Universidad de Sevilla  
Escuela Técnica Superior de Ingeniería Informática

**Testing Report**



Grado en Ingeniería Informática – Ingeniería del Software  
Diseño y Pruebas 2.

Curso 2024 – 2025

|  |  |
| --- | --- |
| **Fecha** | **Versión** |
| 22/05/2025 | v1.0.0 |

|  |  |  |
| --- | --- | --- |
| **Grupo de prácticas: C1.005** | | |
| **Autores por orden alfabético** | **Rol** | **Descripción del rol** |
| Artero Bellido Manuel – manartbel@alum.us.es | Developer | Persona encargada de desarrollar el código. |
| Calderón Rodríguez, Manuel María -mancalrod@alum.us.es | Operador | Encargado de las tareas de campo, de las instalaciones y del mantenimiento de los sistemas de la empresa. |
| González Benito, Claudio – clagonben@alum.us.es | Project Manager | Persona encargada de tomar decisiones de diseño y vigilar el correcto desarrollo |
| Márquez Gutiérrez, José Manuel – josmargut@alum.us.es | Tester | Persona encargada de realizar pruebas sobre el código. |
| Ramos Vargas, Alba – albramvar1@alum.us.es | Developer | Persona encargada de desarrollar el código. |

**Control de Versiones**

|  |  |  |
| --- | --- | --- |
| **Fecha** | **Versión** | **Descripción** |
| 22/05/2025 | v1.0.0 | Desarrollo de la primera versión. |
|  |  |  |
|  |  |  |
|  |  |  |

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

Índice de contenido

[**1.** **Resumen ejecutivo** 2](#_Toc198917376)

[**2.** **Introducción** 3](#_Toc198917377)

[**3.** **Testeo funcional** 4](#_Toc198917378)

[**4.** **Reporte sobre las mutaciones** 13](#_Toc198917379)

[**5.** **Cobertura de las operaciones** 14](#_Toc198917380)

[**6.** **Testeo de rendimiento** 16](#_Toc198917381)

[**7.** **Conclusiones** 19](#_Toc198917382)

[**8.** **Bibliografía** 20](#_Toc198917383)

# **Resumen ejecutivo**

Este documento consiste en el informe necesario para cumplir con el requisito individual del estudiante 2 del reporte de pruebas funcionales y de rendimiento del estudiante 2 sobre sus entidades de cliente, de manera que consten los bugs encontrados en las pruebas, que tipo de pruebas se han realizado sobre las distintas operaciones y como de distinto es el rendimiento de la aplicación en distintos ordenadores.

El principal objetivo de esta entrega es probar de manera formal el funcionamiento correcto de nuestra aplicación, grabando para ello las pruebas necesarias sobre distintas operaciones legales e ilegales en las funcionalidades implementadas en la anterior entrega.

# **Introducción**

Este documento presenta dos capítulos esenciales: pruebas funcionales, con un listado de casos de prueba organizados por característica y su efectividad en detectar errores, y pruebas de rendimiento, con gráficos e intervalos de confianza del 95% para el tiempo de respuesta en dos computadoras, además de un análisis para determinar cuál es más potente. Estos capítulos son cruciales para evaluar la calidad y el rendimiento del software.

# **Testeo funcional**

En esta sección se detallarán todas las pruebas que hemos realizado para testear el funcionamiento de nuestro software. Se explicará qué es lo que se ha hecho, el resultado esperado, el resultado real (que hace referencia al resultado de la prueba corregido después de encontrar bugs que hicieran que el código estuviera mal) y, por último, la columna de bugs encontrados. Si ha habido bugs, se ha arreglado la prueba, por lo que el resultado real ya no los contiene.

Descripción de la prueba: Se detallará cada prueba realizada, incluyendo los pasos específicos llevados a cabo.

Resultado esperado: Se especificará cuál era el comportamiento o el resultado que se esperaba obtener con la prueba.

Resultado real: Se presentará el resultado obtenido tras la ejecución de la prueba, ya corregido después de encontrar y solucionar cualquier bug que afectará el código.

Bugs encontrados: Se listarán los bugs detectados durante la prueba y las correcciones realizadas. Una vez solucionados los bugs, se actualizará el resultado real para reflejar el funcionamiento correcto del software.

