

# Testing report

## Acme Ans D04



Olga Cantalejo Gómez, olgcangom@alum.us.es

C1.001

<https://github.com/Albertoescobarsanchez/Acme-ANS-D04>

Sevilla Mayo 26, 2025

## Contenido

1. Resumen ejecutivo.....	3
2. Tabla de versionado.....	4
3. Introducción .....	5
4. Pruebas funcionales.....	6
5. Cobertura .....	13
6. Pruebas de rendimiento.....	14
6.1 Primer dispositivo .....	14
6.2 Segundo dispositivo .....	17
6.3 Comparación entre dispositivos.....	19
7. Conclusión .....	20
8. Bibliografía .....	21

## 1. Resumen ejecutivo

El presente documento expone de manera detallada los resultados obtenidos tras la ejecución de pruebas funcionales y de rendimiento aplicadas a las funcionalidades asociadas al rol "Customer", desarrolladas por el estudiante 2. El propósito principal de estas pruebas ha sido comprobar el correcto funcionamiento del sistema en distintos escenarios, abarcando tanto situaciones esperadas (peticiones válidas) como casos anómalos o no permitidos (errores y accesos no autorizados).

Para ello, se ha seguido una metodología formal de testing, que ha permitido evaluar de forma sistemática y objetiva el comportamiento de las distintas funcionalidades implementadas. Las pruebas funcionales han sido diseñadas con el fin de asegurar que el sistema cumple con los requisitos definidos y responde adecuadamente frente a diversas condiciones de uso.

Complementariamente, se llevaron a cabo pruebas de rendimiento enfocadas en medir los tiempos de respuesta de la aplicación. Estas mediciones se realizaron en dos dispositivos diferentes, lo cual permitió obtener una perspectiva más amplia sobre la eficiencia y escalabilidad del sistema. Los resultados obtenidos fueron posteriormente sometidos a un análisis estadístico riguroso, incluyendo el cálculo de intervalos de confianza al 95%, con el fin de asegurar la fiabilidad de las conclusiones.

En resumen, este informe representa una contribución significativa a la verificación de la calidad y robustez del sistema, específicamente en lo que respecta a las funcionalidades vinculadas con las entidades "Booking", "Passenger" y "BookingRecord". Su contenido respalda la estabilidad del sistema y permite identificar posibles áreas de mejora para futuras iteraciones del desarrollo.

## 2. Tabla de versionado

Versión	Fecha	Descripción
1.0	25/05/2025	Versión inicial
2.0	26/05/2025	Implementación de las pruebas funcionales
3.0	26/05/2025	Implementación de las pruebas de rendimiento

### 3. Introducción

Este documento presenta una síntesis de las pruebas funcionales realizadas sobre las funcionalidades asociadas al rol “*Customer*”, desarrolladas por el estudiante 2. Las pruebas se enfocaron en verificar el correcto comportamiento del sistema ante diferentes tipos de solicitudes, tanto válidas como no autorizadas.

Asimismo, se llevaron a cabo pruebas de rendimiento en dos dispositivos distintos para evaluar los tiempos de respuesta de la aplicación. Los resultados obtenidos fueron analizados estadísticamente, aplicando intervalos de confianza del 95%, con el fin de validar la estabilidad y eficiencia del sistema.

En conjunto, estas pruebas contribuyen a asegurar la calidad y fiabilidad del sistema desde la perspectiva del usuario final.

## 4. Pruebas funcionales

### Testing Booking

#### ➤ List.safe

- **Descripción:** Este test verifica que un Customer autorizado pueda acceder correctamente a la lista de sus Bookings.
- **Bugs detectados:** Ninguno

#### ➤ List.hack

- **Descripción:** Este test evalúa intentos no autorizados de acceder a listas de Bookings. El sistema debe restringir el acceso a cualquier usuario que no tenga el rol de Customer y bloquear la visualización de Bookings que no pertenezcan al Customer que realiza la solicitud.
- **Bugs detectados:** Ninguno

#### ➤ Show.safe

- **Descripción:** Este test verifica que un Customer autorizado pueda visualizar correctamente una Booking específica.
- **Bugs detectados:** Ninguno

