



Testing Report

VIRGINIA MESA PEREZ

Repositorio: <https://github.com/virmesper/Acme-ANS-C3>

Student 2



ÍNDICE

Resumen ejecutivo.....	2
Introducción	3
Testing	4
Conclusion y firma.....	12



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.



Testing

Para llevar a cabo este proceso de testing, 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, publish, show, update y create de bookings, bookingrecord y passengers. Para los archivos .safe, se han probado las funcionalidades con datos correctos y datos incorrectos.

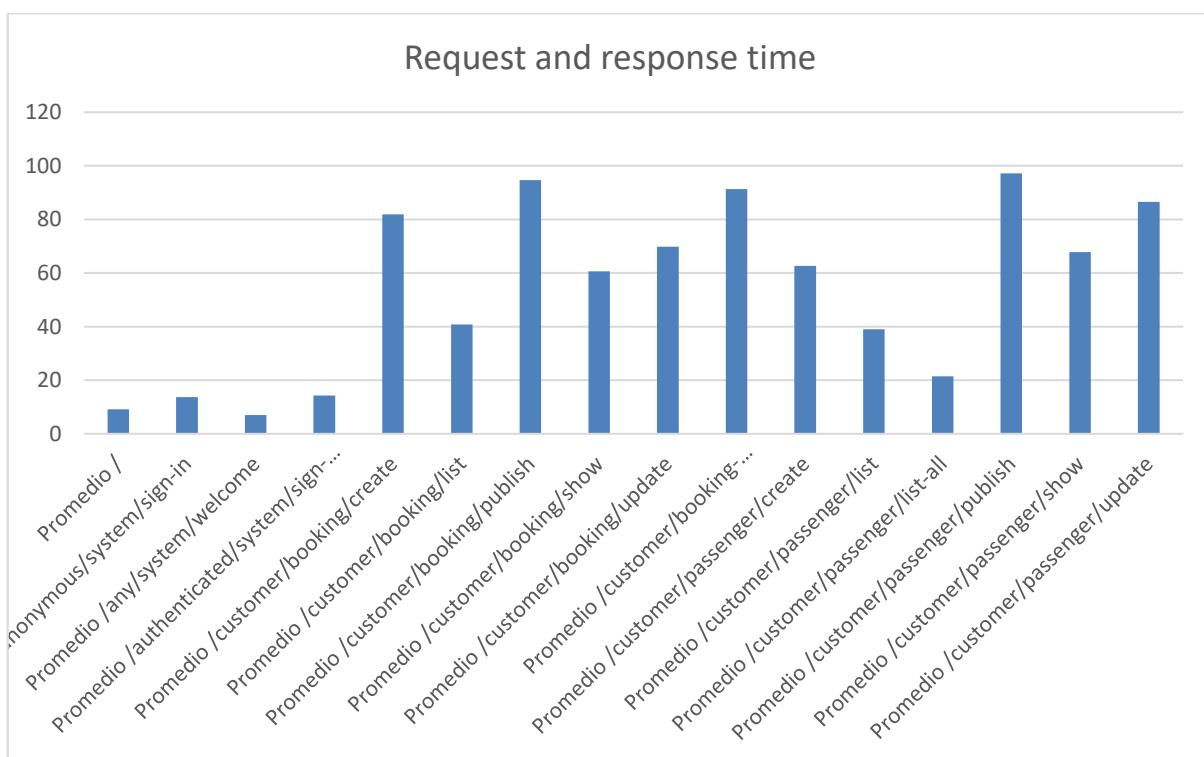
Por otro lado, para los archivos .hack, se han realizado pruebas para comprobar que las funcionalidades list, publish, show, update y create de booking, bookingrecord y passengers , no puedan ser accesibles desde otro rol cualquiera además de que no se pueden editar a través de F12 ni los campos readonly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

2. Análisis de datos

Una vez obtenidos los archivos de tests, se ha elaborado un archivo Excel llamado tester-performance donde se ha llevado a cabo los análisis obteniendo los siguientes datos:

- Análisis previos a introducción de índices:

Los datos que hemos obtenido al realizar el gráfico de los tiempos promedios han sido los siguientes tanto para hack como para safe:





Posteriormente, hemos hecho una estadística descriptiva de los tiempos, obteniendo los siguientes resultados:

