UNIVERSIDAD DE SEVILLA

ESCUELA TECNICA SUPERIOR DE INGENIERIA INFORMATICA

DISEÑO Y PRUEBAS II

CURSO 2024 – 2025

TESTING REPORT

STUDENT #2

GRUPO: C2.003

ENRIQUE ANDA HERNANDEZ

[enrandher@alum.us.es](mailto:enrandher@alum.us.es)

UVUS: HQY6849

REPOSITORIO: https://github.com/manumnzz/DP2-2425-C2-003

FECHA: 03/07/2025

Contenido

[1. Resumen ejecutivo 3](#_Toc202465540)

[2. Introduccion 3](#_Toc202465541)

[3. Contenido 3](#_Toc202465542)

[3.1. Testing funcional 3](#_Toc202465543)

[3.1.1. Funcionalidad para Customer sobre Booking 4](#_Toc202465544)

[3.1.2. Funcionalidad para los Customer sobre Passenger 5](#_Toc202465545)

[3.1.3. Funcionalidad para los Customers sobre los BookingRecord(relación intermedia entre booking y passenger) 6](#_Toc202465546)

[3.2. Testing de rendimiento 7](#_Toc202465547)

[3.2.1. Testing sin índices 8](#_Toc202465548)

[3.2.2. Testing con índices 8](#_Toc202465549)

[3.2.3. comparación entre los dos testing 9](#_Toc202465550)

[4. Conclusión 10](#_Toc202465551)

# Resumen ejecutivo

Este documento recoge de forma detallada los resultados obtenidos en las pruebas funcionales y de rendimiento aplicadas al proyecto. En la sección dedicada a las pruebas funcionales, se explican los distintos casos de prueba diseñados, cada uno acompañado de una descripción precisa y una valoración de su capacidad para identificar errores. Por otro lado, el apartado de rendimiento incluye gráficos ilustrativos y un análisis del tiempo de respuesta del sistema, considerando un intervalo de confianza del 95%, tanto con el uso de índices como sin ellos. Asimismo, se presenta una comparación estadística que permite evaluar, con el mismo nivel de confianza, si la incorporación de índices repercute significativamente en el desempeño del sistema. El informe sigue un enfoque meticuloso y estructurado, garantizando la fiabilidad de los resultados y su utilidad para valorar el funcionamiento general del proyecto.

# Introduccion

Este documento resume los procedimientos y resultados de las pruebas funcionales y de rendimiento realizadas en el proyecto. En primer lugar, se presentan los casos de prueba diseñados para verificar distintas funcionalidades del sistema, incluyendo una breve descripción y su capacidad para detectar errores. Esta sección permite evaluar el comportamiento del sistema en distintos escenarios y facilitar la corrección de fallos.

A continuación, se analiza el rendimiento del sistema mediante tiempos de respuesta obtenidos con y sin índices, acompañados de gráficos y un intervalo de confianza del 95%. También se realiza un contraste de hipótesis para determinar el impacto estadístico del uso de índices. El informe sigue un enfoque sistemático, apoyado en herramientas como Eclipse, que asegura la fiabilidad de los resultados.

# Contenido

## Testing funcional

En esta sección se describen los casos de prueba implementados para cada funcionalidad del sistema, incluyendo pruebas positivas, negativas y de hacking. Además, se analiza la efectividad de estas pruebas en la detección de errores, apoyándonos en las métricas de cobertura proporcionadas por Eclipse, que indican el porcentaje de instrucciones ejecutadas durante la ejecución de las pruebas.

### Funcionalidad para Customer sobre Booking

En primer lugar vamos a ver las pruebas establecidas para las funcionalidades de los customers sobre los bookings

#### List

Esta funcionalidad consiste en que los customers puedan ver su lista de bookings asociados. Para realizar las pruebas positivas nos hemos logueado como customer y hemos listado sus bookings.

Para realizar las pruebas de hacking, sin habernos logueado con ningún customer intentamos listar los bookings y obtenemos un error.

#### Show

Esta funcionalidad consiste en que el customer pueda ver los atributos de un booking.

Para realizar las pruebas positivas nos hemos logueado como un customer y hemos listado los bookings y hemos clickado en uno de ellos, mostrando los atributos.

Para las pruebas de hacking hemos intentado mostrar los atributos sin habernos logueado con ningún customer y también hemos intentado ver los atributos de un customer que no era con el que nos habíamos logueado. En los dos casos hemos obtenido un error.