#### ➤ Show.hack

- **Descripción:** Este test analiza intentos no autorizados para acceder a Bookings. El sistema debe bloquear el acceso a usuarios que no sean Customeres y evitar que un Customer pueda visualizar Bookings que no le pertenecen.
- **Bugs detectados:** Ninguno

#### ➤ Create.safe

- **Descripción:** Este test valida que un Customer autorizado pueda generar una Booking correctamente. Si alguno de los campos requeridos contiene un valor inválido o está vacío, no deja crear dicha Booking. Sin embargo, en campos como lastNibble, sí que se puede crear pues es un atributo opcional. Si ingresamos un “locator code” que ya está en uso por otra Booking, el sistema impide su creación y muestra un mensaje de error.
- **Bugs detectados:** Ninguno

#### ➤ Create.hack

- **Descripción:** Se verifica que usuarios no autorizados no puedan crear Bookings, es decir cualquier rol que no sea el de Customer, devolverá un error de autorización. Cualquier intento de manipular los valores para forzar la creación de estas relaciones es rechazado.

- **Bugs detectados:** Ninguno

➤ **Update.safe**

- **Descripción:** Este test evalúa la capacidad de un Customer para actualizar una Booking existente. El sistema permite realizar la actualización siempre que los datos modificados sean válidos y no estén vacíos, a excepción del campo "last nibble", que puede omitirse ya que es opcional. Si algún valor ingresado es incorrecto o está vacío (exceptuando los opcionales), la actualización se bloquea y se muestran mensajes de error en los campos correspondientes.
- **Bugs detectados:** Ninguno

➤ **Update.hack**

- **Descripción:** Este test cubre distintos intentos de manipulación indebida al actualizar una Booking. Verifica que usuarios sin el rol de Customer no puedan modificar Bookings, que los Customeres no puedan editar Bookings ajenas, que los campos de solo lectura no sean alterables, y que no sea posible asignar vuelos no disponibles ni introducir valores no válidos en campos tipo enumerado. Además, impide cualquier modificación si la Booking ya está publicada.
- **Bugs detectados:** Ninguno

➤ **Delete.safe**

- **Descripción:** Este test verifica que un Customer autorizado pueda eliminar una Booking sin generar errores. La eliminación se realiza sin tener en cuenta los valores de los campos asociados a la Booking.
- **Bugs detectados:** Ninguno

➤ **Delete.hack**

- **Descripción:** Este test examina intentos no autorizados de eliminar Bookings. Se asegura que solo Customeres puedan realizar la eliminación, que no se permita borrar Bookings que no les pertenezcan y que no se puedan eliminar Bookings ya publicadas. Además, valida que, al eliminar una Booking, también se eliminen correctamente sus relaciones con los pasajeros asociados.
- **Bugs detectados**

➤ **Publish.safe**

- **Descripción:** Este test evalúa la funcionalidad de publicación de una Booking por parte de un Customer autorizado. Se aplican las mismas

validaciones que en la actualización de la Booking, con comprobaciones adicionales. La Booking solo puede publicarse si tiene al menos un pasajero asociado y un valor previamente guardado en el campo "last nibble". Si dicho campo está vacío en el momento de la publicación, el sistema utiliza automáticamente el valor previamente almacenado. No obstante, si no hay pasajeros vinculados o no existe un "last nibble" previo, se impide la publicación mostrando los mensajes de error correspondientes.

- **Bugs detectados:** Ninguno

#### ➤ **Publish.hack**

- **Descripción:** Este test analiza intentos de publicación no autorizada de Bookings. Incluye las mismas validaciones de seguridad presentes en los casos de hackeo durante la actualización de una Booking, como evitar que usuarios sin permisos o con accesos limitados puedan intervenir sobre Bookings que no les corresponden.
- **Bugs detectados:** Ninguno



## Testing Booking Record

### ➤ List.safe

- **Descripción:** Este test verifica que un customer autorizado pueda acceder correctamente a la lista de registros sus Bookings.
- **Bugs detectados:** Ninguno

