



# Testing Report

**VIRGINIA MESA PEREZ**

Repository: <https://github.com/DP2-C1-021/Acme-ANS>  
Student 2



## ÍNDICE

Resumen ejecutivo .....	2
Introducción .....	3
Pruebas funcionales...	4
Pruebas de rendimiento .....	5
Conclusion y firma .....	10



## **Resumen Ejecutivo**

Este documento recoge el análisis y evaluación del rendimiento de un sistema tras un proceso de pruebas y optimización. A lo largo del informe se describen los pasos seguidos para diseñar los tests, analizar los resultados, aplicar mejoras y validar su impacto. El trabajo se apoya en técnicas estadísticas para garantizar la objetividad de las conclusiones. Todo el proceso ha sido documentado de forma estructurada con el objetivo de demostrar si se cumple el requisito de rendimiento establecido.



## Introducción

El rendimiento de una aplicación web es un factor crítico para garantizar una experiencia de usuario fluida y eficiente. En entornos donde se gestionan operaciones de base de datos intensivas, como es el caso de sistemas de reservas, es esencial evaluar continuamente los tiempos de respuesta y aplicar técnicas de optimización cuando sea necesario.

El presente documento recoge el análisis realizado sobre el comportamiento del sistema antes y después de aplicar índices en entidades clave del dominio, con el fin de medir su impacto en los tiempos de respuesta. Para ello, se ha diseñado un conjunto de pruebas automatizadas que simulan el uso de las funcionalidades principales (listar, crear, mostrar y actualizar) bajo distintos escenarios de validación (safe) y ataques (hack).

Los datos recogidos han sido procesados y analizados estadísticamente mediante herramientas como Excel y técnicas de contraste de hipótesis (Z-Test), con el propósito de evaluar de manera objetiva si la optimización realizada conduce a mejoras significativas en términos de rendimiento. Este análisis no solo permite validar la eficacia de la optimización aplicada, sino que también proporciona una metodología reproducible para futuras evaluaciones de eficiencia.



## Pruebas funcionales

A continuación, se presenta el listado de casos de prueba implementados, organizados por funcionalidad. Cada funcionalidad se ha probado en escenarios válidos (.safe) y en escenarios maliciosos o incorrectos (.hack).

<b>Booking</b>	<b>create.safe</b>	Un customer crea una booking correctamente y sale en el listado de bookings de ese customer
	<b>list.safe</b>	Cada customer lista solo sus bookings asociadas
	<b>show.safe</b>	Cada customer puede ver solo la booking si le pertenece
	<b>publish.safe</b>	Un customer puede publicar una booking creada y no publicada ya.
	<b>update.safe</b>	Un customer puede modificar los datos de una booking no publicada.
	<b>create.hack</b>	Si no eres customer no puedes acceder a create y no puedes modificar datos desde F12
	<b>list.hack</b>	Si no eres customer no puedes listar tus bookings
	<b>show.hack</b>	Si no eres customer no puedes ver las bookings y no puedes ver una booking que no corresponda a tu customer
	<b>publish.hack</b>	No puedes publicar si no eres customer
	<b>update.hack</b>	No puedes modificar datos readonly desde f12 ni datos incorrectos, además debes ser customer.
<b>BookingRecord</b>	<b>create.safe</b>	Un customer puede crear una bookingrecord añadiendo pasajeros a la booking.
<b>Passenger</b>	<b>create.safe</b>	Un customer puede crear un pasajero correctamente.
	<b>Update-publish.safe</b>	Un customer puede actualizar los valores de un pasajero no publicado y puede publicarlo.
	<b>list.safe</b>	Un customer lista los pasajeros asociados a él.
	<b>show.safe</b>	Un customer puede ver los pasajeros asociados a él.
	<b>list.hack</b>	Un customer no puede listar pasajeros de otro y otro rol no puede listar ningún pasajero.
	<b>show.hack</b>	Un customer no puede ver el pasajero de otro customer y si no eres customer no puedes ver ninguno.
	<b>Publish-update-create.hack</b>	No puedes modificar datos desde F12 en el navegador ni readonly. Además, si no eres customer no puedes actualizar, ni crear ni publicar pasajeros.



## Pruebas de rendimiento

Para llevar a cabo las pruebas de rendimiento del sistema, se han llevado a cabo los siguientes pasos:

### 1. Elaborar archivos. safe y. hack

Para la elaboración de estos archivos, se han probado las funcionalidades de list, show, update, publish y create de bookings, bookingrecord y passengers. Para los archivos .safe, se han probado las funcionalidades con datos correctos y datos incorrectos.

Por otro lado, para los archivos .hack, se han realizado pruebas para comprobar que las funcionalidades list, show, update y créate de booking, bookingrecord y passengers , no puedan ser accesibles desde otro rol cualquiera además de que no se pueden editar a través de F12 ni los campos readonly ni los atributos de navegación.