**List**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Entidad** | **Descripción** | **Resultado Esperado** | **Resultado Real** | **Bugs detectados** |
| **Booking safe** | Verificar que un cliente puede listar sus reservas. | El sistema deberá mostrar las reservas asociados al cliente | El sistema ha mostrado las reservas asociadas al cliente | Ninguno |
| **Booking hack** | Verificar que un rol no cliente no puede listar las reservas de los clientes | El sistema no debe mostrar las reservas, usando un rol no cliente, dando un error de pánico | El sistema ha dado un error 500 de no autorizado cuando el rol no es cliente. | Ninguno |
| **Passenger safe** | Comprobar que un cliente puede listar los pasajeros. | El sistema deberá mostrar sus pasajeros y los públicos | El sistema ha mostrado todos los pasajeros del cliente y los públicos | Ninguno |
| **Passenger hack** | Comprobar que un rol no cliente no pueda acceder a los pasajeros y que un cliente no pueda acceder a los pasajeros de otro cliente. | El sistema no debe mostrar los pasajeros siendo un rol no cliente, intentando hacerlo sobre los pasajeros de otro cliente que no estén publicados, dando un 500 como resultado. | El sistema ha dado un error 500 no autorizado cuando el rol no es cliente. | NInguno |
| **BookingPassenger safe** | Comprobar que un cliente puede listar los reservas-pasajero | El sistema deberá mostrar sus reservas-pasajero y los públicos | El sistema ha mostrado todos los reservas-pasajeros del cliente y los públicos | Ninguno |
| **BookingPassenger hack** | Comprobar que un rol no cliente no pueda acceder a los reservas-pasajero y que un cliente no pueda acceder a los pasajeros de otro cliente. | El sistema no debe mostrar los reservas-pasajero siendo un rol no cliente, intentando hacerlo sobre los pasajeros de otro cliente que no estén publicados, dando un 500 como resultado. | El sistema ha dado un error 500 no autorizado cuando el rol no es cliente. | Ninguno |

**Show**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Entidad** | **Descripción** | **Resultado Esperado** | **Resultado Real** | **Bugs detectados** |
| **Booking safe** | Verificar que un cliente puede ver toda la información de sus reservas. | El sistema deberá mostrar la información de las reservas asociadas al cliente. | El sistema ha mostrado la información de las reservas asociadas al cliente. | Ninguno |
| **Booking hack** | Verificar que un rol no cliente no puede ver la información de las reservas, que otro cliente ver la información de una reserva de otro cliente, ni un cliente intentar mostrar información de una reserva no existente. | El sistema no debe mostrar la información de reservas, usando un rol no cliente, ni mostrar información de reservas de otro cliente, ni autorizar el intento de mostrar información de reservas no existentes, dando un error de pánico. | El sistema ha dado un error 500 de no autorizado usando roles no cliente, usando reservas de otro cliente y usando reservas no existentes. | Ninguno |
| **Passenger safe** | Verificar que un cliente puede ver toda la información de sus pasajeros y los públicos. | El sistema deberá mostrar la información de sus pasajeros y los públicos. | El sistema ha mostrado la información de sus pasajeros y los públicos. | Ninguno |
| **Passenger hack** | Verificar que un rol no cliente no puede ver la información de los pasajeros, que otro cliente vea la información de un pasajero de otro cliente, ni un cliente intentar mostrar información de un pasajero no existente. | El sistema no debe mostrar la información de pasajeros, usando un rol no cliente, ni mostrar información de pasajeros de otro cliente, ni autorizar el intento de mostrar información de pasajeros no existentes, dando un error de pánico. | El sistema ha dado un error 500 de no autorizado en cada rol El sistema ha dado un error 500 de no autorizado usando roles no cliente, usando pasajeros de otro cliente y usando pasajeros no existentes. | Ninguno |
| **Passenger safe** | Verificar que un cliente puede ver toda la información de las reservas-pasajeros de sus reservas. | El sistema deberá mostrar la información de sus reservas-pasajero y los públicos. | El sistema ha mostrado la información de sus reservas-pasajeros y los públicos. | Ninguno |
| **Passenger hack** | Verificar que un rol no cliente no puede ver la información de los reservas-pasajeros, que otro cliente vea la información de un reservas-pasajero de otro cliente, ni un cliente intentar mostrar información de un reservas-pasajero no existente. | El sistema no debe mostrar la información de reservas-pasajeros, usando un rol no cliente, ni mostrar información de reservas-pasajeros de otro cliente, ni autorizar el intento de mostrar información de reservas-pasajeros no existentes, dando un error de pánico. | El sistema ha dado un error 500 de no autorizado en cada rol El sistema ha dado un error 500 de no autorizado usando roles no cliente, usando reservas- reservas-pasajeros de otro cliente y usando reservas-pasajeros no existentes. | Ninguno |

