Universidad de Sevilla  
Escuela Técnica Superior de Ingeniería Informática

**Diseño y Pruebas II**

**Testing Report**

**D04-Student 2**



Grado en Ingeniería Informática – Ingeniería del Software  
Curso 2024 – 2025

|  |  |  |
| --- | --- | --- |
| **Fecha** | **Versión** | **Repositorio** |
| 26/05/2025 | 1.0 | https://github.com/javsorbla/Acme-ANS-D04 |

|  |  |  |
| --- | --- | --- |
| **Grupo de prácticas: C1.032** | |  |
| **Autores por orden alfabético** | **Roles** | **Correo Corporativo** |
| Castilla Rodríguez, Javier | Tester, Developer | javcasrod1@alum.us.es |
| De los Reyes Pérez, Alejandro | Operator, Tester, Developer | alereyper@alum.us.es |
| Del Pino Escalante, Nuno José | Analyst Tester, Developer | nundelesc@alum.us.es |
| Gutiérrez Pastor, Javier | Tester, Developer | javgutpas@alum.us.es |
| Soria Blanco, Javier | Manager, Tester, Developer | javsorbla@alum.us.es |

**Fecha:** 26/05/2025

**Tabla de contenidos**

[**1.** **Resumen Ejecutivo**: 2](#_Toc199187465)

[**2.** **Tabla de revisión**: 3](#_Toc199187466)

[**3.** **Introducción**: 4](#_Toc199187467)

[**4.** **Contenidos**: 5](#_Toc199187468)

[**4.1** **Testing funcional**: 5](#_Toc199187469)

[**5.** **Rendimiento de testing:** 13](#_Toc199187470)

[**5.1** **Resultados de la primera prueba (sin índices)**: 13](#_Toc199187471)

[**5.2** **Resultados de la primer prueba (con índices):** 15](#_Toc199187472)

[**5.3** **Resultados de la segunda prueba (sin índices en ordenador local):** 16](#_Toc199187473)

[**5.4** **Resultados de la segunda prueba (sin índices en otro ordenador):** 16](#_Toc199187474)

[**5.5** **Primera pruebaZ (mismo ordenador sin índices – con índices):** 18](#_Toc199187475)

[**5.6** **Segunda pruebaZ (distinto ordenador sin índices)** 19](#_Toc199187476)

[6 **Conclusiones**: 20](#_Toc199187477)

[7 **Bibliografía**: 21](#_Toc199187478)

### **Resumen Ejecutivo**:

El presente documento tiene como objetivo proporcionar un análisis de esta última fase de testing que conlleva con la finalización de la entrega del proyecto. A través de este documento, se busca detallar los distintos requisitos establecidos, con la finalidad de justificar y validar las soluciones propuestas y tomadas para cumplir con los mismos.

### **Tabla de revisión**:

|  |  |  |
| --- | --- | --- |
| **Fecha** | **Versión** | **Descripción** |
| 26/05/2025 | 1.0 | Versión inicial del documento |
|  |  |  |
|  |  |  |
|  |  |  |

### **Introducción**:

Este informe ofrece una explicación de los requisitos establecidos para la fase de testing de esta última entrega, enfocándose en aquellos que requieren un estudio más profundo. Su propósito es documentar y justificar las acciones tomadas para garantizar el cumplimiento de dichos requisitos.

Durante el período transcurrido por el informe, se han realizado diversas tareas que han implicado la toma de decisiones clave durante la fase de prueba (testing), las cuales deben ser validadas para asegurar el cumplimiento adecuado de los requisitos y la satisfacción del cliente.

### **Contenidos**:

### **Testing funcional**:

En esta sección se presentarán de manera general los distintos casos de prueba (tanto para el .safe como para el hacking (.hack)), correspondiente a los requisitos individuales 8 y 9 del Student 2. Se dará una breve descripción de los mismos y se comentará si se encontró algún bug.

