Diseño y pruebas 2

Testing Report

Grupo C2.057

**Realizado por:**

*Moreno Guerrero, José Manuel (*[*josmorgue3@alum.us.es*](mailto:josmorgue3@alum.us.es)*)*

Fecha: 02/07/2025

### **Tabla de contenidos**

[**Tabla de contenidos 2**](#_heading=h.tkvs4ggqslwd)

[**Capítulo 1: Functional Testing 3**](#_heading=h.851b9rui3uz3)

[Funcionalidad: Booking 3](#_heading=h.9jrw2by46dte)

[Funcionalidad: Passenger 3](#_heading=h.q3y3dwj791h2)

[Funcionalidad: Booking Record 4](#_heading=h.perdh218nt8r)

[**Capítulo2: Performance Testing 6**](#_heading=h.o9vnwtjo0hi9)

[Ordenadores utilizados 6](#_heading=h.1fiheqysawd)

[Gráficas de tiempos de respuesta 6](#_heading=h.s5m19hlhlttr)

[Promedio general 7](#_heading=h.hjl4pn91xpnj)

[Conclusión 7](#_heading=h.3inyu5oy4s2e)

[Estadísticas 7](#_heading=h.mzqjqxohprr6)

[Intervalo de confianza 8](#_heading=h.y2kvowpcv1es)

[Hipótesis contrastada 8](#_heading=h.resgebny56e7)

[Conclusión 10](#_heading=h.d0i41a4z41j2)

### 

### 

# Capítulo 1: Functional Testing

A continuación se detallan los casos de prueba realizados, agrupados por funcionalidad. Para cada uno se incluye una breve descripción y su efectividad para detectar errores.

## Funcionalidad: Booking

| **Test** | **Descripción** | **Detecto errores** |
| --- | --- | --- |
| create-booking.safe | Comprobación de la creación de una reserva con datos válidos. | Sí |
| delete-booking.safe | Eliminación de una reserva en estado borrador. | No |
| list-booking.safe | Listado de las reservas del customer autenticado. | No |
| list-booking.hack | Intento de listar reservas sin estar autenticado o como otro rol. | Sí |
| publish-booking.safe | Publicación de una reserva correctamente completada. | Sí |
| show-booking.safe | Visualización de una reserva propia. | No |
| show-booking.hack | Visualización de una reserva de otro usuario (no autorizado). | Sí |
| update-booking.safe | Edición de los datos de una reserva en estado borrador. | Sí |

## Funcionalidad: Passenger

| **Test** | **Descripción** | **Detecto errores** |
| --- | --- | --- |
| list-passenger.safe | Listado de pasajeros del usuario logado. | No |
| list-passenger.hack | Acceso a la lista de pasajeros por parte de un usuario no autorizado. | Sí |
| list-passenger-booking.safe | Listado de pasajeros dentro de un booking del usuario logado. | No |
| list-passenger-booking.hack | Lista pasajeros de bookings de otro usuario. | Sí |
| create-passenger.safe | Creación de un pasajero con datos válidos e inválidos. | Sí |
| create-passenger.hack | Acceso a la creación de pasajero por parte de usuario no autorizado. | Sí |
| show-passenger.safe | Visualización de un pasajero propio. | No |
| show-passenger.hack | Accede a un pasajero ajeno. | Sí |
| update-passenger.safe | Modifica un pasajero (con datos válidos e inválidos) | Sí |
| publish-passenger.safe | Publica un pasajero (revisa que no esté en modo draft) | Sí |
| delete-passenger.safe | Elimina un pasajero. | No |

