Universidad de Sevilla

Escuela Técnica Superior de Ingeniería Informática

**Chartering Report**

****

**Grado en Ingeniería Informática – Ingeniería del Software  
 Diseño y Pruebas II**

**Curso 2024 – 2025**

|  |  |  |
| --- | --- | --- |
| **Grupo: C1.052** | |  |
| **Autores por orden alfabético** | **Roles** | **Correo Corporativo** |
| Miguel Yan García Azuara | Tester, Developer | miggarazu@alum.us.es |

Fecha: 26/05/2025

**Índice**

[1. Functional Testing: 2](#_Toc192589823)

[1.2. Booking Feature: 2](#_Toc192589824)

[1.3. BookingRecord Feature: 10](#_Toc192589824)

[1.4. Passenger Feature: 12](#_Toc192589824)

[2. Performance Testing: 20](#_Toc192589824)

[2.1 Before indices: 20](#_Toc192589824)

[2.2 After indices: 22](#_Toc192589824)

[2.3 Hypotesis: 23](#_Toc192589824)

[2.4 Comparative and conclusion: 25](#_Toc192589824)

Este documento consiste en un reporte del testing funcional y de rendimiento al que se ha sometido las features correspondientes al Student 2.

El documento está dividido en 2 partes, por un lado, el análisis correspondiente al testing funcional y por otro el del testing de rendimiento. En el testing funcional se describen los casos de pruebas (tanto positivos como negativos) utilizados en cada feature que dieron lugar a los archivos .safe, así como los intentos de hacking que derivaron en los archivos .hack. Por otro lado, en el testing de rendimiento, se analizarán las diferencias de rendimiento utilizando el intervalo de confianza y el p-value obtenido con el Z-test lo que nos permitirá comparar el rendimiento de las features antes y después de la utilización de índices.

**FUNCTIONAL TESTING:**

A graph with green squares

AI-generated content may be incorrect.

Booking List Feature:

Para realizar el testing de la feature de listado de bookings, me he logeado como customer1 y he accedido al listado de las bookings. Con esto, he logrado un 100% de cobertura ya que el método authorise del listado de bookings era bastante sencillo no tenía más restricciones que comprobar que el cliente está registrado como Customer.

Para comprobar el hacking, he iniciado sesión con un realm distinto de Customer y he comprobado que se desautoriza la url del listado de bookings.

A screenshot of a computer program

AI-generated content may be incorrect.

Booking Show Feature:

Para realizar el testing de la feature de mostrar los detalles de las bookings, me he logueado como customer1, he entrado en el listado de bookings y he accedido a una booking cualquiera con draftMode = false. Además, he accedido a la booking con draftMode en true para poder testear el código relativo al else del bloque if(!booking.getDraftMode()) en específico el bloque de if(!flightIsValid). Esta parte del código la realicé para manejar el caso en el que un cliente creara un booking y le asignara un flight pero no publicara la booking. Si posteriormente intentara acceder a esa booking y el vuelo ya hubiera pasado, se producía un error ya que el flight asignado inicialmente no estaba disponible en el navegable ya que ahora es anterior a la fecha del booking. Este error se producía en el selected del choices el cual era nulo ya que se filtraban los vuelos con fecha de salida futura y como el vuelo anterior ya había pasado no lo encontraba en la lista produciéndose un null.

Por último, cree un booking con vuelo futuro pero no lo publiqué, con esto conseguí probar la sección de código else del if(!flightIsValid) la cual se utiliza para mostrar las choices del desplegable correctamente al igual que el selected.

Con esto y junto con los tests del hacking, he logrado un 97,2% de cobertura. Esto se debe a las líneas assert booking != null la cual nunca se cumple en estos casos de prueba y al filtrado de los flights.

Para testear el hacking, he iniciado sesión con un realm distinto de Customer y he comprobado que se desautoriza la url del show de una booking cualquiera. Además, me he logueado como customer2 y he comprobado que no puedo acceder al show de una booking del customer1.

A screenshot of a computer program

AI-generated content may be incorrect.

A screenshot of a computer code

AI-generated content may be incorrect.

Booking Create Feature:

Para realizar el testing de la feature de crear las bookings, me he logueado como customer1, accedido al listado de bookings y posteriormente le ha dado a crear una booking.

En primer lugar, probé a crear una booking con todos los campos vacíos, y a continuación probé todos los campos uno a uno dejando el resto vacíos. En el desplegable de flight, probé los 2 disponibles, en locatorCode probé distintas longitudes tanto válidas como no válidas, en travelClass tanto economy como bussiness y en lastCardNibble distintas longitudes y valores.