**Funcionalidades de Reservas (*Booking*):**

|  |  |  |
| --- | --- | --- |
| **.safe** | **Descripción** | **Bugs encontrados** |
| List | Se comprobó que las distintas bookings se listaban correctamente. | Ninguno |
| Show | Se comprobó que al entrar en un booking los detalles del mismo se mostraban correctamente junto a sus botones que permitían las distintas acciones. | Ninguno |
| Create | Se comprobó que los bookings se creaban correctamente, para ello se iteraron las distintas entradas con todos los valores posibles, tanto erróneos como correctos, de uno en uno. | Ninguno |
| Update | Se comprobó que los bookings se actualizaban correctamente, para ello se iteraron las distintas entradas con todos los valores posibles, tanto erróneos como correctos, de uno en uno. | Ninguno |
| Publish | Se comprobó que los bookings se publicaban correctamente, para ello se iteraron las distintas entradas con todos los valores posibles, tanto erróneos como correctos, de uno en uno. | Ninguno |
| Delete | Se comprobó que los bookings se eliminaban correctamente, para ello se probó a eliminar bookings con pasajeros asociados y sin pasajeros. | Ninguno |

|  |  |  |
| --- | --- | --- |
| **.hack** | **Descripción** | **Bugs encontrados** |
| List | Se probó si se podía acceder al listado de bookings sin estar logueado en la aplicación. | Ninguno |
| Show | Se probó acceder al show sin estar logueado , acceder a show de bookings de otros usuarios, acceder a un booking publicado, otro sin publicar y uno inexistente todo esto con otros customer que no eran los que habían creado dicha reserva. | Ninguno |
| Create | Se probó a acceder al créate sin estar logueado y se probó a hackear con el customer 1, el selector del flight poniendo ids que no existen y de bookings publicados. El selector de travel Class se hackeo para poner valores incorrectos y los valores readOnly se modificaron para ver si permitía el cambio. | Ninguno |
| Update | Se comprobó a acceder sin estar logueado a la url que permitía actualizar un booking sin publicar, otro publicado y otro inexistente, posteriormente se intentó lo mismo con otro customer, (customer 2). | Ninguno |
| Publish | Se comprobó a acceder sin estar logueado a la url que permitía publicar un booking sin publicar, otro publicado y otro inexistente, posteriormente se intentó lo mismo con otro customer, (customer 2). También se probó las restricciones de que no se podía publicar un bookign con pasajeros sin publicar o sin pasajeros asociados. Por último, se modificó el id y el value por valores inexistente (999) además se probó hackear el travelClass y los atributos readOnly. | Ninguno |
| Delete | Se comprobó a acceder sin estar logueado a la url que permitía eliminar un booking sin publicar, otro publicado y otro inexistente, posteriormente se intentó lo mismo con otro customer, (customer 2). | Ninguno |

**Funcionalidades de Reservas-pasajeros (*BookingRecord*):**

|  |  |  |
| --- | --- | --- |
| **.safe** | **Descripción** | **Bugs encontrados** |
| List | Se comprobó que los pasajeros de un booking se listaban correctamente. | Ninguno |
| Create | Se comprobó que se podían asociador pasajeros a un bookign correctamente para ello se eligieron valores vacíos en el formulario y valores correctos. | Ninguno |
| Delete | Se comprobó que se podía eliminar pasajeros asociados a un booking siempre y cuando éste esté sin publicar y tenga algún pasajero. | Ninguno |

