

26 Mayo de 2025



TESTING REPORT STUDENT 2

REPOSITORIO: <https://github.com/AlbertoValenzuelaMunoz1/DP2-C1.002>

GRUPO: C1.002

Alberto Valenzuela Muñoz (albvalmunn@alum.us.es)

## Índice

1 Resumen ejecutivo .....	3
2 Introducción .....	3
3 Pruebas realizadas .....	4
4 Rendimiento obtenido .....	6
5 Comparativa rendimiento tras creación índices .....	8
6 Comparativa entre distintos ordenadores .....	9
7 Conclusión .....	9
8 Bibliografía .....	9

## 1. Resumen ejecutivo

Este informe presenta los resultados de las pruebas funcionales y de rendimiento realizadas sobre el sistema desarrollado. En la primera parte, se detalla el procedimiento realizado para probar la aplicación, de forma general para todas las entidades y a continuación se han detallados los detalles concretos para probar algunas entidades. Las pruebas han permitido identificar y corregir diversos fallos críticos, especialmente en áreas como autenticación y validación de datos de entrada.

En la segunda parte, se analizó el rendimiento del sistema ejecutando las pruebas funcionales en dos equipos con diferente capacidad de procesamiento. Se recopilaron los tiempos de respuesta y se generaron intervalos de confianza del 95% para cada conjunto de datos. Asimismo, se realizó una prueba de hipótesis estadística con un 95% de confianza para realizar una comparativa entre el rendimiento obtenido antes y después de la creación de los índices.

### Tabla de versiones

Versión	Fecha	Descripción
1.0.0	23/5/2025	Creación documento e informe de pruebas realizadas
1.1.0	24/5/2025	Informe de rendimiento y comparativa tras la creación de índices
1.2.0	26/5/2025	Comparativa de rendimiento con otro miembro del grupo y finalización del documento

## 2. Introducción

Este documento recoge de manera detallada los resultados obtenidos durante el proceso de pruebas del sistema desarrollado, tanto a nivel funcional como de rendimiento. El objetivo principal de este informe es evaluar la calidad del software mediante la verificación del cumplimiento de sus funcionalidades esperadas y el análisis de su en diferentes entornos de ejecución.

Las pruebas funcionales se centraron en verificar que las características clave del sistema operan correctamente en distintos escenarios, incluyendo tanto casos de uso comunes como situaciones de error. Para ello, se diseñaron casos de prueba específicos para cada funcionalidad del sistema, los cuales permitieron detectar errores, validar la lógica de negocio y comprobar la robustez de las validaciones implementadas.

Por otro lado, las pruebas de rendimiento se llevaron a cabo para medir el tiempo de respuesta del sistema durante la ejecución de las pruebas funcionales, utilizando dos ordenadores con distintas capacidades de hardware y en el mismo ordenador antes y después de la creación de índices. Esta evaluación permitió generar intervalos de confianza del 95% y realizar un contraste de hipótesis que permitió determinar en cuál de los equipos el sistema se desempeña mejor.

Este documento se estructura en dos capítulos principales. En el Capítulo 1, se presenta el conjunto de pruebas funcionales, organizadas por característica. En el Capítulo 2, se expone el análisis de las pruebas de rendimiento, incluyendo gráficos representativos, los intervalos de confianza calculados y los resultados del contraste estadístico.

### **3. Pruebas realizadas**

#### **3.1. Procedimiento común realizado en las pruebas**

El objetivo de este apartado es detallar el procedimiento que se ha realizado de forma común en todas las entidades. Cabe destacar que además de este procedimiento, en algunas entidades más complejas sí se han llevado a cabo otras pruebas adicionales que se explicarán más adelante.

##### **3.1.1 Procedimiento común para pruebas de edición/creación**

Para las pruebas de creación/edición/publicado de entidades se ha llevado a cabo el procedimiento que se ha explicado en la asignatura. Dicho procedimiento consiste en enviar primero el formulario en blanco y luego ir probando variaciones de los distintos atributos dejando siempre todos en blanco menos el que se está probando.

Finalmente, una vez probado suficientes variaciones de los atributos utilizando como ayuda los valores proporcionados en la hoja de Excel Sample Data se ha enviado el formulario con valores válidos de cada uno de los atributos y se ha comprobado que se realiza correctamente el proceso sin producirse ningún error.