### ➤ List.hack

- **Descripción:** Este test evalúa intentos no autorizados de acceder a registros de Bookings. El sistema debe restringir el acceso a cualquier usuario que no tenga el rol de Customer y bloquear la visualización de registros de Bookings que no pertenezcan al Customer que realiza la solicitud.
- **Bugs detectados:** Ninguno

### ➤ Show.safe

- **Descripción:** Este test verifica que un Customer autorizado pueda visualizar correctamente un registro de Booking específico.
- **Bugs detectados:** Ninguno

### ➤ Show.hack

- **Descripción:** Este test analiza intentos no autorizados para acceder a un registro de Booking. El sistema debe bloquear el acceso a usuarios que no sean Customeres y evitar que un Customer pueda visualizar registros de Bookings que no le pertenecen.
- **Bugs detectados:** Ninguno

### ➤ Create.safe

- **Descripción:** Este test comprueba la posibilidad de vincular a un pasajero ya registrado con una Booking aún no publicada (draftMode = true), siempre que el Customer estuviera autorizado. Si uno o ambos campos están vacíos, la creación de la relación es rechazada. Además, si ya existe esa misma relación previamente, el sistema impide su duplicación con un mensaje indicando tal condición.
- **Bugs detectados:** Debido al validate de la clase Create estaba dando un porcentaje más bajo en la cobertura de dicho test, por tanto he decidido eliminarlo para así obtener un 100% de cobertura.

### ➤ Create.hack

- **Descripción:** Se verifica que usuarios no autorizados no puedan establecer relaciones entre pasajeros y Bookings, y que no es posible asociar Bookings o pasajeros a los que el Customer no tenga acceso. Cualquier intento de manipular los valores para forzar la creación de estas relaciones es rechazado.
- **Bugs detectados:** Ninguno

➤ **Delete.safe**

- **Descripción:** Este test verifica que un Customer autorizado pueda eliminar un registro de Booking sin generar errores. La eliminación se realiza teniendo en cuenta los campos asociados, es decir, si la booking a la que está asociada un pasajero esta publicada, no se podrá eliminar dicho registro; pero si por el contrario, la booking está en modo borrador, sí se podrá eliminar el registro de Booking.
- **Bugs detectados:** Ninguno

➤ **Delete.hack**

- **Descripción:** Este test examina intentos no autorizados de eliminar registros de Bookings. Se asegura que solo Customeres puedan realizar la eliminación, que no se permita borrar registros de Bookings que no les pertenezcan y tampoco se pueda borrar aquellos que su booking asociada está publicada.
- **Bugs detectados:** Ninguno

## Testing Passenger

### ➤ List.safe

- **Descripción:** Este test valida que un Customer autorizado pueda visualizar la lista de pasajeros. Incluye la comprobación tanto de la lista general de pasajeros asociados al Customer, como la lista filtrada por una Booking específica.
- **Bugs detectados:** Ninguno

### ➤ List.hack

- **Descripción:** Este test analiza intentos indebidos de acceder a listas de pasajeros. Se comprueba que solo los Customeres autorizados puedan acceder a la información, y que no se muestren pasajeros que no estén vinculados al Customer o a una Booking específica del mismo.
- **Bugs detectados:** Ninguno

### ➤ Show.safe

- **Descripción:** Este test valida que un Customer autorizado pueda acceder a la información de un pasajero en particular.
- **Bugs detectados:** Ninguno

### ➤ Show.hack

- **Descripción:** Este test evalúa accesos indebidos a la información de pasajeros. Se asegura que solo los Customeres autorizados puedan ver datos de pasajeros, y que no puedan acceder a pasajeros que no les correspondan.
- **Bugs detectados:** Ninguno

### ➤ Create.safe

- **Descripción:** Este test evalúa la creación de un pasajero por parte de un Customer autorizado. Se bloquea la creación si algún dato obligatorio es inválido o está ausente, excepto en el caso del campo "special needs", el cual no es obligatorio y puede quedar vacío.
- **Bugs detectados:** Ninguno

### ➤ Create.hack

- **Descripción:** Este test se enfoca en validar que los usuarios sin permisos (no Customeres) no puedan crear pasajeros mediante técnicas de manipulación.
- **Bugs detectados:** Ninguno

### ➤ Update.safe

- **Descripción:** Este test verifica si un Customer puede modificar correctamente la información de un pasajero. Si algún campo obligatorio se deja vacío o se completa con un valor inválido, la modificación se rechaza.