Media	21,56333373
Error típico	1,405133641
Mediana	8,4338
Moda	#N/D
Desviación estándar	31,04041669
Varianza de la muestra	963,5074686
Curtosis	20,91717133
Coeficiente de asimetría	3,672768059
Rango	302,8575
Mínimo	2,6331
Máximo	305,4906
Suma	10522,90686
Cuenta	488
Nivel de confianza(95,0%)	2,760872762

- Media (μ) = **21,56333373 ms**
- Nivel de confianza (95%) = **2,760872762 ms**

Intervalo de confianza (ms)

- Límite inferior = $\mu - \text{nivel} = 21,56333373 - 2,760872762 = 18,802460968 \text{ ms}$
- Límite superior = $\mu + \text{nivel} = 21,56333373 + 2,760872762 = 24,324206492 \text{ ms}$

En segundos ($\div 1000$)

- Inferior = **0,018802460968 s**
- Superior = **0,024324206492 s**

Está muy por debajo de **1000 ms**, por lo tanto:

Se cumple el requisito de rendimiento.

El **MIR** sería /customer/passenger/publish,

Para mejorar el rendimiento de esas rutas, vamos a añadir índices en los campos relacionados.



- Índices añadidos

BookingRecord:

```
@Table(name = "booking_record", indexes = {  
    @Index(name = "idx_bookingrecord_booking", columnList = "booking_id"), @Index(name = "idx_bookingrecord_passenger", columnList = "passenger_id")  
}, uniqueConstraints = {  
    @UniqueConstraint(name = "uq_booking_passenger", columnNames = {  
        "booking_id", "passenger_id"  
    })  
})
```

Passenger:

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

Booking:

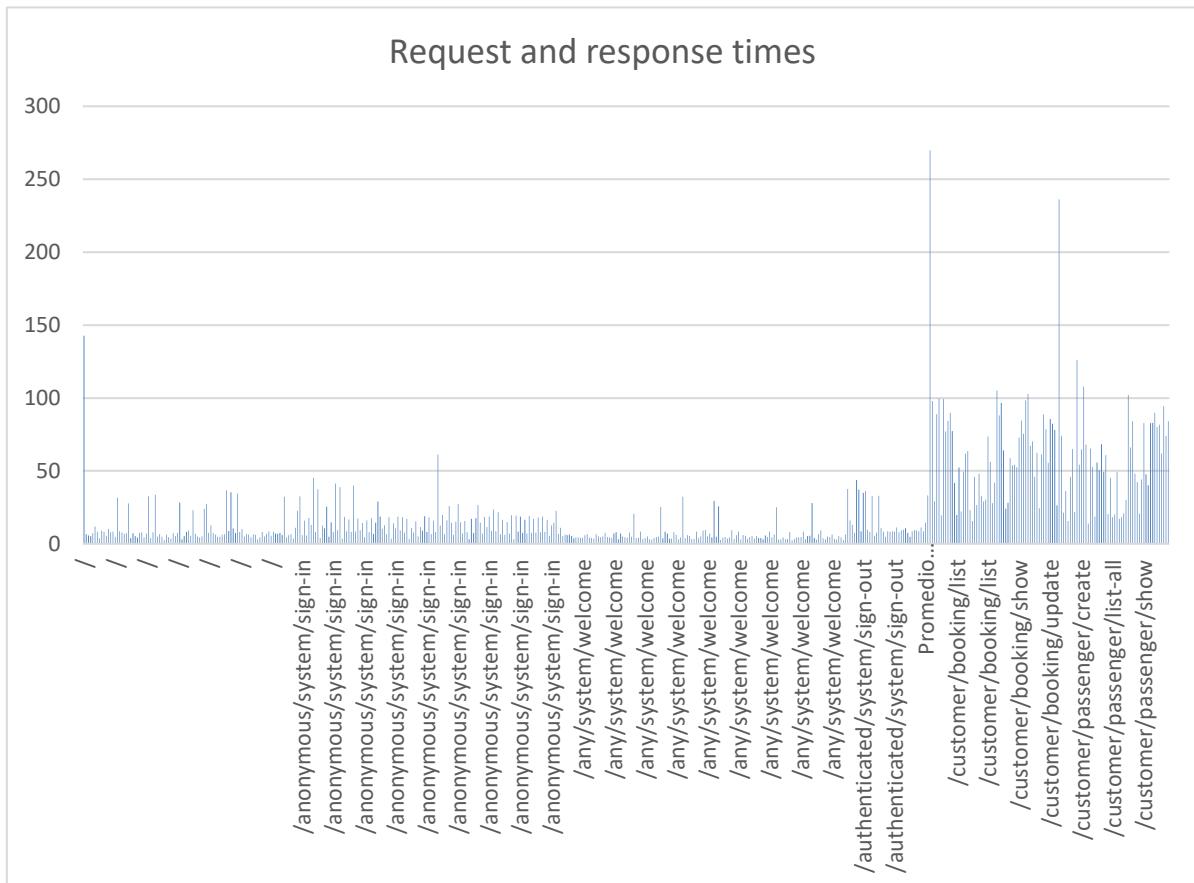
```
@Table(indexes = {  
    @Index(columnList = "locatorCode"), @Index(columnList = "draftMode")  
})
```