#### Create

Esta funcionalidad consiste en que el customer pueda crear nuevos bookings.

Para realizar las pruebas positivas y negativas hemos listado los bookings del customer y hemos pulsado en el botón de Add Booking y la aplicación ha mostrado los campos necesarios para crear el booking. Hemos ido rellenándolos, comprobando las excepciones de cada campo de texto y finalmente creando un booking con los datos correctos.

Para realizar las pruebas de hacking, hemos intentado crear un booking sin habernos logueado con ningún customer y ha dado el correspondiente error.

#### Update

Esta funcionalidad consiste en que el customer pueda modificar la información de los customer ya creados pero que no estén publicados.

Para realizar las pruebas positivas y negativas, hemos listado los bookings de uno de los customer y hemos clickado en un booking sin publicar. Despues hemos modificado los campos uno a uno probando las diferentes excepciones que pueden darse en los diferentes campos de texto y finalmente hemos actualizado la información con datos correctos.

Para realizar las pruebas de hacking, hemos intentado realizar el update sin habernos logueado como customer y también hemos probado ha hacer el update a un booking ya publicado, dándonos los correspondientes errores.

#### Publish

Esta funcionalidad consiste en que el customer pueda publicar un booking.

Para realizar las pruebas buenas y las pruebas malas, hemos intentado publicar un booking que no tenía asociada los últimos dígitos de una tarjeta de crédito dando la correspondiente excepción. La otra prueba mala que hemos realizado es intentar publicar un booking con passengers sin publicar y sin passengers asociados y ha dado la correspondiente excepción. Para la prueba buena hemos listado los bookings, hemos entrado en uno de ellos, el cual sus passengers estaban publicados y hemos pulsado el botón publicar.

### Funcionalidad para los Customer sobre Passenger

En segundo lugar, vamos a ver las pruebas establecidas para las funcionalidades de los customers sobre los passengers.

#### List

Esta funcionalidad consiste en que el customer pueda listar los passengers.

Esta funcionalidad el customer puede realizarla por dos vías. Por un lado, puede listar los passengers asociados a un booking y por el otro puede listar directamente todos los passengers que están asociados a sus bookings.

Para realizar las pruebas positivas, nos hemos logueado con un customer y hemos listado los Passenger directamente y también introduciéndonos en un booking y listando los asociados a este.

Para las pruebas de hacking, hemos intentado listar los passengers sin habernos logueado con ningún customer, y también nos hemos logueado con un customer y hemos intentado listar los passengers asociados a otro customer. Hemos obtenido los correspondientes errores.

#### Show

Esta funcionalidad consiste en que el customer pueda ver los atributos de un Passenger.

Para las pruebas positivas nos hemos logueado con un customer, hemos listado los pasengers y hemos clickado en uno de ellos.

Para las pruebas de hacking hemos intentado mostrar los atributos de un passengers sin habernos logueado con ningún customer y también nos hemos logueado con un customer y hemos intentado ver los atributos de un Passenger asociado a otro customer. Hemos obtenido los correspondientes errores.

#### Create

Esta funcionalidad consiste en que el customer pueda crear nuevos passengers.

Para las pruebas positivas y negativas, nos hemos logueado con uno de los customer, hemos entrado en uno de los bookings que no estaba publicado, hemos listado los Passenger y hemos clickado en el botón de add Passenger. Una vez se han listado los campos de texto de los correspondientes atributos, hemos ido introduciendo los datos teniendo en cuenta las excepciones de los campos y hemos terminado pulsando en el boton créate.

Para las pruebas de hacking, primero hemos intentado realizar el créate sin habernos logueado como customer. Para la segunda prueba hemos intentado crear un Passenger en un booking que ya estaba publicado. Hemos obtenido los correspondientes errores para estas pruebas.

#### Update

Esta funcionalidad consiste en que el customer pueda actualizar la información de los Passengers que tiene asociados.

Para las pruebas positivas y negativas nos hemos logueado como customer y hemos intentado actualizar el Passenger comprobando las excepciones de cada campo de texto, terminando al pulsar el botón de update con los datos correctos.

Para las pruebas de hacking hemos intentado realizar el update sin habernos logueado con ningún customer, hemos intentado también realizar el update de un Passenger ya publicado y de un Passenger que no pertenecía al customer en el que estábamos logueados. En todos los casos hemos obtenido el error correspondiente.

#### Publish

Esta funcionalidad consiste en que el customer pueda publicar un Passenger.