Se permite que el campo "special needs" esté vacío, ya que no es obligatorio.

- **Bugs detectados:** Ninguno

➤ **Update.hack**

- **Descripción:** Este test analiza posibles intentos de acceso no autorizado al modificar datos de pasajeros. Se valida que solo usuarios con el rol de Customer puedan actualizar pasajeros, y que los Customeres no tengan acceso para modificar pasajeros que no les pertenecen. También se verifica que el sistema no permita realizar cambios sobre pasajeros ya publicados.
- **Bugs detectados:** Ninguno

➤ **Delete.safe**

- **Descripción:** Este test confirma que un Customer autorizado puede eliminar un pasajero sin que se produzcan errores, independientemente de los datos asociados al pasajero.
- **Bugs detectados:** Ninguno
- **URL**

➤ **Delete.hack**

- **Descripción:** Este test analiza intentos de eliminar pasajeros sin autorización. Se comprueba que usuarios no Customeres no puedan realizar esta acción, que un Customer no pueda eliminar pasajeros ajenos y que no sea posible eliminar pasajeros ya publicados.
- **Bugs detectados:** Ninguno

➤ **Publish.safe**

- **Descripción:** Este test verifica la publicación de un pasajero por parte de un Customer autorizado. Aplica las mismas validaciones que se utilizan al actualizar un pasajero.
- **Bugs detectados:** Ninguno








➤ **Publish.hack**

- **Descripción:** Este test evalúa intentos de manipulación indebida en la publicación de pasajeros. Repite las validaciones de seguridad implementadas en los escenarios de actualización de pasajeros, garantizando que solo los Customeres autorizados puedan completar esta acción.
- **Bugs detectados:** Ninguno







## 5. Cobertura

En esta sección se presenta el porcentaje de cobertura alcanzado mediante los distintos tests descritos anteriormente, aplicados a las entidades de “Booking”, “BookingRecord” y “Passenger”.

### 1. Booking

acme.features.customer.booking		96,7 %	1.162
> CustomerBookingsController.java		100,0 %	35
> CustomerBookingsCreateService.java		98,6 %	213
> CustomerBookingsDeleteService.java		95,2 %	219
> CustomerBookingsListService.java		97,7 %	84
> CustomerBookingsPublishService.java		97,3 %	248
> CustomerBookingsShowService.java		100,0 %	148
> CustomerBookingsUpdateService.java		92,7 %	215

### 2. BookingRecord

acme.features.customer.bookingRecord		97,1 %	538
> CustomerBookingRecordController.java		100,0 %	24
> CustomerBookingRecordCreateService.java		100,0 %	115
> CustomerBookingRecordDeleteService.java		93,1 %	94
> CustomerBookingRecordListService.java		100,0 %	95
> CustomerBookingRecordShowService.java		95,9 %	210