##### **3.1.2 Procedimiento común para pruebas de hacking en edición/publicado/borrado**

Para las pruebas de edición y publicado se ha llevado a cabo la siguiente prueba en todas las entidades. Al entrar en un formulario si se inspecciona el HTML haciendo uso de las herramientas de desarrollador del navegador se observa que aparece un input oculto que genera el framework de forma automática. Este id es el que se utiliza a la hora de editar, publicar o borrar la entidad y si no se valida que se corresponda con una entidad que le corresponda al usuario y este publicada, podría hacer modificar o eliminar entidades que no le corresponde. Por tanto, la prueba realizada consiste en modificar ese id poniendo uno de una entidad que no le corresponda a ese usuario o que este publicada y verificar que se obtiene una pantalla de pánico. La realización de estas pruebas ha sido de gran utilidad porque se ha detectado casos de hacking de los que no estaba protegida la aplicación durante la realización de las mismas.

##### **3.1.3. Procedimiento común para pruebas de hacking en creación**

La prueba realizada en este caso es muy similar a la del apartado anterior y consiste en el mismo problema que ocurría con el input oculto con el id. Se ha observado que este input también está presente en el formulario de creación y que además se hacía bind de este atributo de forma automática. Esto habría las puertas a casos de hacking contra la aplicación ya que bastaba con editarlo y poner un id de otra entidad que ya

existía y se podía editar una entidad que ya existe desde el formulario de creación, incluso aunque esta este publicada o no pertenezca al usuario. La prueba realizada consiste en editar dicho id y verificar que si era distinto de 0 (que es el valor que debe tener el id para que al usar el método save del repositorio no se cree una entidad nueva) debía aparecer una pantalla de pánico. La realización de esta prueba ha sido de gran utilidad, ya que es un caso que no se había planteado y que habría las puertas a casos de hacking.

#### **3.1.4 Procedimiento común para pruebas de listado**

Para las pruebas de listado la prueba realizada consiste en acceder al listado de la entidad y comprobar que aparece la información correcta.

#### **3.1.5 Procedimiento común para pruebas de tipo show**

Para las pruebas de tipo show el procedimiento ha consistido en acceder a la pantalla de show de una entidad y comprobar que aparece toda la información de forma correcta.

Además, como prueba de hacking se ha probado a modificar la URL para colocar el id de una entidad que no pertenece al usuario y se ha verificado que se obtiene una pantalla de pánico en estos casos.

#### **3.1.6 Procedimiento común para pruebas de tipo borrado**

Para las pruebas de borrado el procedimiento realizado ha consistido en acceder a una entidad posible de borrar y eliminarla y comprobar que el proceso finaliza correctamente.

Además, como prueba de hacking se ha probado a eliminar una entidad que no pertenece al usuario o que está publicada modificando el input oculto del id mencionado anteriormente y se ha comprobado que en estos casos se ha obtenido una pantalla de pánico.

### **3.2 Pruebas realizadas entidad Passenger**

En esta entidad no se han realizado pruebas adicionales a las descritas anteriormente, ya que se trata de la entidad más simple y con estas pruebas se considera que se han probado los casos necesarios. Además, la cobertura obtenida con dichas pruebas ha sido de un 100%.

### **3.3 Pruebas realizadas entidad Booking**

En este caso si ha sido necesario realizar pruebas adicionales, ya que se trata de la entidad más compleja. Aún con todo eso no se ha alcanzado una cobertura del 100% en todos los servicios (aunque prácticamente del 100%). Esto se ha debido a instrucciones booleanas complejas compuestas por muchas operaciones and y que se utilizan principalmente para evitar NullPointerException comprobando que las variables utilizadas no tengan valor nulo al principio debido a la evaluación cortocircuitada de and.

#### **3.3.1 Pruebas extras creación/edición/publicado**

En este caso ha sido necesario realizar varias pruebas extra:

- Uso de valores no únicos para locatorCode: esta propiedad esta anotada con @Column(unique=true) y debido a que esta anotación no es propia del framework producía que se llegara a una pantalla de pánico al introducir un valor repetido en el formulario, por lo que hubo que incluir la restricción en el método validate del servicio. La prueba realizada consistió en que no se

obtuviera la pantalla de pánico, sino un mensaje indicando el problema de que ya estaba usado ese valor en el mismo formulario.

- Modificación del desplegable de vuelos: en el formulario aparecía un desplegable donde podíamos seleccionar el vuelo, este vuelo debía ocurrir en el pasado y estar publicado. La prueba realizada consistió en modificar los valores cambiando el HTML haciendo uso de las herramientas del navegador y comprobar que se obtuviera una pantalla de pánico al introducir un vuelo del futuro o que no estuviera publicado.