Para realizar las pruebas positivas nos hemos logueado con un customer y hemos listado los bookings, introduciéndonos en un booking sin publicar con passengers no publicados asociados y hemos entrado en uno de ellos y hemos pulsado el botón de publicar.

Para las pruebas de hacking hemos intentado realizar el publish de un Passenger sin habernos logueado con ningún customer. También, hemos intentado publicar un Passenger ya publicado y un Passenger que pertenecía a otro customer. Hemos obtenido los correspondientes errores.

### Funcionalidad para los Customers sobre los BookingRecord(relación intermedia entre booking y passenger)

Para terminar las pruebas, vamos a realizar las correspondientes sobre los bookingRecord para poder asociar pasajeros a las reservas.

#### Create

Esta funcionalidad permite crear un BookingRecord, es decir, una relación entre un booking y un Passenger.

Las pruebas positivas y negativas las relaizamos logueandonos con un customer e intentando tanto crear una relación ya existente, lo cual nos lanza la excepción correspondiente, como realizarla con los valores correctos.

Para realizar las pruebas de hacking, intentamos crear el bookingrecord sin estar logueado con ningún customer. Lanza el error correspondiente.

Puntualización: los booking records también se pueden crear un Passenger desde un booking directamente desde este.

#### List

Esta funcionalidad permite a un customer listar sus bookingRecord asociados.

Para la prueba positiva nos logueamos con un customer y listamos los bookingrecords.

Para la prueba de hacking intentamos listar los bookingrecords sin estar logueado con ningún customer. Lanza el error correspondiente.

#### Show

Esta funcionalidad permite al customer ver el identificador del booking y del Passenger que forman la relación.

Para la prueba positiva nos logueamos con un customer, listamos sus bookingrecords y entramos en uno de ellos.

Para la prueba de hacking, intentamos mostrar los atributos de un bookingrecord sin habernos logueado con ningún customer. Obteniendo el error correspondiente.

## Testing de rendimiento

A continuación, nos centramos en el análisis del tiempo de respuesta del sistema al procesar solicitudes durante las pruebas funcionales. Se realizan comparaciones entre la ejecución en nuestro equipo portátil, tanto sin como con índices, con el objetivo de evaluar si la inclusión de índices ha mejorado el rendimiento.

### Testing sin índices

Tabla

El contenido generado por IA puede ser incorrecto.Siguiendo los pasos de las diapositivas hemos obtenido la siguiente informacion

Gráfico, Gráfico de dispersión

El contenido generado por IA puede ser incorrecto.

### Testing con índices

Gráfico

El contenido generado por IA puede ser incorrecto.

Tabla

El contenido generado por IA puede ser incorrecto.

### comparación entre los dos testing

Interfaz de usuario gráfica, Aplicación, Tabla, Excel

El contenido generado por IA puede ser incorrecto.

Tabla

El contenido generado por IA puede ser incorrecto.

Podemos observar que el valor critico de z para dos colas esta en el intervalo [0, Alpha) pero esta muy cerca de Alpha lo que indica que tampoco es demasiado significativa la diferencia de tiempos medios de las dos ejecuciones

# Conclusión

Este informe presenta los resultados obtenidos en las pruebas funcionales y de rendimiento realizadas sobre el proyecto. En el apartado funcional, se describen distintos casos de prueba diseñados para validar el comportamiento del sistema, evaluando su capacidad para detectar errores.

Las pruebas de rendimiento se han centrado en analizar el tiempo de respuesta del sistema al procesar solicitudes en dos equipos distintos. Se han utilizado gráficos e intervalos de confianza del 95% para asegurar la fiabilidad de los resultados.

En cuanto a las pruebas funcionales, se han implementado casos positivos, negativos y de hacking para cada funcionalidad. La herramienta Eclipse ha permitido medir la cobertura de código, alcanzando una media del 98%. Aunque algunos archivos han llegado al 100%. Aun así, la alta cobertura general indica que el sistema ha sido ampliamente validado y muestra un buen nivel de robustez frente a fallos y ataques.

Respecto al rendimiento, se compararon los tiempos de respuesta con y sin índices. Los resultados mostraron que el uso de índices no generó mejoras significativas, por lo que su impacto en el rendimiento fue limitado.

En conjunto, este informe concluye que el sistema es funcionalmente sólido y que su rendimiento es adecuado, proporcionando una base firme para futuras mejoras.