

TESTING REPORT



Repository: <https://github.com/marrivbec/gii-is-DP2-C1.033>

Members:

- Jesús Villalba Fernández – jesvilfer@alum.us.es
- Juan Ramírez Morales – juarammor2@alum.us.es
- Mario Rivas Becerra – marrivbec1@alum.us.es
- Álvaro Baladrón Caballero – alvbalcab@alum.us.es
- Ángel Manuel Ferrer Álvarez – angferalv@alum.us.es

Date: May 26, 2025

Contenido

Resumen ejecutivo.....	3
Tabla de revisión	3
Introducción	3
Airport	4
Capítulo de pruebas de rendimiento.....	6
Gráficos de rendimiento	6
Sin índices	6
Ejecutado en otro ordenador	7
Intervalo de confianza	8
Sin índices	8
Ejecutado en otro ordenador	8
Contraste de hipótesis	9
Conclusión	9
Bibliografía	9

Resumen ejecutivo

Este informe presenta las pruebas funcionales y de rendimiento realizadas sobre la entidad grupal **Airport**. Se evaluaron procesos como listado, creación, actualización, publicación y eliminación para reservas, registros, pasajeros y aeropuertos. Las pruebas cubren casos positivos, negativos y ataques de seguridad para garantizar la protección contra accesos no autorizados. Además, se realizaron análisis de rendimiento con y sin índices, y comparativas entre distintos equipos. Los resultados muestran que la aplicación funciona correctamente y de manera segura, identificando las operaciones más costosas en tiempo de procesamiento, especialmente las relacionadas con la gestión de reservas y aeropuertos. Este informe proporciona una visión integral que permite asegurar la eficiencia del sistema antes de su despliegue.

Tabla de revisión

Versión	Fecha	Descripción
1.0.0	26/5/2025	Versión inicial

Introducción

Este informe presenta las pruebas realizadas sobre una aplicación que gestiona reservas, que hemos elaborado durante todo el curso. Se validaron las funciones principales como crear, listar, actualizar y borrar, junto con pruebas de seguridad para evitar accesos no autorizados de distintas entidades y ahora en este informe se llevarán a cabo las de airport. Además, se analizaron los tiempos de respuesta con y sin índices, evaluando el rendimiento en distintos equipos. El documento describe la metodología, resultados y conclusiones para asegurar la calidad y eficiencia del sistema antes de su despliegue.

Capítulo sobre pruebas funcionales

Para cada prueba, antes de todo me he logueado como administrador, cuyo usuario es “administrator” y contraseña “administrator”. Estos casos de prueba se han llevado a cabo siguiendo el documento “Sample-Data.xlsx” y se han realizado tanto casos de pruebas positivos y negativos, como de hacking. En administrator podemos observar que tenemos acceso a listar cuentas de usuario, listar aerolíneas, listar aeropuertos, listar aeronaves, listar quejas y listar reservas. A continuación explicaremos paso a paso cómo se han llevado sus respectivas pruebas.

Primero de todo, tenemos que diferenciar entre el testing.safe y el testing.hack. En el testing.safe lo que se ha llevado a cabo los casos de prueba positivos y negativos, y en el testing.hack se han llevado pruebas de hacking.

Airport

Listar

Para llevar a cabo el **list.safe** lo que hemos hecho es listar los airports desde administrator, accediendo a este listado mediante el botón List Airports.

Para el **list.hack** me he metido con otro usuario (ej:manager1) y he intentado acceder a la URL del listado de los airports del administrator.

Mostrar

Para el **show.safe** lo que hemos hecho es listar los airports, accediendo a este listado mediante el botón List Airports, y pinchar en uno de los airports de esa lista.

Para el **show.hack** he probado a cambiar la URL del airport para que en vez de aparecerme ese airport me apareciera uno que no esté creado y también me he metido con otro usuario y he intentado acceder a la URL del show de ese airport. En ambos casos me ha salido error 500.

Crear

Para el **create.safe** hemos entrado en el formulario de crear un airport mediante el botón de create que aparece en la lista de airports. Primero de todo, hacíamos una prueba dejando el formulario en blanco, todos los atributos sin valor.

Después de esto, se prueba cada atributo individualmente con los valores positivos y negativos del Excel “Sample-Data”, dándole a “create” tras probar cada atributo y esperando los correspondientes mensajes. Por último, se crea unbooking con todos los valores positivos.

Para el **create.hack** con la herramienta para desarrolladores de Firefox se prueba cambiando la propiedad “value” del atributo “operationalScope” por un valor que no existiera, esperando un error “Invalid value”. Aquí las demás propiedades las debo tener con casos de prueba positivos. También me he metido con otro usuario y he intentado acceder a la URL del create, saliendo un error 500.