### **3.4 Pruebas realizadas entidad BookingRecord**

El proceso de prueba de esta entidad ha sido algo más especial, debido a que se trata de una entidad creada para evitar el uso de una relación ManyToMany entre Booking y Passenger. En cuanto a la cobertura alcanzada, en este caso sí que ha sido de un 100% en todos los servicios. A continuación se detallan las pruebas extra realizadas en este caso.

#### **3.4.1 Pruebas realizadas listado**

En el caso del listado ha habido que realizar una prueba adicional de hacking que consiste en entrar al listado de pasajeros de una booking creada por otro usuario. En estos casos se ha verificado que se obtiene una pantalla de pánico.

#### **3.4.2 Pruebas realizadas creación**

El caso de creación ha sido más simple, porque solo se dispone de un desplegable con los pasajeros del usuario, por lo que las únicas pruebas realizadas son enviar el formulario en blanco y enviarlo probando varios pasajeros.

Además, como prueba de hacking se han modificado las opciones del desplegable haciendo uso de las herramientas del navegador y se ha comprobado que se obtiene una pantalla de pánico al seleccionar un pasajero de otro usuario o que ya había sido añadido previamente a la booking.

### **4. Rendimiento obtenido**

El objetivo de este apartado es realizar un análisis del rendimiento obtenido en las pruebas descritas anteriormente.

#### **4.1. Gráficos rendimiento**

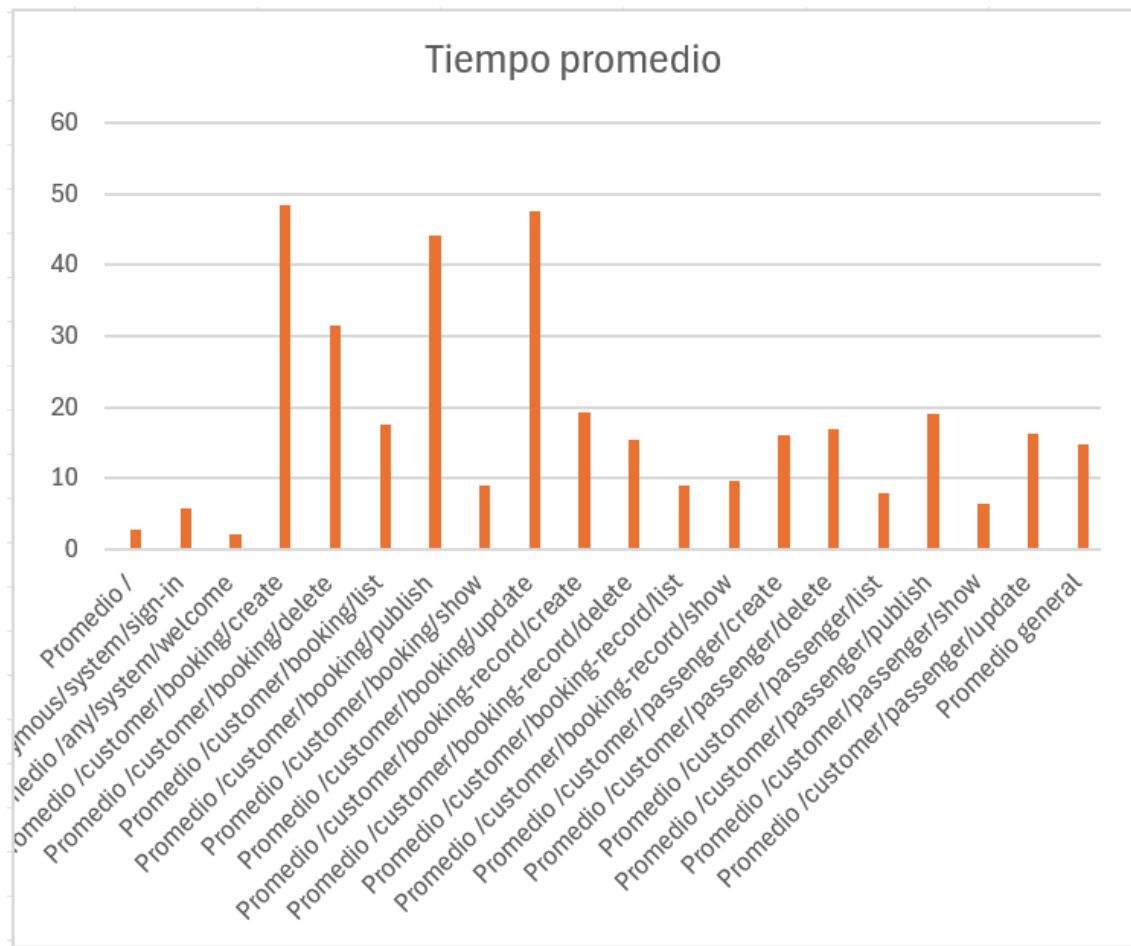


Figura 1: Promedio de tiempo de cada tipo de petición

Como se puede observar las peticiones más ineficientes son las relacionadas con booking, esto se debe a que se realizan varias consultas a la base de datos a la hora de buscar los datos necesarios para el desplegable de los vuelos y de realizar las validaciones necesarias.