Con estos casos de prueba y junto con el hacking he logrado un 97,8% apareciendo líneas en amarillo correspondientes a asegurarse de que el booking no sea null en el método unbind y filtrando los flight que no tienen ninguna leg. Esta última no se ejecuta totalmente ya que no hay ningún flight sin legs asociadas, pero en el que caso que lo hubiera, se filtraría.

Para comprobar el hacking, he iniciado sesión con un realm distinto de Customer y he comprobado que se desautoriza la url del create de bookings. Además, he probado POST hacking del atributo flight introduciendo un id negativo (-9 o -1, no lo recuerdo), id positivo cualquiera (12) y un id correspondiente a un flight no publicado (96).

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

A close-up of a text

AI-generated content may be incorrect.

Booking Update Feature:

Para realizar el testing de la feature de editar las bookings, me he logueado como customer1, accedido al listado de bookings y posteriormente a la booking con draftMode true.

En primer lugar, probé a actualizar la booking con todos los campos vacíos, y a continuación probé todos los campos uno a uno dejando el resto vacíos. En el desplegable de flight, probé los 2 disponibles, en locatorCode probé distintas longitudes tanto válidas como no válidas, en travelClass tanto economy como bussiness y en lastCardNibble distintas longitudes y valores.

Con estos casos de prueba y junto con el hacking he logrado un 99,6% apareciendo líneas en amarillo correspondientes a líneas del authorise que se deben a que la variable status se ha ido concatenando con && y para que aparezca en verde creo que deben probarse tanto como true y false lo cual es difícil al concatenarse varios boolean. Además, la línea que comprueba que la petición sea un post, no aparece en verde porque al hacer el update siempre se hace un post y por tanto el caso negativo no se puede probar.  
El filtrado de los flights sin legs asociadas también aparecen en amarillo ya que no hay ningún flight sin legs asociadas, pero en el que caso que lo hubiera, se filtraría.

Para testear el hacking, he iniciado sesión con un realm distinto de Customer y he comprobado que se desautoriza la url del update de una booking cualquiera. Además, me he logueado como customer2 y he comprobado que no puedo acceder al update de una booking del customer1. Por otro lado, estando registrado como customer 1, he comprobado que no puedo acceder a la url del update de una booking ya publicada del customer1. Además, he probado POST hacking del atributo flight introduciendo un id negativo (-9 o -1, no lo recuerdo), id positivo cualquiera (12) y un id correspondiente a un flight no publicado (96).

A close-up of a computer screen

AI-generated content may be incorrect.

A screenshot of a computer program

AI-generated content may be incorrect.

Booking Publish Feature:

Para realizar el testing de la feature de publicar las bookings, me he logueado como customer1, accedido al listado de bookings y posteriormente a la booking con draftMode true.

En primer lugar, probé a publicar la booking con todos los campos vacíos, y a continuación probé todos los campos uno a uno dejando el resto vacíos. En el desplegable de flight, probé los 2 disponibles, en locatorCode probé distintas longitudes tanto válidas como no válidas, en travelClass tanto economy como bussiness y en lastCardNibble distintas longitudes y valores. Además, probé a publicar la booking con los pasajeros publicados y sin publicar para cubrir el mensaje del validate.

Con estos casos de prueba y junto con el hacking he logrado un 98,4% apareciendo algunas líneas en amarillo en el authorise ya que se concatenan los status y las restricciones y es difícil lograr casos tanto true como false en las restricciones finales ya que concantenan todas. Además, la línea que comprueba que la petición sea un post, no aparece en verde porque al hacer el update siempre se hace un post y por tanto el caso negativo no se puede probar. El filtrado de los flights sin legs asociadas también aparecen en amarillo ya que no hay ningún flight sin legs asociadas, pero en el que caso que lo hubiera, se filtraría. Por otro lado, las líneas en amarillo del validate no entiendo algunas por qué están en amarillo ya que probé a publicar una booking con los pasajeros sin publicar y con los pasajeros publicados. Por otro lado, la validación de que el vuelo esté publicado no la pude probar porque los vuelos que salen en el desplegable están publicados.