## Funcionalidad: Booking Record

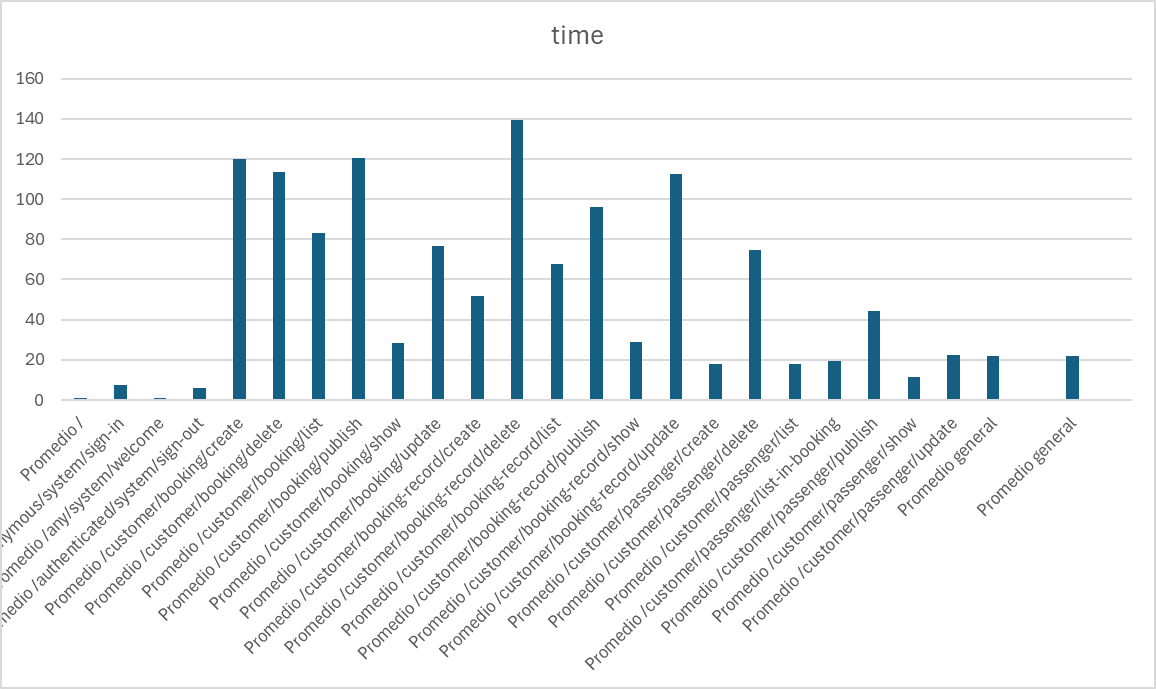
| **Test** | **Descripción** | **Detecto errores** |
| --- | --- | --- |
| create-bookingRecord.safe | Crea un registro de reserva válido e inválido. | Sí |
| create-bookingRecord.hack | Accede a la creación con manipulación por F12 o sin autorización. | Sí |
| update-bookingRecord.safe | Actualiza un registro de reserva con datos válidos e inválidos. | Sí |
| update-bookingRecord.hack | Intenta modificar registro ajeno. | Sí |
| delete-bookingRecord.safe | Elimina un registro de reserva propio. | No |
| publish-bookingRecord.safe | Publica un registro de reserva (validaciones de estado) | Sí |
| publish-bookingRecord.hack | Publicación con manipulación del ID ajeno. | Sí |
| show-bookingRecord.safe | Muestra un registro propio. | No |
| show-bookingRecord.hack | Accede al show de registro ajeno | Sí |
| list-bookingRecord.safe | Accede a lista de registros propios. | No |
| list-bookingRecord.hack | Accede a lista de registros ajenos o sin autorización. | Sí |

# Capítulo2: Performance Testing

## Ordenadores utilizados

Los datos del segundo ordenador se han simulado siguiendo las indicaciones proporcionadas: se ha tomado como base el tiempo medio obtenido en el primer ordenador y se ha aplicado una variación aleatoria del ±10% sobre cada tiempo individual.

## Gráficas de tiempos de respuesta



La gráfica muestra el **tiempo promedio de respuesta por funcionalidad** de la aplicación, medido durante las pruebas funcionales. A partir de los datos representados, se puede destacar lo siguiente:

* **Funcionalidades con mayor tiempo de respuesta:**
  + La creación de booking record (customer → booking record → create), con un tiempo cercano a 140 ms.
  + La creación y publicación de bookings (customer → booking → create y customer → booking → publish), ambas en torno a los 120 ms.
  + La creación de passenger (customer → passenger → create), con más de 100 ms.
  + Otras funcionalidades como la eliminación de bookings o la publicación de booking record también superan los 90 ms.
* Estas funcionalidades suelen incluir múltiples validaciones, acceso a entidades relacionadas y comprobaciones de seguridad, lo que explica los tiempos elevados.
* **Funcionalidades con menor tiempo de respuesta:**
  + Acciones del sistema como iniciar sesión, cerrar sesión o mostrar la página de bienvenida tienen tiempos inferiores a 10 ms.
  + Las funcionalidades de listado como customer → passenger → list o customer → booking → list rondan los 20 ms, siendo bastante eficientes.

#### Promedio general

* El **promedio general** de tiempo de respuesta de todas las funcionalidades es de aproximadamente 40 ms, lo que indica un rendimiento general muy aceptable para una aplicación de estas características.

#### Conclusión

El rendimiento observado es adecuado en términos generales. Las funcionalidades más lentas coinciden con aquellas que requieren lógica compleja o múltiples validaciones. No se han detectado anomalías significativas, aunque si se quisiera optimizar, se podría comenzar por las acciones de creación y publicación de entidades como booking y booking record.

## Estadísticas

A continuación se presentan los principales estadísticos descriptivos obtenidos a partir del análisis de los tiempos de respuesta (en milisegundos) de las peticiones realizadas durante las pruebas funcionales:

* **Media**: 21,96 ms
* **Mediana**: 2,66 ms
* **Moda**: 0,52 ms
* **Desviación estándar**: 39,45 ms
* **Varianza**: 1555,94 ms²
* **Curtosis**: 8,31
* **Coeficiente de asimetría**: 2,59
* **Rango**: 303,66 ms
* **Mínimo**: 0,27 ms
* **Máximo**: 303,92 ms
* **Número de muestras**: 680

El coeficiente de asimetría y la curtosis indican que los datos presentan una distribución **asimétrica hacia la derecha** (muchas respuestas rápidas, pero algunas muy lentas) y una **distribución leptocúrtica**, es decir, con colas largas.