**Create**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Entidad** | **Descripción** | **Resultado Esperado** | **Resultado Real** | **Bugs detectados** |
| **Booking safe** | Verificar que un cliente puede crear reservas de forma correcta. | El sistema deberá permitir crear reservas válidas. Y rechazar aquellos que no cumplan con las restricciones. | El sistema ha dejado crear reservas con datos válidos y ha rechazado la creación de reservas con campos nulos cuando debía haber un mínimo o cuando el tipo de campo era incorrecto. | Ninguno |
| **Booking hack** | Verificar que un rol no cliente no pueda acceder a la creación de reservas, que un cliente no pueda crear reservas a otros clientes y que no pueda asociar vuelos que no sean públicos no existan. | El sistema deberá dar un 500 al intentar acceder a la creación desde un rol no cliente, ha de permitir la creación siempre que se sea cliente ya que se hace sobre el cliente autenticado, es decir que no se puede hacer Get Hacking y que no pueda asociar vuelos que no sean públicos no existan. | El sistema ha dado un error 500 de no autorizado en los roles no cliente. Ha permitido la creación desde un rol cliente, ya que la url no es posible hacerle Get Hacking y a saltado un pánico al intentar acceder a vuelos que no pueda acceder. | Ninguno |
| **Passenger safe** | Verificar que un cliente puede crear pasajeros de reservas de forma correcta. | El sistema deberá permitir crear pasajeros válidos y rechazar aquellos datos que no cumplan con las restricciones de negocio. | El sistema ha permitido crear pasajeros válidos. Y ha rechazado la creación de pasajeros que incumplían reglas del del negocio | Ninguno |
| **Passenger hack** | Verificar que un rol no cliente no pueda acceder a la creación de pasajeros, que un cliente no pueda crear pasajeros a otros clientes. | El sistema debe de lanzar error de pánico si un rol no cliente intenta acceder a la URL, si se intenta crear un pasajero cuya reserva pertenece a otro cliente, si se intenta crear para una reserva publicada. | El sistema ha dado un 500 al intentar crear pasajero en un rol no cliente, al intentar hacerlo sobre un pasajero de una reserva de otro cliente y sobre una reserva publicada. | Ninguno |
| **BookingPassenger safe** | Verificar que un cliente puede crear pasajeros de reservas-pasajeros de forma correcta. | El sistema deberá permitir crear reservas-pasajeros válidos y rechazar aquellos datos que no cumplan con las restricciones de negocio. | El sistema ha permitido crear reservas-pasajeros válidos. Y ha rechazado la creación de reservas-pasajeros que incumplían reglas del del negocio | Ninguno |
| **BookingPassenger hack** | Verificar que un rol no cliente no pueda acceder a la creación de reservas-pasajeros, que un cliente no pueda crear reservas a otros clientes y que no pueda asociar reservas que no sean públicas o no existan o pasajeros que no sean públicos o no le pertenezcan. | El sistema deberá dar un 500 al intentar acceder a la creación desde un rol no cliente, ha de permitir la creación siempre que se sea cliente ya que se hace sobre el cliente autenticado, es decir que no se puede hacer Get Hacking y que no pueda asociar reservas que no sean públicas o no existan o pasajeros que no sean públicos o no le pertenezcan. | El sistema ha dado un error 500 de no autorizado en los roles no cliente. Ha permitido la creación desde un rol cliente, ya que la url no es posible hacerle Get Hacking y a saltado un pánico al intentar acceder a las reservas que no pueda acceder y los pasajeros que no pueda acceder. | Ninguno |