Para testear el hacking, he iniciado sesión con un realm distinto de Customer y he comprobado que se desautoriza la url del publish de una booking cualquiera. Además, me he logueado como customer2 y he comprobado que no puedo acceder al publish de una booking del customer1. Por otro lado, estando registrado como customer 1, he comprobado que no puedo acceder a la url del publish de una booking ya publicada del customer1. Además, he probado POST hacking del atributo flight introduciendo un id negativo (-9 o -1, no lo recuerdo), id positivo cualquiera (12) y un id correspondiente a un flight no publicado (96).

A screenshot of a computer

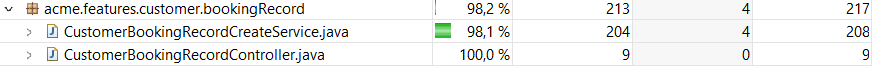
AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

A close-up of a line

AI-generated content may be incorrect.



BookingRecord Create Feature:

Esta es una entidad intermedia que relaciona Passenger y Booking, no he necesitado una feature de listado ya que, desde el show de una booking sin publicar, puedo añadir un pasajero lo cual me lleva a una pantalla donde elijo un pasajero para añadirlo al booking y tras esto, vuelve al show del booking. Además, cuando le doy a mostrar los pasajeros dentro del show de un booking, me lleva a un listado de pasajeros asociados a esa booking. Por todo ello, solo es necesaria la feature create.

Para realizar el testing de la feature de crear los bookingRecord, me he logueado como customer1, accedido al listado de bookings y posteriormente a la booking con draftMode true. Tras esto, le he dado a añadir pasajero y en esa pantalla he intentado añadir el pasajero, pero sin seleccionar uno y tras esto, he elegido un pasajero y lo he añadido.

Para testear el hacking, estando registrado como customer2, he intentado acceder a la pantalla de añadir un pasajero a una booking del customer1. Tras esto, mediante post hacking, he intentado añadir un pasajero con id negativo (-9), id positivo cualquiera (12) y un pasajero con id correcto pero que no pertenece al customer1 (368). También, he intentado añadir el mismo pasajero 2 veces a una booking lo cual está desautorizado.

A screenshot of a computer program

AI-generated content may be incorrect.

A green and blue text

AI-generated content may be incorrect.

A screenshot of a graph

AI-generated content may be incorrect.

Passenger List Feature:

Para realizar el testing de la feature de listar los passenger, me he logueado como customer1 y he accedido al listado de passengers. Por otro lado, el listado de pasajeros también se hace al listar los pasajeros añadidos a una booking. Por ello, he accedido también a una booking y le he dado a mostrar los pasajeros.

Con esto, he logrado un 100% de cobertura ya que el método authorise del listado de bookings era bastante sencillo no tenía más restricciones que comprobar que el cliente está registrado como Customer y que el customer registrado era el mismo que el que creó la booking. La única línea amarilla se debe a que no ha recibido ningún Passenger vacío para mostrar sus detalles ya que he accedido desde el propio listado.

Para comprobar el hacking, he iniciado sesión con un realm distinto de Customer y he comprobado que se desautoriza la url del listado de passengers. A continuación, he accedido a una booking cualquiera y le he dado a mostrar pasajeros. He copiado la url y he comprobado que estando registrado como customer2 no puedo acceder al listado de pasajeros de la booking del customer1.

A screenshot of a computer program

AI-generated content may be incorrect.

Passenger Show Feature:

Para realizar el testing de la feature de mostrar los detalles de los passengers, me he logueado como customer1, he entrado en el listado de passengers y he accedido a un Passenger cualquiera.

Con esto y junto con los tests del hacking, he logrado un 100% de cobertura. Esto se debe a que el código es bastante simple sin demasiadas restricciones.

Para testear el hacking, he iniciado sesión con un realm distinto de Customer y he comprobado que se desautoriza la url del show de un Passenger cualquiera. Además, me he logueado como customer2 y he comprobado que no puedo acceder al show de un passenger del customer1.

A screenshot of a computer code

AI-generated content may be incorrect.

Passenger Create Feature:

Para realizar el testing de la feature de crear los passenger, me he logueado como customer1, accedido al listado de passengers y posteriormente le ha dado a crear un Passenger.

En primer lugar, probé a crear un Passenger con todos los campos vacíos, y a continuación probé todos los campos uno a uno dejando el resto vacíos. En los strings, he probado todos los casos de prueba que aparecen en el Excel tanto positivos como negativos (probando los límites superiores e inferiores en función del máximo de caracteres, caracteres raros, de otros idiomas, inyecciones sql, etc.). En cuanto a los emails, también he probado con los casos positivos y negativos del Excel de pruebas. Por otro lado, en el passportNumber he probado distintas longitudes forzando los límites en casos positivos y negativos y probando números y letras. Finalmente, en cuanto al dateOfBirth, he probado los casos positivos y negativos del Excel forzando los límites.