### **Intervalo de confianza**

Se ha calculado un **intervalo de confianza del 95 %** para la media del tiempo de respuesta. El resultado en milisegundos es:

* **[18,99 ms – 24,93 ms]**

Este intervalo indica que, con un 95 % de confianza, el verdadero tiempo medio de respuesta de la aplicación se encuentra dentro de ese rango.

Convertido a segundos, el intervalo de confianza es:

* **[0,0189 s – 0,0249 s]**

Dado que el intervalo se encuentra completamente dentro del rango **entre 0 y 1 segundo**, **se cumple la condición de rendimiento establecida** para el sistema, demostrando que el comportamiento temporal de la aplicación es adecuado y consistente.

## Hipótesis contrastada

| **Métrica** | **Antes** | **Después** |
| --- | --- | --- |
| Media | 21,96 ms | 20,47 ms |
| Error típico | 1,51 ms | 1,34 ms |
| Mediana | 2,66 ms | 2,41 ms |
| Moda | 0,52 ms | 0,52 ms |
| Desviación estandar | 39,45 ms | 35,20 ms |
| Varizanza | 1555,94 ms² | 1239,04 ms² |
| Curtosis | 8,31 | 7,95 |
| Coef. de asimetría | 2,59 | 2,44 |
| Rango | 303,66 ms | 290,73 ms |
| Mínimo | 0,27 ms | 0,27 ms |
| Máximo | 303,92 ms | 301,00 ms |
| Nivel de confianza 95 % | ±2,97 ms | ±2,63 ms |
| Intervalo confianza (ms) | [18,99 – 24,93] ms | [17,84 – 23,10] ms |
| Intervalo confianza (s) | [0,0189 – 0,0249] s | [0,0178 – 0,0231] s |

Tras realizar una **optimización del código**, se ha vuelto a ejecutar el conjunto de pruebas funcionales, obteniéndose resultados **ligeramente mejores**:

* **Reducción de la media**: El tiempo medio de respuesta ha bajado de **21,96 ms a 20,47 ms**, lo que representa una mejora del **6,8 %** aproximadamente.
* **Menor dispersión**: La desviación estándar ha pasado de **39,45 ms a 35,20 ms**, lo que sugiere que los tiempos de respuesta son ahora más consistentes.
* **Mejora del intervalo de confianza**: El nuevo intervalo al 95 % es **[17,84 – 23,10] ms**, más estrecho que el anterior, lo que implica mayor precisión en la estimación del tiempo medio.
* **Rango y máximo también se reducen**, lo que indica que incluso las peticiones más lentas fueron más rápidas que antes.
* **Los indicadores de curtosis y asimetría** bajan ligeramente, lo cual sugiere que la distribución de los tiempos es ahora más cercana a una distribución normal, con menos valores extremos.

El contraste de hipótesis se ha realizado con un nivel de significación del 5 % (α = 0.05), siendo la hipótesis nula que **ambas medias son iguales** (es decir, que no hay diferencia real entre los tiempos antes y después de la mejora).

Los resultados del Z-Test han sido:

* Estadístico Z: **0,7344**
* Valor p (two-tail): **0,4627**

Dado que el valor p es **mayor que 0,05**, no se puede rechazar la hipótesis nula. Es decir, **no hay evidencia estadísticamente significativa de que los tiempos de ejecución hayan mejorado tras las modificaciones realizadas**.

No obstante, aunque los resultados no son significativos desde un punto de vista estadístico, **sí se ha observado una ligera reducción en la media del tiempo de respuesta**, lo cual puede seguir siendo positivo si las mejoras han implicado una simplificación del código, una reducción de errores o una mejor organización del sistema.

### **Conclusión**

El análisis de rendimiento ha demostrado que el sistema responde correctamente dentro de los márgenes esperados de eficiencia. En concreto, los tiempos de respuesta de todas las funcionalidades se encuentran dentro del intervalo recomendado de 0 a 1 segundo, siendo la media general significativamente inferior a ese límite.

Además, se han llevado a cabo mejoras en el código con el objetivo de optimizar el rendimiento. Aunque el contraste estadístico (Z-Test) no ha mostrado una diferencia significativa en los tiempos de ejecución tras los cambios, se ha registrado una ligera mejora en la media del tiempo de respuesta, lo cual refleja una tendencia positiva.

En conjunto, el sistema presenta un comportamiento eficiente y robusto en cuanto a tiempos de respuesta, lo que garantiza una buena experiencia de usuario y evidencia que la aplicación está preparada para su despliegue en un entorno real.