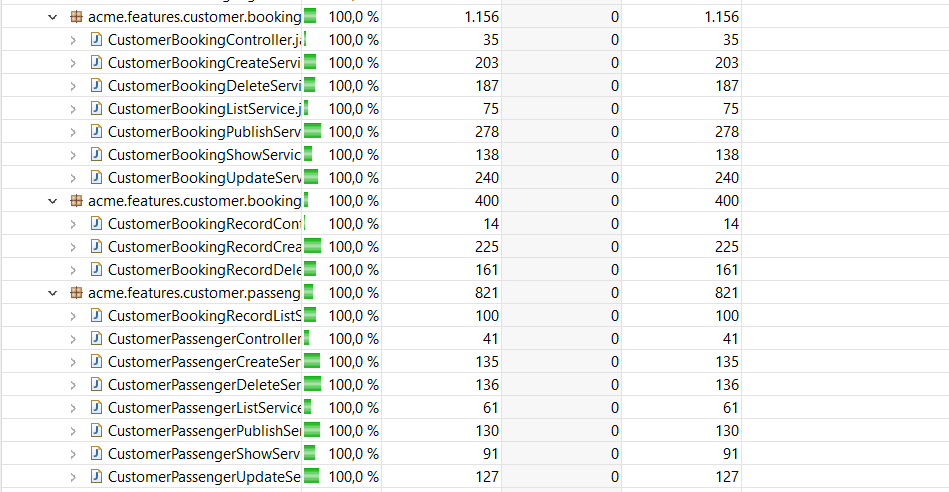
|  |  |  |
| --- | --- | --- |
| **.hack** | **Descripción** | **Bugs encontrados** |
| List | Se probó acceder a ver los pasajeros de un booking sin estar logueado y luego se probó lo mismo con otro customer que no era el creador de dicha booking. | Ninguno |
| Create | Se probó a añadir pasajeros a un booking sin publicar, publicado, que no existe, de otro customer y otro inexistente, todo ello sin tener sesión iniciada, posteriormente se hizo lo mismo para otro customer (customer 2). Por último, se probó con el customer 1 ha hackear los id de los pasajeros a añadir, con ids incorrectos, id de pasajeros que no había creado el customer y un id de un pasajero que ya estaba en el booking, además se probó a hackear el id y el versión con valores incorrectos (999). | No daba un error “Access not authorise” al intentar añadir un pasajero a un booking con id inexistente. |
| Delete | Se probó a eliminar pasajeros de un booking sin publicar, publicado, que no existe, perteneciente a otro customer y otro inexistente, todo ello sin tener sesión iniciada, posteriormente se hizo lo mismo para otro customer (customer 2). | Ninguno |

**Funcionalidades de Pasajeros (*Passenger*)**

|  |  |  |
| --- | --- | --- |
| **.safe** | **Descripción** | **Bugs encontrados** |
| List | Se comprobó que los distintos passengers se listaban correctamente. | Ninguno |
| Show | Se comprobó que al entrar en un passenger los detalles del mismo se mostraban correctamente junto a sus botones que permitían las distintas acciones. | Ninguno |
| Create | Se comprobó que los passengers se creaban correctamente, para ello se iteraron las distintas entradas con todos los valores posibles, tanto erróneos como correctos, de uno en uno. | Ninguno |
| Update | Se comprobó que los passengers se actualizaban correctamente, para ello se iteraron las distintas entradas con todos los valores posibles, tanto erróneos como correctos, de uno en uno. | Ninguno |
| Publish | Se comprobó que los passengers se publicaban correctamente, para ello se iteraron las distintas entradas con todos los valores posibles, tanto erróneos como correctos, de uno en uno. | Ninguno |
| Delete | Se comprobó que los bookings se eliminaban correctamente, para ello se probó a eliminar bookings con pasajeros asociados y sin pasajeros. | Ninguno |

|  |  |  |
| --- | --- | --- |
| **.hack** | **Descripción** | **Bugs encontrados** |
| List | Se probó si se podía acceder al listado de passengers sin estar logueado en la aplicación. | Ninguno |
| Show | Se probó acceder al show sin estar logueado , acceder a show de passengers de otros usuarios, acceder a un passenger publicado, otro sin publicar y uno inexistente, todo esto con otros customer que no eran los que habían creado dicho Passenger. | Ninguno |
| Create | Se probó a acceder a crear un Passenger sin estar logueado y posteriormente con el customer 1 se probó ha hackear el id y el version con valores incorrectos (999). | Ninguno |
| Update | Se probó actualizar un passenger sin estar logueado , actualizar passengers de otros usuarios, actualizar a un passenger publicado, otro sin publicar y uno inexistente, todo esto con otros customer que no eran los que habían creado dicho Passenger. | Ninguno |
| Publish | Se probó publicar un passenger sin estar logueado , publicar passengers de otros usuarios, publicar a un passenger publicado, otro sin publicar y uno inexistente, todo esto con otros customer que no eran los que habían creado dicho Passenger. | Ninguno |
| Delete | Se probó eliminar un passenger sin estar logueado , eliminar passengers de otros usuarios, eliminar a un passenger publicado, otro sin publicar y uno inexistente, todo esto con otros customer que no eran los que habían creado dicho Passenger. | Ninguno |