Con estos casos de prueba y junto con el hacking he logrado un 96,6% apareciendo líneas en amarillo correspondientes a asegurarse de que el passenger no sea null en el método unbind que no se realmente como probarlo, pero en acme Jobs se incluía esa comprobación.

Para comprobar el hacking, he iniciado sesión con un realm distinto de Customer y he comprobado que se desautoriza la url del create de bookings.

A screenshot of a computer program

AI-generated content may be incorrect.

A computer screen shot of a computer code

AI-generated content may be incorrect.

Passenger Update Feature:

Para realizar el testing de la feature de editar los passengers, me he logueado como customer1, accedido al listado de passengers y posteriormente al Passenger con draftMode true.

En primer lugar, probé a editar el Passenger con todos los campos vacíos, y a continuación probé todos los campos uno a uno dejando el resto vacíos. En los strings, he probado todos los casos de prueba que aparecen en el Excel tanto positivos como negativos (probando los límites superiores e inferiores en función del máximo de caracteres, caracteres raros, de otros idiomas, inyecciones sql, etc.). En cuanto a los emails, también he probado con los casos positivos y negativos del Excel de pruebas. Por otro lado, en el passportNumber he probado distintas longitudes forzando los límites en casos positivos y negativos y probando números y letras. Finalmente, en cuanto al dateOfBirth, he probado los casos positivos y negativos del Excel forzando los límites.

Con estos casos de prueba y junto con el hacking he logrado un 100% apareciendo una única línea en amarillo la cual es difícil convertirla a verde ya que es una concatenación de todas las restricciones y es complicado conseguir los casos true y false con las posibles combinaciones.

Para testear el hacking, he iniciado sesión con un realm distinto de Customer y he comprobado que se desautoriza la url del update de un Passenger cualquiera. Además, me he logueado como customer2 y he comprobado que no puedo acceder al update de un passenger del customer1. Por otro lado, estando registrado como customer 1, he comprobado que no puedo acceder a la url del update de un passenger ya publicado del customer1.

A screenshot of a computer program

AI-generated content may be incorrect.

A computer screen shot of a computer code

AI-generated content may be incorrect.

Passenger Publish Feature:

Para realizar el testing de la feature de publicar los passengers, me he logueado como customer1, accedido al listado de passengers y posteriormente al passenger con draftMode true.

En primer lugar, probé a publicar el Passenger con todos los campos vacíos, y a continuación probé todos los campos uno a uno dejando el resto vacíos. En los strings, he probado todos los casos de prueba que aparecen en el Excel tanto positivos como negativos (probando los límites superiores e inferiores en función del máximo de caracteres, caracteres raros, de otros idiomas, inyecciones sql, etc.). En cuanto a los emails, también he probado con los casos positivos y negativos del Excel de pruebas. Por otro lado, en el passportNumber he probado distintas longitudes forzando los límites en casos positivos y negativos y probando números y letras. Finalmente, en cuanto al dateOfBirth, he probado los casos positivos y negativos del Excel forzando los límites.

Con estos casos de prueba y junto con el hacking he logrado un 100% de cobertura con una única línea a en amarillo que no se muy bien la razón ya que he probado a loguearme y publicar un passenger con un realm distinto de customer.

Para testear el hacking, he iniciado sesión con un realm distinto de Customer y he comprobado que se desautoriza la url del publish de un passenger cualquiera. Además, me he logueado como customer2 y he comprobado que no puedo acceder al publish de un passenger del customer1. Por otro lado, estando registrado como customer 1, he comprobado que no puedo acceder a la url del publish de un passenger ya publicado del customer1.

A screenshot of a computer program

AI-generated content may be incorrect.

A computer screen shot of a computer code

AI-generated content may be incorrect.