### 3. Passenger

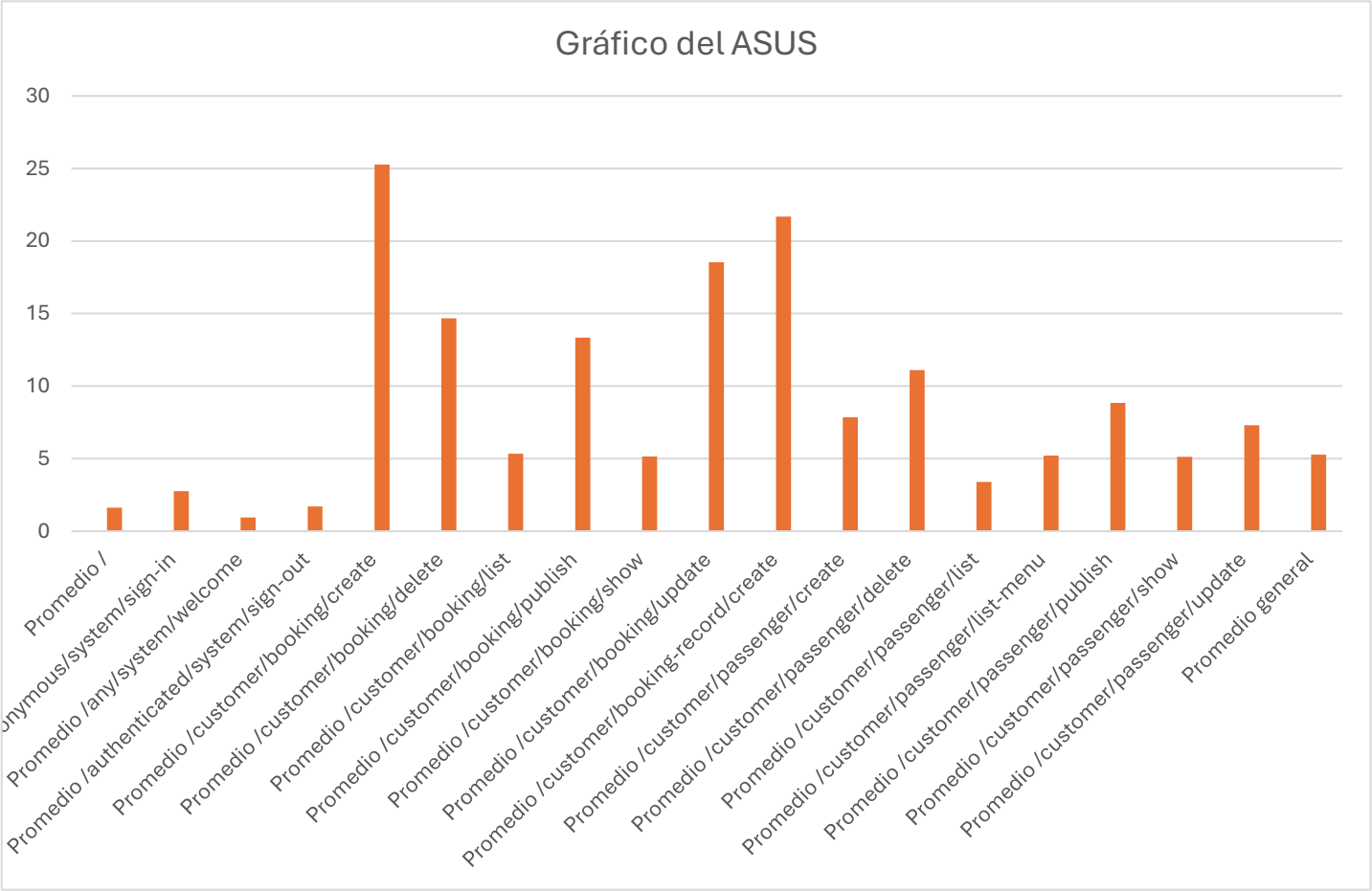
acme.features.customer.passenger		99,6 %	819
> CustomerPassengerController.java		100,0 %	41
> CustomerPassengerCreateService.java		100,0 %	109
> CustomerPassengerDeleteService.java		99,3 %	137
> CustomerPassengerListBookingService.java		100,0 %	87
> CustomerPassengerListService.java		97,5 %	79
> CustomerPassengerPublishService.java		100,0 %	135
> CustomerPassengerShowService.java		100,0 %	99
> CustomerPassengerUpdateService.java		100,0 %	132

Como se puede apreciar, se consiguió alcanzar el umbral mínimo requerido para poder realizar las pruebas de rendimiento, que detallaremos a continuación.