Actualizar

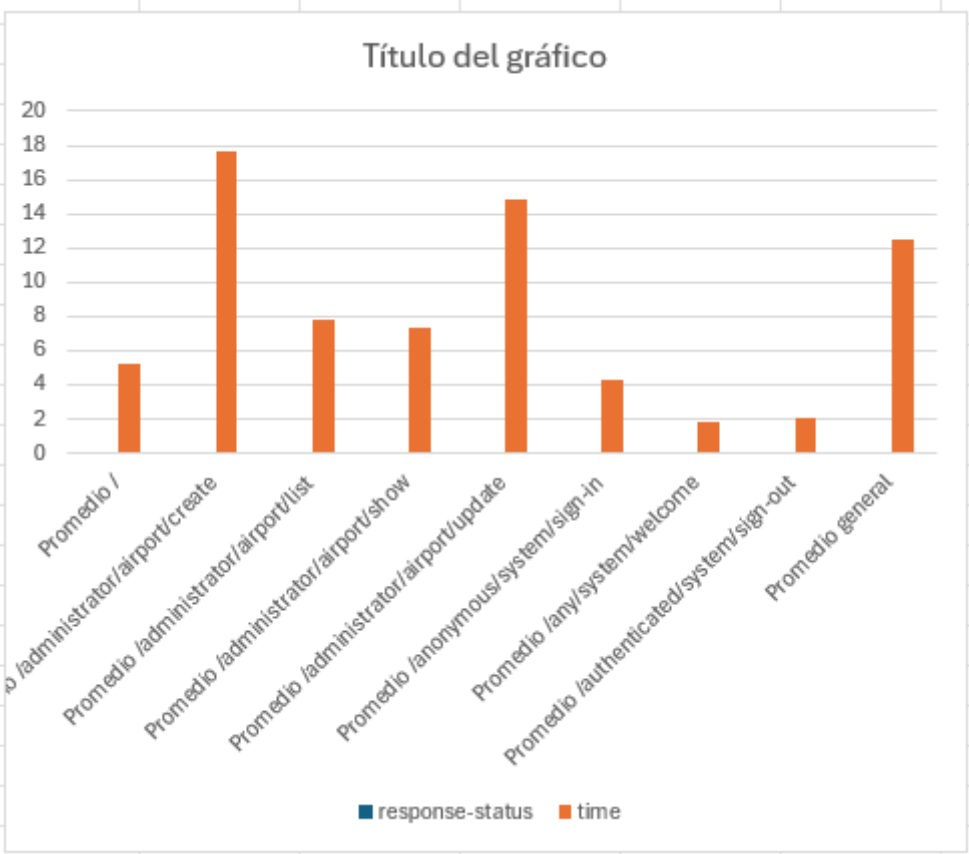
Para el **update.safe** nos metemos en un airport ya creado y hacemos lo mismo que hemos hecho en el create, primero dejamos todo el formulario en blanco y después se prueba cada atributo individualmente con los valores positivos y negativos del Excel “Sample-Data”. Por último lo actualizamos con nuevos valores positivos.

Para el **update.hack** con la herramienta para desarrolladores de Firefox se prueba cambiando la propiedad “value” del atributo “operationalScope” por un valor que no existiera, esperando un error “Invalid value”. Aquí las demás propiedades las debo tener con casos de prueba positivos. También me he metido con otro usuario y he intentado acceder a la URL del update, saliendo un error 500.