Por otra parte, el promedio general obtenido ha sido de 14,82 milisegundos, lo cual es una cifra que es bastante buena.

## 4.2. Datos estadísticos

Media	14,826747	
Error típico	0,7285085	
Mediana	8,9818	
Moda	1,3724	
Desviación estándar	17,1625222	
Varianza de la muestra	294,552169	
Curtosis	5,47604549	
Coefficiente de asimetría	2,18845686	
Rango	120,9853	
Mínimo	1,0138	
Máximo	121,9991	
Suma	8228,8446	
Cuenta	555	
Nivel de confianza(95,0%)	1,43097666	
Intervalo confianza	13,3957704	16,2577237

Figura 2: datos estadísticos obtenidos

Estos son los resultados estadísticos obtenidos en el análisis. El intervalo de confianza se sitúa entre 13,4 y 16,26 ms, que se puede considerar como válido, ya que en el proyecto no se impone ningún requisito relacionado con el rendimiento.

## 5. Comparativa rendimiento tras creación índices

En este apartado se realizará una comparativa de rendimiento tras la creación de los índices para optimizar las consultas de la base de datos, utilizando la herramienta del z-analysis de Excel.



Figura 3: Promedio de tiempos obtenido tras creación de índices

Media	15,7925782	
Error típico	0,80182805	
Mediana	9,5455	
Moda	2,1965	
Desviación estándar	18,8898165	
Varianza de la muestra	356,825166	
Curtosis	6,03918101	
Coefficiente de asimetría	2,26467251	
Rango	133,5172	
Mínimo	1,0557	
Máximo	134,5729	
Suma	8764,8809	
Cuenta	555	
Nivel de confianza(95,0%)	1,57499498	
Intervalo de confianza	14,2175832	17,3675732

Figura 4: Datos estadísticos obtenidos tras creación de índices

En cuanto al valor de p-value obtenido es de 0,37, lo cual significa que no hay una diferencia significativa de rendimiento, esto es de esperar debido a que el volumen de datos que se utiliza en las pruebas es muy pequeño, por lo que es lógico que no se obtenga a penas diferencia en el rendimiento.



## 6. Comparativa entre distintos ordenadores

A continuación, se realizará una comparativa entre el rendimiento obtenido en mi ordenador y el ordenador de otro integrante del grupo (después de la creación de índices).



Figura 5: Promedio de tiempos obtenidos en otro ordenador

Media	10,3408796
Error típico	0,45511506
Mediana	7,2198
Moda	1,2421
Desviación estándar	10,7217999
Varianza de la muestra	114,956993
Curtosis	13,1789386
Coefficiente de asimetría	2,64203084
Rango	101,6294
Mínimo	0,8073
Máximo	102,4367
Suma	5739,1882
Cuenta	555
Nivel de confianza(95,0%)	0,89396215
Intervalo confianza	9,44691749 11,2348418

Figura 6: Datos estadísticos obtenidos en otro ordenador

Como el p-value obtenido es de  $2.5 \times 10^{-9}$ , que es un valor muy cercano a 0, por tanto se pueden comparar las medias obtenidas. Como en mi ordenador la media obtenida es de 15,79 ms y en el otro es de 10,34 ms, se llega a la conclusión de que el rendimiento en mi ordenador es bastante peor (alrededor de un 33%).

## 7. Conclusión

La elaboración del conjunto completo de pruebas ha resultado de gran utilidad para identificar errores residuales en la aplicación, así como para detectar posibles vectores de ataque que no se habían considerado inicialmente. Este proceso ha permitido reforzar tanto la robustez funcional como la seguridad del sistema.

En lo que respecta a la comparativa de rendimiento, las diferencias observadas tras la incorporación de índices fueron mínimas, lo cual era previsible dado el reducido volumen de datos utilizado durante las pruebas. No obstante, al comparar los resultados con los de otro integrante del grupo, se constató que el rendimiento en mi equipo fue significativamente inferior, lo que evidencia las limitaciones del hardware empleado en mi caso.

## 8. Bibliografía

Intencionalmente en blanco.