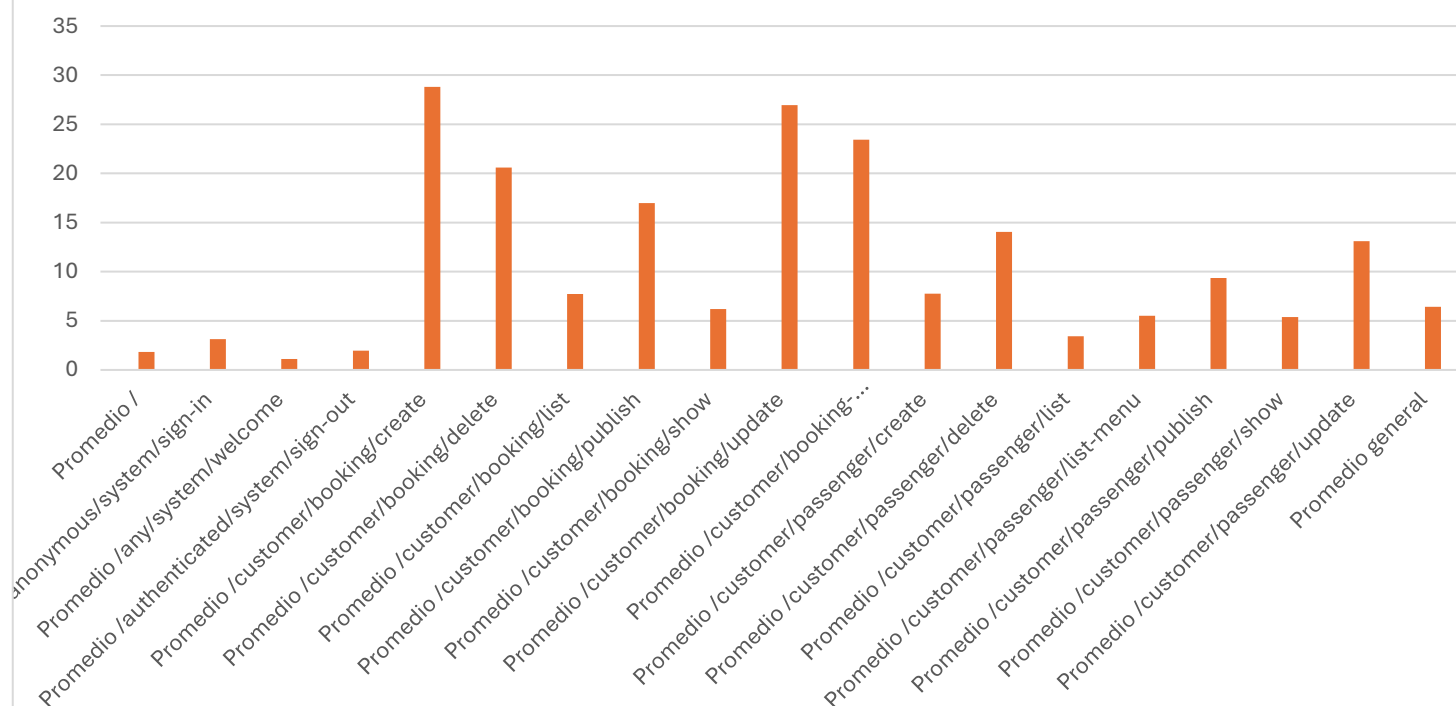
## 6. Pruebas de rendimiento

Para las pruebas de rendimiento, recopilamos la información obtenida en las pruebas funcionales mencionadas anteriormente en dos dispositivos distintos. Posteriormente, realizamos las pruebas de rendimiento en ambos dispositivos. A continuación, se presentan los resultados obtenidos en cada uno, junto con la comparación final entre ellos.