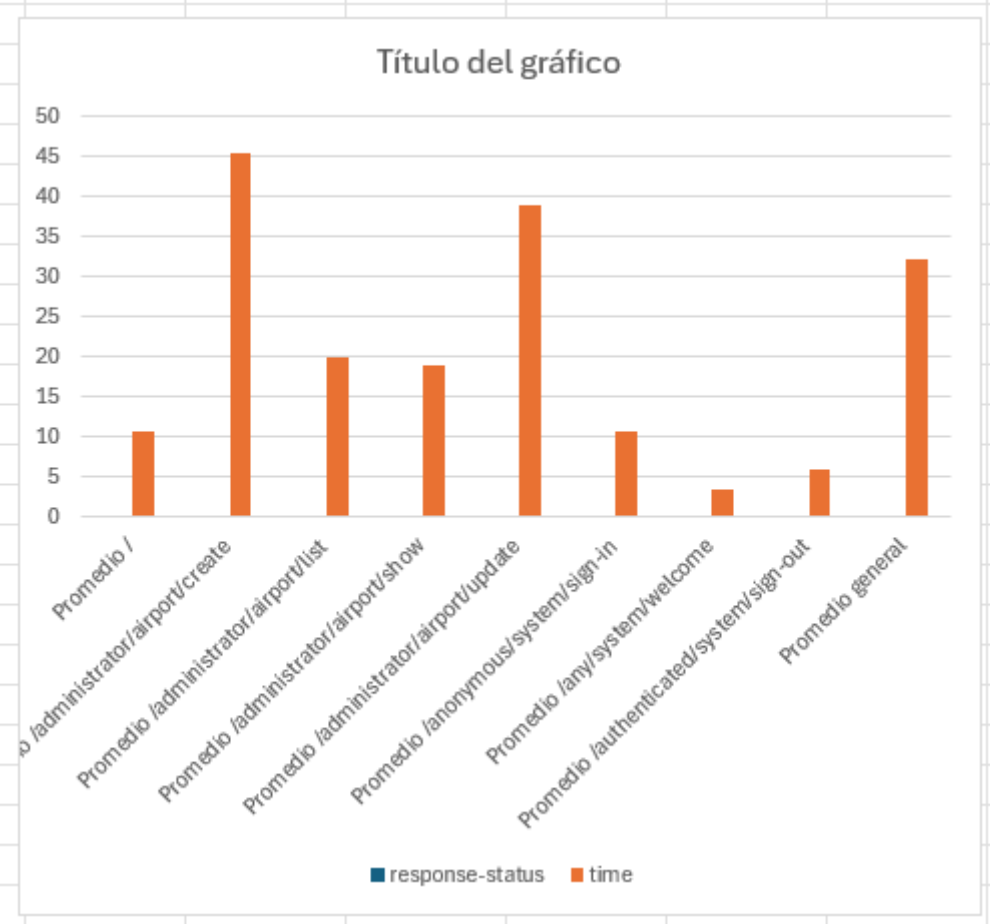
Capítulo de pruebas de rendimiento

Gráficos de rendimiento

Sin índices



Ejecutado en otro ordenador



En ambos se cumple lo mismo, las peticiones más ineficientes son las de crear y la de actualizar, tal y como se puede ver en el dibujo de la gráfica.

Intervalo de confianza

Sin índices

	A	B	C	D	E	F
1	Columna1					
2				Interval(ms)	11,3833186	13,6134067
3	Media	12,4983626		Interval(s)	0,01138332	0,01361341
4	Error típico	0,56610017				
5	Mediana	14,27845				
6	Moda	#N/D				
7	Desviación e	8,87893428				
8	Varianza de	78,835474				
9	Curtosis	16,8134454				
10	Coeficiente d	2,56817325				
11	Rango	78,611198				
12	Mínimo	0,973801				
13	Máximo	79,584999				
14	Suma	3074,59721				
15	Cuenta	246				
16	Nivel de confi	1,11504407				

Estos son los resultados estadísticos obtenidos en el análisis. El intervalo de confianza se sitúa entre 11,38 y 13,61 ms.

Ejecutado en otro ordenador

	A	B	C	D	E	F
1	Columna1					
2				Interva(ms)	29,1819735	35,0458371
3	Media	32,1139053		Interval(s)	0,02918197	0,03504584
4	Error típico	1,48852152				
5	Mediana	35,16625				
6	Moda	#N/D				
7	Desviación e	23,3465478				
8	Varianza de	545,061293				
9	Curtosis	8,38369563				
10	Coeficiente d	1,76612471				
11	Rango	164,0286				
12	Mínimo	1,8026				
13	Máximo	165,8312				
14	Suma	7900,0207				
15	Cuenta	246				
16	Nivel de confi	2,93193179				

Estos son los resultados estadísticos obtenidos en el análisis. El intervalo de confianza se sitúa entre 29,18 y 35,06 ms. Con esto se intuye que ese otro ordenador dispone de peor hardware que el primero.

Contraste de hipótesis

	A	B	C
1	Prueba z para medias de dos muestras		
2			
3		62,167899	133,4615
4	Media	12,2956298	31,7002416
5	Varianza (co	78,835474	545,061293
6	Observacion	245	245
7	Diferencia hi	0	
8	z	-12,1599457	
9	P(Z<=z) una c	0	
10	Valor crítico	1,64485363	
11	Valor crítico	0	
12	Valor crítico	1,95996398	

El valor p se encuentra en el intervalo [0.00, 0.05), por lo que se puede afirmar que las medias obtenidas son estadísticamente comparables. Dado que la media en este ordenador es de 12,30 ms y en el otro es de 31,7 ms, se concluye que el rendimiento del primer ordenador es superior, aproximadamente un 61% mejor.

Conclusión

Las pruebas realizadas sobre la entidad Airport han demostrado que sus funciones principales —listado, creación, actualización y eliminación— funcionan correctamente y de manera segura. Se ha verificado que el sistema protege adecuadamente el acceso a los datos, evitando modificaciones no autorizadas. Aunque el rendimiento es aceptable, se identificaron algunas operaciones que podrían beneficiarse de optimizaciones para mejorar la eficiencia. En conjunto, la entidad Airport muestra estabilidad y fiabilidad, asegurando su correcto funcionamiento dentro de la aplicación antes de su despliegue.

Bibliografía

Intencionadamente en blanco