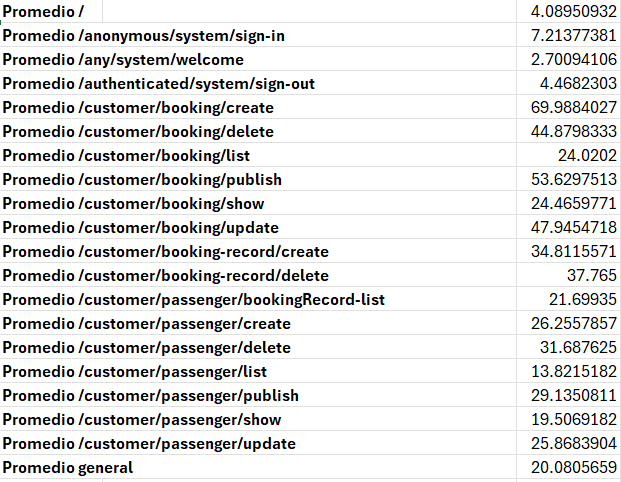
Después de probar todas las pruebas mencionadas anteriormente conseguimos llegar al 100% de coverage para todas las features:



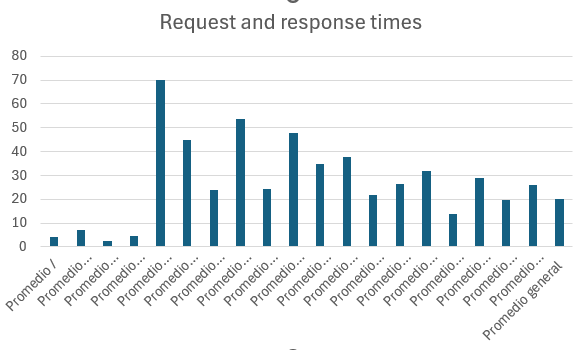
### **Rendimiento de testing:**

Se probaron de dos maneras distintas todas las pruebas mencionadas anteriormente. Primero se hicieron las mismas en el mismo ordenador de tal manera que primero se probó sin índices implementados y luego con índices implementados. Por último, se probaron todas las pruebas, sin índices implementados, en ordenadores distintos.

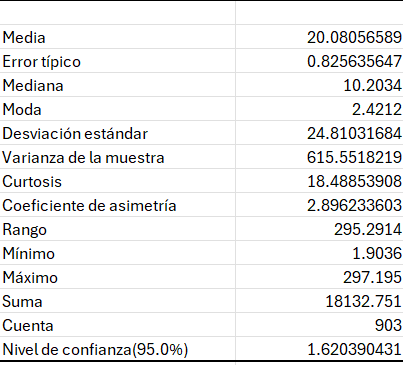
### **Resultados de la primera prueba (sin índices)**:

 Se muestran para este caso las siguientes medidas de tiempo (en milisegundos) para cada URI correspondiente a la petición que representa:

A continuación, representamos estos datos de manera gráfica:



Por tanto, de estos datos obtenemos la siguiente estadística:

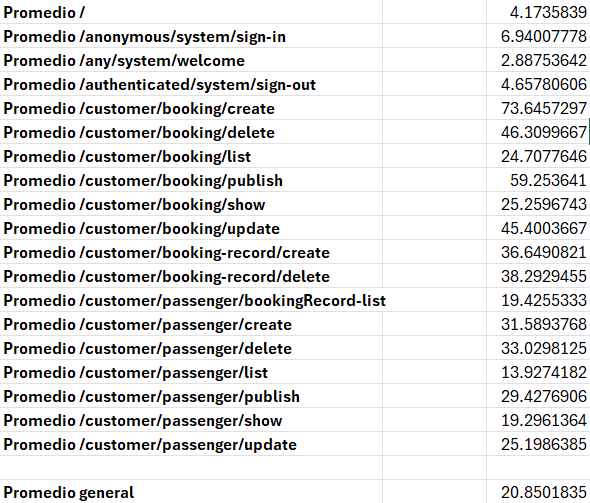


Como se puede observar, obtenemos un nivel de confianza de **1.620390431** y por tanto podemos establecer el siguiente intervalo de confianza:

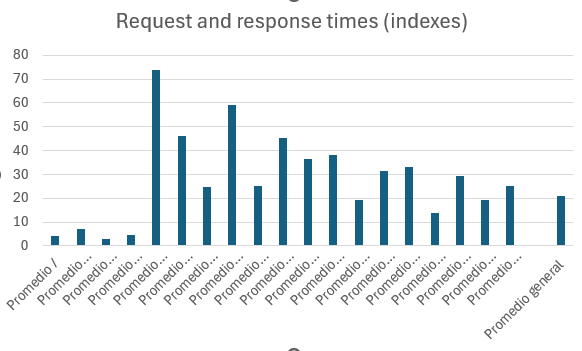


### **Resultados de la primer prueba (con índices):**

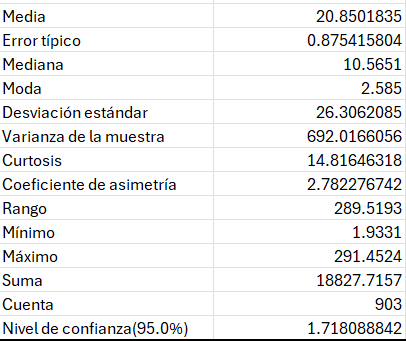
Se muestran para este caso las siguientes medidas de tiempo (en milisegundos) para cada URI correspondiente a la petición que representa:



A continuación, representamos estos datos de manera gráfica:



Por tanto, de estos datos obtenemos la siguiente estadística:



Como se puede observar, obtenemos un nivel de confianza de **1.718088842** y por tanto podemos establecer el siguiente intervalo de confianza:

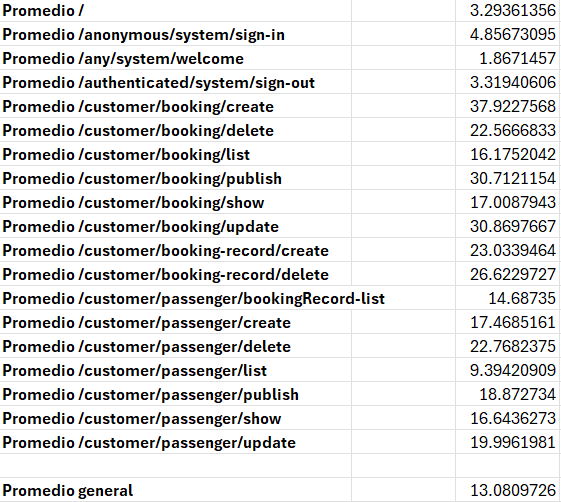


### **Resultados de la segunda prueba (sin índices en ordenador local):**

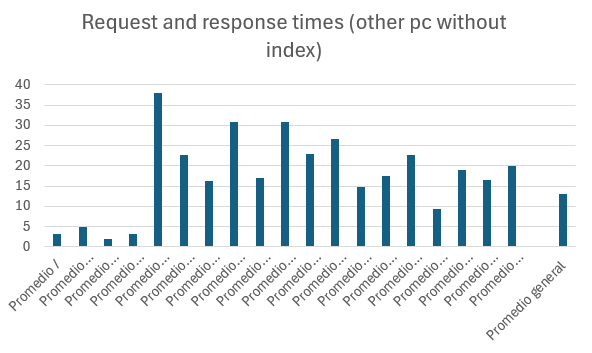
Estos son los mismos resultados que los del apartado 5.1.