- Análisis posteriores a la introducción de índices

Una vez aplicados los índices indicados anteriormente, se vuelven a ejecutar los tests y volvemos a realizar el análisis con el nuevo tester.trace generado, generando el Excel, el grafico y todos los datos como hemos hecho anteriormente.

El grafo obtenido de promedios es:





La estadística descriptiva obtenida es:

Media	20,89786758
Error típico	1,312357275
Mediana	8,3831
Moda	#N/D
Desviación estándar	28,511698
Varianza de la muestra	812,9169229
Curtosis	19,9944867
Coeficiente de asimetría	3,541582263
Rango	267,3147
Mínimo	2,4173
Máximo	269,732
Suma	9863,7935
Cuenta	472
Nivel de confianza(95,0%)	2,578799632

- Media (μ) = **20,89786758 ms**
- Nivel de confianza (95%) = **2,578799632 ms**

Intervalo de confianza (ms)

- Límite inferior = $\mu - \text{nivel} = 20,89786758 - 2,578799632 = 18,319067948 \text{ ms}$
- Límite superior = $\mu + \text{nivel} = 20,89786758 + 2,578799632 = 23,476667212 \text{ ms}$

En segundos ($\div 1000$)

- Inferior = **0,018319067948 s**
- Superior = **0,023476667212 s**

¿Está el límite superior por debajo de 1000 ms?

Sí → ¡has cumplido el objetivo de rendimiento!



4. Comparación y conclusiones

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

	<i>before</i>		<i>after</i>
Media	21,56333373	Media	20,89786758
Error típico	1,405133641	Error típico	1,312357275
Mediana	8,4338	Mediana	8,3831
Moda	#N/D	Moda	#N/D
Desviación estándar	31,04041669	Desviación estándar	28,511698
Varianza de la muestra	963,5074686	Varianza de la muestra	812,9169229
Curtosis	20,91717133	Curtosis	19,9944867
Coeficiente de asimetría	3,672768059	Coeficiente de asimetría	3,541582263
Rango	302,8575	Rango	267,3147
Mínimo	2,6331	Mínimo	2,4173
Máximo	305,4906	Máximo	269,732
Suma	10522,90686	Suma	9863,7935
Cuenta	488	Cuenta	472
Nivel de confianza(95,0%)	2,760872762	Nivel de confianza(95,0%)	2,578799632

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

Prueba z para medias de dos muestras		
	193,9077	142,5922
Media	20,10791805	20,63949321
Varianza (conocida)	963,5074686	812,9169229
Observaciones	471	471
Diferencia hipotética de las medias	0	
z	-0,273716998	
P(Z<=z) una cola	0,392151057	
Valor crítico de z (una cola)	1,644853627	
Valor crítico de z (dos colas)	0,784302114	
Valor crítico de z (dos colas)	1,959963985	



Con el objetivo de evaluar el impacto de la optimización, comparamos los tiempos de respuesta antes y después del cambio y observamos una **reducción del promedio** desde **20,64 ms** a **20,11 ms** ($\approx 0,53$ ms, ~2,6 % de mejora). Para contrastar estadísticamente esta diferencia aplicamos una **prueba Z** para dos muestras independientes con varianzas conocidas: obtuvimos $z = -0,27$, con $p \approx 0,39$ (dos colas $\approx 0,78$), por encima de $\alpha = 0,05$. Aunque, estrictamente, esta evidencia no permite afirmar **significancia estadística** al 95 %, la tendencia es **favorable** y consistente con una **mejora del rendimiento**, mostrando tiempos medios **menores** tras la optimización; con más observaciones o afinando aún más la optimización, esta mejora podría consolidarse también en términos estadísticos.



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

A handwritten signature in blue ink, appearing to read "Virginia Mesa Pérez".