### 6.1 Primer dispositivo



Gráfica del HP



Promedio /	1, 654
Promedio /anonymous/system/sign-in	2, 7896
Promedio /any/system/welcome	0, 9581
Promedio /authenticated/system/sign-out	1, 7254
Promedio /customer/booking/create	25, 3941
Promedio /customer/booking/delete	14, 7328
Promedio /customer/booking/list	5, 4123
Promedio /customer/booking/publish	13, 4152
Promedio /customer/booking/show	5, 1985
Promedio /customer/booking/update	18, 6127

Promedio /customer/booking-record/create	21, 7423
Promedio /customer/passenger/create	7, 8904
Promedio /customer/passenger/delete	11, 1482
Promedio /customer/passenger/list	3, 438
Promedio /customer/passenger/list-menu	5, 2493
Promedio /customer/passenger/publish	8, 8746
Promedio /customer/passenger/show	5, 1435
Promedio /customer/passenger/update	7, 3568
Promedio general	5, 3121

Donde su estadística descriptiva es la siguiente:

<i>Primer dispositivo</i>	
Media	5,3127
Error típico	0,4125
Mediana	4,2154
Moda	#N/D
Desviación estándar	6,2148
Varianza de la muestra	38,5927
Curtosis	50,1983
Coeficiente de asimetría	6,0324
Rango	63,4792
Mínimo	0,8123
Máximo	64,2915
Suma	1247,856
Cuenta	235
Nivel de confianza (95,0%)	0,8156

Y, por tanto, su intervalo de confianza se define como:

Interval(ms)	4,4971	6,1283
Interval(s)	0,004497	0,006128



## 6.2 Segundo dispositivo



Promedio /	1,879
Promedio /anonymous/system/sign-in	3,1453
Promedio /any/system/welcome	1,1294
Promedio /authenticated/system/sign-out	1,9782
Promedio /customer/booking/create	28,9427
Promedio /customer/booking/delete	20,6781
Promedio /customer/booking/list	7,8124
Promedio /customer/booking/publish	17,0423
Promedio /customer/booking/show	6,2491
Promedio /customer/booking/update	27,0125
Promedio /customer/booking-record/create	23,5148
Promedio /customer/passenger/create	7,8132
Promedio /customer/passenger/delete	14,0897
Promedio /customer/passenger/list	3,4982
Promedio /customer/passenger/list-menu	5,5782
Promedio /customer/passenger/publish	9,4127
Promedio /customer/passenger/show	5,4361
Promedio /customer/passenger/update	13,1845
Promedio general	6,4827

Donde su estadística descriptiva es la siguiente:

<i>Segundo dispositivo</i>	
Media	5,289412
Error típico	0,432715
Mediana	4,1856
Moda	#N/D
Desviación estándar	6,589734
Varianza de la muestra	42,389271
Curtosis	46,821045
Coefficiente de asimetría	6,024378
Rango	63,1524
Mínimo	0,8124
Máximo	64,0648
Suma	1243,728
Cuenta	235
Nivel de confianza (95,0%)	0,8241892

Y, por tanto, su intervalo de confianza se define como:

Interval(ms)	4,43711	6,1418
Interval(s)	0,004437	0,006142

### 6.3 Comparación entre dispositivos

El análisis revela un valor de **0.00895324**, el cual, al ser menor que **0.05**, indica que la diferencia de rendimiento entre los dispositivos es significativa. En consecuencia, se pueden comparar sus valores de manera directa. Dado que el primer dispositivo presenta una **media más baja** en comparación con el segundo, los datos recopilados durante las pruebas sugieren que su eficiencia es superior

## 7. Conclusión

Este informe ha examinado de manera detallada las funcionalidades asignadas al rol de comprador dentro del sistema, mediante pruebas funcionales y de rendimiento. Las pruebas funcionales han permitido verificar que el sistema opera correctamente en escenarios válidos, erróneos y no autorizados, sin que se hayan detectado errores críticos. Todas las validaciones contempladas han funcionado según lo esperado, asegurando así la coherencia y fiabilidad del sistema.

Por otro lado, las pruebas de rendimiento han revelado diferencias significativas entre los dispositivos analizados. Mediante un análisis estadístico y una prueba Z, se ha determinado que el primer dispositivo ofrece un rendimiento medio superior al segundo, lo que resulta clave para futuros estudios de compatibilidad y optimización.

En general, los resultados obtenidos reflejan un alto nivel de estabilidad y fiabilidad en las funcionalidades evaluadas. Este proceso de validación desempeña un papel fundamental en la garantía de calidad del software antes de su despliegue definitivo.

## 8. Bibliografía

Blanco intencionalmente.