### **Resultados de la segunda prueba (sin índices en otro ordenador):**

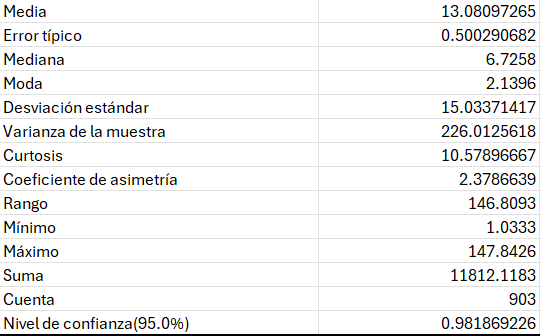
Se muestran para este caso las siguientes medidas de tiempo (en milisegundos) para cada URI correspondiente a la petición que representa:

****

A continuación, representamos estos datos de manera gráfica:

****

Por tanto, de estos datos obtenemos la siguiente estadística:

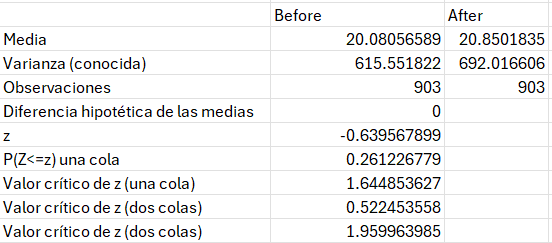
****

Como se puede observar, obtenemos un nivel de confianza de **0.981869226** y por tanto podemos establecer el siguiente intervalo de confianza:



### **Primera pruebaZ (mismo ordenador sin índices – con índices):**

Estos son los resultados obtenidos al contrastar ambas pruebas. Esta prueba recibe el nombre de Prueba Z.

****

Estos son los intervalos de confianza del antes (sin índices):

Imagen que contiene Texto

El contenido generado por IA puede ser incorrecto.

Y estos son los intervalos de confianza del después (con índices):



Podemos observar que el valor de P es mayor que 0.5, esto quiere decir que los cambios realizados no tienen un impacto en el rendimiento de mi aplicación. A nivel general, el sistema en cuanto a rendimiento se comporta en la misma distribución de tiempo, no podemos comparar los valores y por consecuente no se puede concluir nada porque no es significante.

### **Segunda pruebaZ (distinto ordenador sin índices)**

Estos son los resultados obtenidos al contrastar ambas pruebas:

**Tabla

El contenido generado por IA puede ser incorrecto.**

Estos son los intervalos de confianza del antes (ordenador 1 sin índices):

****

Y estos son los intervalos de confianza del después (ordenador 2 con índices):

****

Podemos observar que el valor de P es menor que 0.5, esto quiere decir que la diferencia en cuanto al rendimiento entre un ordenador y otro es suficientemente significativa y por tanto podemos comparar ambos valores. Se demuestra a través de la media que el ordenador 2 (After) es más potente que el ordenador 1 (Before).

### **Conclusiones**:

El testing es una fase clave del desarrollo, ya que asegura que el sistema cumpla con los requisitos, funcione correctamente y brinde buena experiencia al usuario. Sin embargo, no basta con validar funcionalidades y rendimiento, también es vital probar la seguridad para detectar vulnerabilidades y proteger los datos más vulnerables. Por ello, el testing debe entenderse como un proceso integral que combina calidad, eficiencia y protección, garantizando un producto robusto y confiable.

Personalmente esta fase de testing ha sido una experiencia muy enriquecedora, me ha hecho entender situaciones muy importantes que debo contemplar a la hora de abordar un proyecto de forma profesional. En ocasiones ha sido un completo desafío, pero gracias a los contenidos dados en clase, sesiones de *Follow Up* y por supuesto a la comunicación y ayuda de mis compañeros, hemos podido completar esta última fase del proyecto, de nuevo y como ha sido costumbre a lo largo del desarrollo de Acme-ANS, el trabajo en equipo ha sido la clave para lograr completar todos los requisitos esperados.

### **Bibliografía**:

Intencionalmente en blanco.