**Update**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Entidad** | **Descripción** | **Resultado Esperado** | **Resultado Real** | **Bugs detectados** |
| **Booking safe** | Verificar que un cliente puede editar sus reservas de reservas de forma correcta. | El sistema deberá permitir actualizar reservas válidas. Y deberá rechazar la edición de reservas con datos que incumplan las reglas de negocio. | El sistema ha dejado actualizar reservas con datos válidos. Y ha rechazado los de los datos inválidos. | Ninguno |
| **Booking hack** | Verificar que un rol no cliente no pueda acceder a la edición de reservas, que un cliente no pueda editar reservas a que no estén en modo borrador por un show previo y q que no pueda asociar vuelos que no sean públicos no existan. | El sistema deberá dar un 500 al intentar acceder a la edición desde un rol no cliente, o cuando se pone la url de update sin haber hecho un show antes y que no pueda asociar vuelos que no sean públicos no existan. | El sistema ha generado un 500 al intentar acceder a la edición desde un rol no cliente, o cuando se pone la url de update sin haber hecho un show antes y que no pueda asociar vuelos que no sean públicos no existan. | Ninguno |
| **Passenger safe** | Verificar que un cliente puede editar pasajeros de reservas de forma correcta. | El sistema deberá actualizar pasajeros válidos. Y rechazar la actualización de pasajeros que incumplan las reglas de negocio. | El sistema ha dejado actualizar pasajeros con datos válidos. Y ha rechazado los de los datos inválidos. | Ninguno |
| **Passenger hack** | Verificar que un rol no cliente no pueda acceder a la edición de pasajeros, que un cliente no pueda editar pasajeros que no estén en modo borrador por un show previo.. | El sistema debe de lanzar error de pánico si un rol no cliente intenta acceder a la URL y si se intenta editar un pasajero sin haberle hecho un show.. | El sistema ha dado un 500 al intentar editar pasajeros en un rol no cliente y al intentar hacerlo sobre un pasajero que no estuviera en un modo borrador por un show previo. | Ninguno |
| **BookingPassenger safe** | Verificar que un cliente puede editar pasajeros de reservas-pasajeros de forma correcta. | El sistema deberá permitir editar reservas-pasajeros válidos y rechazar aquellos datos que no cumplan con las restricciones de negocio. | El sistema ha permitido editar reservas-pasajeros válidos. Y ha rechazado la creación de reservas-pasajeros que incumplían reglas del del negocio | Ninguno |
| **BookingPassenger hack** | Verificar que un rol no cliente no pueda acceder a la edición de reservas-pasajeros, que un cliente no pueda editar reservas a otros clientes y que no pueda asociar reservas que no sean públicas o no existan o pasajeros que no sean públicos o no le pertenezcan. | El sistema deberá dar un 500 al intentar acceder a la edición desde un rol no cliente, ha de permitir la edición siempre que se sea cliente ya que se hace sobre el cliente autenticado, es decir que no se puede hacer Get Hacking y que no pueda asociar reservas que no sean públicas o no existan o pasajeros que no sean públicos o no le pertenezcan. | El sistema ha dado un error 500 de no autorizado en los roles no cliente. Ha permitido la edición desde un rol cliente, ya que la url no es posible hacerle Get Hacking y a saltado un pánico al intentar acceder a las reservas que no pueda acceder y los pasajeros que no pueda acceder. | Ninguno |

