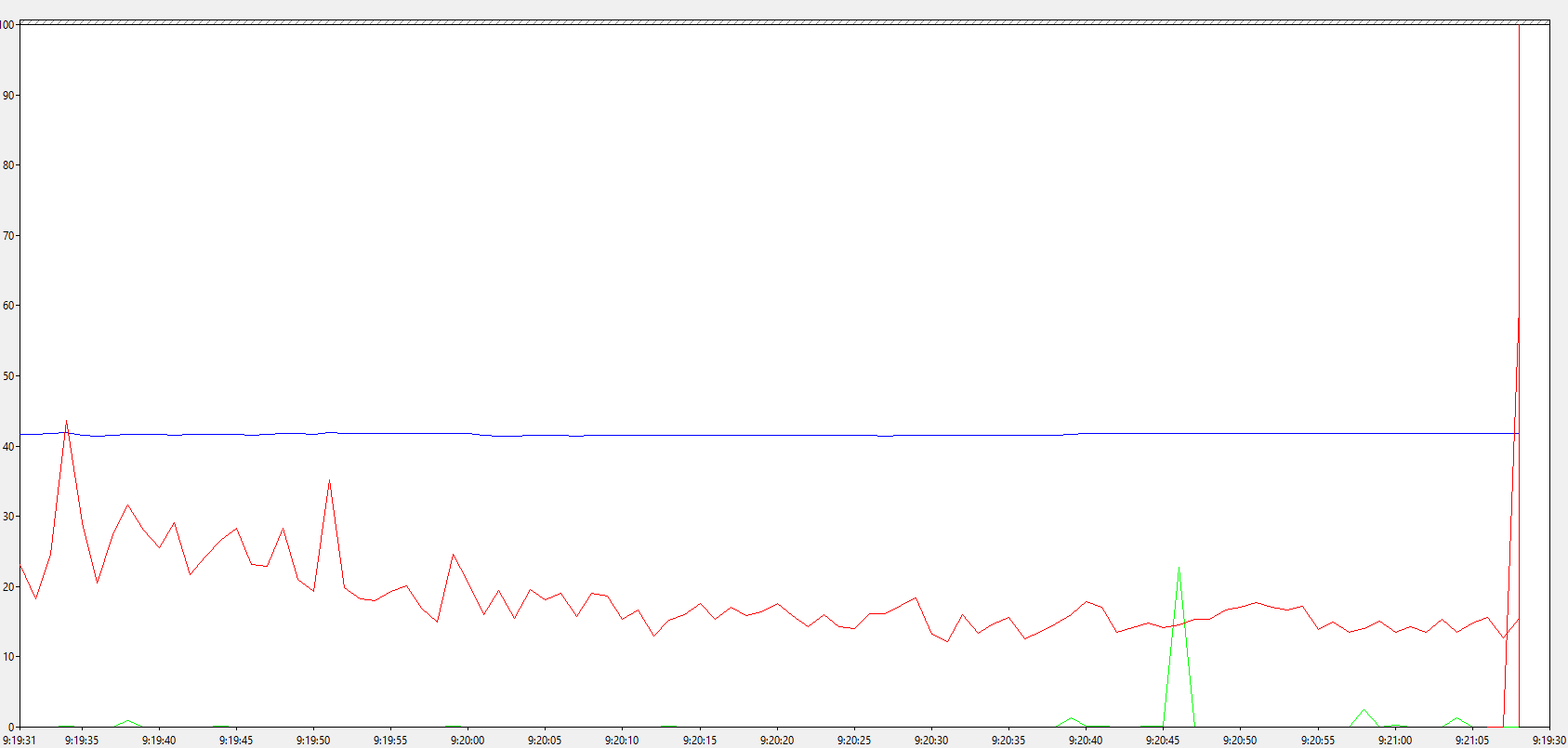
AI-generated content may be incorrect.

Al realizar el software profiling se obtuvo los siguientes datos:

A screenshot of a computer

AI-generated content may be incorrect.

Como podemos apreciar el problema estaba en la función bind, concretamente en el stream para obtener los vuelos disponibles que consumía mucho tiempo:  
A screen shot of a video

AI-generated content may be incorrect.  
Con respecto al hardware profiling aun sin estar optimizado no se apreció un cuello de botella importante:  




Análisis 2:

Gracias al análisis 1 se hizo una refactorización de código eliminando el stream remplazándolo por una query personalizada y se añadieron índices para optimizar otras queries.

Se redujo considerablemente el tiempo para las peticiones de las reservas:

Se obtuvo los siguientes intervalos y nivel de confianza:

A screenshot of a spreadsheet

AI-generated content may be incorrect.

Al realizar el software profiling se obtuvo los siguientes datos:

A screenshot of a computer

AI-generated content may be incorrect.

Viendo tanto el software profilling como los intervalos y nivel de confianza se puede notar más la mejora tras los arreglos.

Comparativa:

A table with numbers and a few digits

AI-generated content may be incorrect.

Se puede ver que el valor de two-tail p-value está en el intervalo 0-0,05 por lo que ha habido una mejora significativa real.

# Conclusiones

Tras la realización de las pruebas se ha obtenido una cobertura más del 99% en las features 8 y 9 asociadas al Student 2, ayudando a encontrar bugs y posibles hackeos que han ayudado al cumplimiento de la ley orgánica 3/2018. Además, el análisis de rendimiento a permitido ver donde el código era menos optimo y ofrecer una alternativa que además podemos confirmar que ha sido optima comparada con la anterior gracias al z-Test entre el antes y el después.

# Bibliografía

Intencionalmente en blanco.