### 2. Análisis de datos

Una vez obtenidos los archivos de tests, se ha creado un archivo de Excel con todos los datos de los tests (tester-performance-clean.xlsx) y se han realizado los siguientes análisis:

#### - Análisis previos a introducción de índices:

Los datos que hemos obtenido al realizar el gráfico de los tiempos promedios han sido los siguientes:





⌚ MIR (sin índices) = 19,0276235 ms

Esta gráfica nos muestra que el promedio general está muy por debajo de un segundo por lo que globalmente el sistema no tiene cuellos de botella graves, pero hay puntos mejorables.

Los endpoints con tiempos más altos serían:

- ☒ /customer/passenger/update
- ☒ /customer/booking-record/create
- ☒ /customer/booking/publish
- ☒ /customer/booking/create
- ☒ /customer/passenger/publish
- ☒ /customer/booking/update

Finalmente se ha calculado el intervalo de confianza usando Descriptive Statistics:

<i>estadísticas</i>	
Media	19,0276235
Error típico	1,13234748
Mediana	9,06675
Moda	#N/D
Desviación estándar	22,0153512
Varianza de la muestra	484,675688
Curtosis	11,636131
Coeficiente de asimetría	2,74226637
Rango	179,8067
Mínimo	2,9186
Máximo	182,7253
Suma	7192,4417
Cuenta	378
Nivel de confianza(95,0%)	2,22650811

El intervalo de confianza sería:

- ☒ Límite inferior:  $\max(0, 19,0276 - 2,2265) = \max(0, 16,8011) = 16,8011$
- ☒ Límite superior:  $19,0276 + 2,2265 = 21,2541$



## - Índices añadidos

### **Booking:**

```
@Table(indexes = {  
    @Index(columnList = "locatorCode", unique = true), @Index(columnList =  
    "flight_id"), @Index(columnList = "customer_id"), @Index(columnList =  
    "draftMode"),  
})
```

### **Passenger:**

```
@Table(indexes = {  
    @Index(columnList = "passportNumber"), @Index(columnList =  
    "customer_id"), @Index(columnList = "draftMode"),  
})
```

### **BookingRecord:**

```
@Table(indexes = {  
    @Index(columnList = "booking_id"), @Index(columnList =  
    "passenger_id"), @Index(columnList = "booking_id, passenger_id", unique =  
    true)  
})
```



### - Análisis posteriores a la introducción de índices

Una vez añadidos los índices volvemos a realizar el mismo análisis tras ejecutar nuestros tests de nuevo para poder comparar resultados:

El grafo obtenido de promedios es:



La estadística descriptiva obtenida es:

<i>estadísticas</i>	
Media	16,24755476
Error típico	0,971431018
Mediana	7,82545
Moda	#N/D
Desviación estándar	18,88677761
Varianza de la muestra	356,7103684
Curtosis	11,40890263
Coeficiente de asimetría	2,631752647
Rango	162,8458
Mínimo	2,444
Máximo	165,2898
Suma	6141,5757
Cuenta	378
Nivel de confianza(95,0%)	1,910101873



El intervalo de confianza seria:

- Limite inferior:  $\max(0,16,2476 - 1,9101) = \max(0,14,3375) = 14,3375$
- Limite superior:  $16,2476 + 1,9101 = 18,1577$

#### 4. Comparación y conclusiones

Realizamos un análisis de los tiempos antes y después de aplicar los índices comparando sus resultados.

Tras hacer el **Z-Test** hemos obtenido lo siguiente:

Prueba z para medias de dos muestras			
		Variable 1	Variable 2
Media		19,0276235	16,2475548
Varianza (conocida)		484,675688	356,710368
Observaciones		378	378
Diferencia hipotética de las medias		0	
z		1,86339011	
P(Z<=z) una cola		0,0312037	
Valor crítico de z (una cola)		1,64485363	
Valor crítico de z (dos colas)		0,0624074	
Valor crítico de z (dos colas)		1,95996398	

- El **estadístico z calculado** es 1,8634
- El **valor crítico de z (una cola)** es 1,6449
- El **p-valor** (una cola) es 0,0312

La media de los tiempos **sin índices** es **significativamente mayor** que la media de los tiempos **con índices**, ya que el p-valor es menor a 0,05.



## Conclusión Y Firma

A lo largo de este informe se ha llevado a cabo un análisis riguroso del rendimiento del sistema mediante la aplicación de técnicas de testing formal y análisis estadístico. Inicialmente, se identificaron las funcionalidades más ineficientes en términos de tiempo de respuesta, lo que permitió enfocar los esfuerzos de optimización en los puntos críticos. La incorporación de índices en las entidades correspondientes tuvo un impacto notable, evidenciado por la mejora significativa en los tiempos promedio y la validación estadística mediante una prueba Z.

Los resultados obtenidos permiten concluir que el objetivo de rendimiento se ha cumplido satisfactoriamente, manteniendo los tiempos de respuesta dentro de los márgenes esperados y muy por debajo del umbral de 1000 milisegundos. Esta mejora no solo refuerza la eficiencia del sistema, sino que también garantiza una mejor experiencia de usuario y una mayor escalabilidad en contextos de carga creciente. En definitiva, el proceso ha demostrado la importancia de combinar pruebas funcionales con análisis cuantitativo para lograr sistemas robustos y optimizados.

**Virginia Mesa Pérez**