**Publish**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Entidad** | **Descripción** | **Resultado Esperado** | **Resultado Real** | **Bugs detectados** |
| **Booking safe** | Verificar que un cliente puede publicar sus reservas de reservas de forma correcta. Solo en caso de que sus reservas-pasajeros estén publicados. | El sistema deberá permitir publicar reservas válidas. Y deberá rechazar la publicación de reservas con datos que incumplan las reglas de negocio. | El sistema ha dejado publicar reservas con datos válidos. Y ha rechazado los de los datos inválidos. | Ninguno |
| **Booking hack** | Verificar que un rol no cliente no pueda acceder a la publicación de reservas, que un cliente no pueda publicar reservas a que no estén en modo borrador por un show previo y que no pueda asociar vuelos que no sean públicos no existan. | El sistema deberá dar un 500 al intentar acceder a la publicación desde un rol no cliente, o cuando se pone la url de publish sin haber hecho un show antes y que no pueda asociar vuelos que no sean públicos no existan. | El sistema ha generado un 500 al intentar acceder a la publicación desde un rol no cliente, o cuando se pone la url de publish sin haber hecho un show antes y que no pueda asociar vuelos que no sean públicos no existan. | Ninguno |
| **Passenger safe** | Verificar que un cliente puede publicar pasajeros de reservas de forma correcta. | El sistema deberá publicar pasajeros válidos. Y rechazar la publicación de pasajeros que incumplan las reglas de negocio. | El sistema ha dejado publicar pasajeros con datos válidos. Y ha rechazado los de los datos inválidos. | Ninguno |
| **Passenger hack** | Verificar que un rol no cliente no pueda acceder a la publicación de pasajeros, que un cliente no pueda publicar pasajeros que no estén en modo borrador por un show previo.. | El sistema debe de lanzar error de pánico si un rol no cliente intenta acceder a la URL y si se intenta publicar un pasajero sin haberle hecho un show.. | El sistema ha dado un 500 al intentar publicar pasajeros en un rol no cliente y al intentar hacerlo sobre un pasajero que no estuviera en un modo borrador por un show previo. | Ninguno |
| **BookingPassenger safe** | Verificar que un cliente puede publicar pasajeros de reservas-pasajeros de forma correcta. Siguiendo restricciones como que sus pasajeros estén publicados | El sistema deberá permitir publicar reservas-pasajeros válidos y rechazar aquellos datos que no cumplan con las restricciones de negocio. Solo permitiéndolo en caso de que el pasajero esté publicado. | El sistema ha permitido publicar reservas-pasajeros válidos. Y ha rechazado la creación de reservas-pasajeros que incumplían reglas del del negocio | Ninguno |
| **BookingPassenger hack** | Verificar que un rol no cliente no pueda acceder a la publicación de reservas-pasajeros, que un cliente no pueda publicar reservas a otros clientes y que no pueda asociar reservas que no sean públicas o no existan o pasajeros que no sean públicos o no le pertenezcan. | El sistema deberá dar un 500 al intentar acceder a la publicación desde un rol no cliente, ha de permitir la publicación siempre que se sea cliente ya que se hace sobre el cliente autenticado, es decir que no se puede hacer Get Hacking y que no pueda asociar reservas que no sean públicas o no existan o pasajeros que no sean públicos o no le pertenezcan. | El sistema ha dado un error 500 de no autorizado en los roles no cliente. Ha permitido la publicación desde un rol cliente, ya que la url no es posible hacerle Get Hacking y a saltado un pánico al intentar acceder a las reservas que no pueda acceder y los pasajeros que no pueda acceder. | Ninguno |

**Delete**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Entidad** | **Descripción** | **Resultado Esperado** | **Resultado Real** | **Bugs detectados** |
| **BookingPassenger safe** | Verificar que un cliente puede eliminar reservas-pasajeros de reservas de forma correcta. | El sistema deberá eliminar reservas-pasajeros válidos. Y rechazar la eliminación de reservas-pasajeros que incumplan las reglas de negocio. | El sistema ha dejado eliminar reservas-pasajeros con datos válidos. Y ha rechazado los de los datos inválidos. | Ninguno |
| **BookingPassenger hack** | Verificar que un rol no cliente no pueda acceder a la eliminación de reservas-pasajeros, que un cliente no pueda eliminar reservas-pasajeros que no estén en modo borrador por un show previo. | El sistema debe de lanzar error de pánico si un rol no cliente intenta acceder a la URL y si se intenta eliminar un reservas-pasajero sin haberle hecho un show. . | El sistema ha dado un 500 al intentar eliminar reservas-pasajeros en un rol no cliente y al intentar hacerlo sobre un reservas-pasajero que no estuviera en un modo borrador por un show previo. | Ninguno |

**ListByBooking**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Entidad** | **Descripción** | **Resultado Esperado** | **Resultado Real** | **Bugs detectados** |
| **Passenger safe** | Comprobar que un cliente puede listar los pasajeros de sus reservas. | El sistema deberá mostrar los pasajeros asociados a una reserva del cliente. | El sistema ha mostrado todos los pasajeros asociados a las reservas de los de un cliente. | Ninguno |
| **Passenger hack** | Comprobar que un rol no cliente no pueda acceder a los pasajeros de las reservas y que un cliente no pueda acceder a los pasajeros de una reserva de otro cliente y de una reserva que no existe. | El sistema no debe mostrar los pasajeros siendo un rol no cliente, intentando hacerlo sobre los pasajeros de una reserva de otro cliente y sobre una reserva inexistente, dando un 500 como resultado. | El sistema ha detectado cuando se intentaba acceder desde un rol no cliente y desde la reserva de otro cliente, pero daba un 500 de null pointer exception cuando la reserva era inexistente. | Saltaba **null pointer exception** cuando se **modificaba** la **url** usando un **id** de **reserva** **no existente**, pues no se comprobaba si existía antes de intentar comprobar que no fuera una reserva de otro cliente. Se solucionó y ahora el 500 es Not Authorised. |

# **Cobertura de las operaciones**

En este apartado se muestra el porcentaje de cobertura obtenido con las pruebas formales, confirmando que supera el umbral mínimo del 90 % antes de iniciar las pruebas de rendimiento. De este modo, queda documentado que todas las operaciones críticas han sido validadas y cumple el requisito de cobertura establecido.