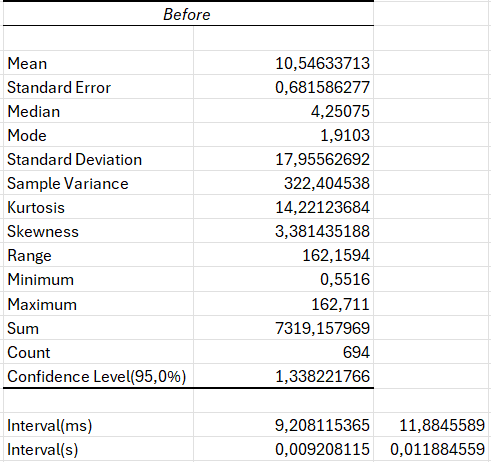
**PERFORMANCE TESTING:**

En esta sección se analizará el efecto de los índices en el rendimiento del proyecto. En primer lugar, analizaré el rendimiento de las features (antes y después de usar los índices) midiéndolo con el tiempo promedio de las distintas features y otras estadísticas calculadas con el data analysis. A continuación, haré un contraste de hipótesis relativo a la diferencia entre los intervalos de confianza antes y después. Finalmente, compararé el rendimiento antes y después de los índices basándome en el P-value obtenido con el Z-test.

Performance before índices:

A screenshot of a computer

AI-generated content may be incorrect.



Como se puede observar, las request más ineficientes son las relacionadas con booking lo cual es algo normal ya que son las más complejas y las que más validaciones incluyen. Destacan las features de create, publish y update de booking las cuales son las más complejas y con más dependencias. Por otro lado, el intervalo de confianza se encuentra en (0.009, 0.011) y por tanto, como el límite superior está bastante por debajo de 1 segundo, podemos deducir que se logra un buen rendimiento ajustado a los requisitos. Además, como se puede observar la media de tiempo de ejecución de las requests es de 10,546 ms.

Hipótesis sobre el uso de índices:

A screenshot of a computer code

AI-generated content may be incorrect.

A computer screen shot of text

AI-generated content may be incorrect.

A computer code with text

AI-generated content may be incorrect.

A la hora de analizar los repositorios y queries de las features, he llegado a la conclusión de que solo es necesario añadir índice al atributo locatorCode de booking. Para llegar a esto, he analizado el bloque where de las queries y he seleccionado aquellos atributos que no son ids ni atributos navegables ya que el propio framework se encarga de asignar los índices en estos casos.

Además, la entidad booking incluye la MIR (request más ineficiente), la relativa a la creación de una booking (también posee las otras 2 requests más ineficientes). Por ello, sería interesante añadir índices a esta entidad para intentar mejorar su rendimiento.

En conclusión, la inclusión del índice debería de mejorar el rendimiento de las requests relacionadas con booking aunque al solo incluir 1, la mejoría se supone que será muy pequeña.

Performance after índices:

A graph with blue and black bars

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a data sheet

AI-generated content may be incorrect.

Como se puede observar, las request más ineficientes siguen siendo relacionadas con booking lo cual es algo normal ya que aunque se han añadido índices, la diferencia con el resto de entidades es muy grande. Por otro lado, el intervalo de confianza se encuentra en (0.009, 0.012) lo cual nos indica que prácticamente no ha variado respecto a la anterior medición y por tanto, como el límite superior está bastante por debajo de 1 segundo, podemos deducir que se logra un buen rendimiento ajustado a los requisitos. Además, como se puede observar la media de tiempo de ejecución de las requests es de 10,645 ms, un pelín superior pero prácticamente igual que la anterior medición. Esto significa que ha aumentado un poco el tiempo medio de ejecución de las requests. Esto no tiene mucho sentido ya que se supone que los índices deben mejorar el rendimiento, pero al ser solo 1 el cambio no es notable. Además, el tiempo de ejecución puede variar cada vez que se repitan los tests (no es un valor constante) y por ello esta vez ha salido un poco superior pudiendo salir menor en otros momentos con menos carga computacional.

Comparación de los valores y conclusión:

Como se puede observar no se ha conseguido mejorar el rendimiento de las requests introduciendo índices (incluso ha empeorado ínfimamente, pero es algo que pude ocurrir) ya que el intervalo de confianza ha permanecido casi invariante al igual que la media de tiempo. Todo esto está respaldado con el resultado obtenido para el p-value en el Z-test.

A table with numbers and letters

AI-generated content may be incorrect.

Como se observa, el valor P(Z>=z)two-tail es de 0,919 lo cual está bastante cercano al 1 y muy por encima del Alpha(0,05). Esto respalda que los cambios añadidos (índices) no han resultado en una gran mejora de rendimiento y que los tiempos de ejecución son muy similares.  
Para concluir, el índice añadido no ha significado prácticamente ninguna mejora de rendimiento. Para lograr esta mejora sería necesario refactorizar código y quizás añadir otros atributos que aparecen en otra parte de código a las queries para posteriormente añadir sus índices.