1. Cobertura de Booking

A screenshot of a graph

AI-generated content may be incorrect.

1. Cobertura de BookingPassenger

A screenshot of a graph

AI-generated content may be incorrect.

1. Cobertura de Passenger

A screenshot of a graph

AI-generated content may be incorrect.

En las capturas se ve cómo el replay de pruebas cubre prácticamente todo el módulo “estudiante 2”, reduciendo al mínimo la posibilidad de errores ocultos.

# **Testeo de rendimiento**

En este capítulo, se evaluará el rendimiento de nuestro proyecto mediante pruebas detalladas que analizan el tiempo de respuesta al atender las solicitudes en nuestras pruebas funcionales. Utilizaremos dos archivos de testeo, uno como replay de las pruebas hechas un portátil y el segundo como replay de las pruebas en un pc con unas características algo mejores que el primero.

Para realizar el análisis de estas pruebas, presentaremos gráficos que ilustran los resultados obtenidos. Además, calcularemos intervalos de confianza del 95% para los tiempos de respuesta medidos y realizaremos un contraste de hipótesis mediante un test z para pruebas de dos medias, también con un intervalo de confianza del 95%, para determinar la diferencia de hacer replay sobre pruebas en un ordenador con respecto a otro.

**Gráficas de pruebas en primer ordenador:**

**Gráficas de pruebas en segundo ordenador:**

**Comparación de resultados:**

|  |  |  |  |
| --- | --- | --- | --- |
| Prueba z para medias de dos muestras | | |  |
|  |  |  |  |
|  | *Before* | *After* |  |
| Media | 5,647044 | 5,501365 |  |
| Varianza (conocida) | 78,1272 | 62,23939 |  |
| Observaciones | 834 | 834 |  |
| Diferencia hipotética de las medias | 0 |  |  |
| z | 0,355099 |  |  |
| P(Z<=z) una cola | 0,361258 |  |  |
| Valor crítico de z (una cola) | 1,644854 |  |  |
| Valor crítico de z (dos colas) | 0,722516 |  |  |
| Valor crítico de z (dos colas) | 1,959964 |  |  |

En nuestras pruebas de rendimiento, hemos obtenido un valor crítico de z de **0,722516** para un nivel de significancia (α) de **0.05**. Este valor se utiliza para determinar si las diferencias en los tiempos de respuesta entre las dos pruebas son estadísticamente significativas.

De acuerdo con nuestra metodología, si el p-valor se encuentra en el intervalo (α, 1.00], esto indica que los cambios no resultaron en mejoras relevantes; es decir, aunque los tiempos de muestra sean diferentes, globalmente son equivalentes. Dado que nuestro valor crítico de z está en este intervalo, concluimos que no hay una mejora significativa en el rendimiento al comparar los tiempos de respuesta de las pruebas. En resumen, las diferencias observadas no son lo suficientemente grandes como para considerarse estadísticamente relevantes con el nivel de confianza del 95%. Por lo que tampoco tiene sentido comparar las medias de tiempo de ejecución de las baterías de pruebas ejecutadas en un ordenador con respecto al otro.

# **Conclusiones**

En este proceso de testing hemos aprendido que definir con precisión las pruebas para cada funcionalidad solicitada por el cliente es fundamental para alinear resultados esperados y reales. Un entorno de pruebas aislado, por su parte, facilita la reproducción de errores y acelera su resolución en caso de haber encontrado bugs haciendo el testing. Asimismo, el análisis de métricas de rendimiento nos permite evaluar requisitos no funcionales, detectar ineficiencias en el código y determinar si su origen es de hardware o software para proceder, en caso de ser razón de software, con la mejora en la eficiencia del código.

# **Bibliografía**

Intencionalmente en blanco.