



Testing Report

VIRGINIA MESA PEREZ

Repositorio: <https://github.com/virmesper/Acme-ANS-C3>

Student 2



ÍNDICE

Resumen ejecutivo.....	2
Introducción	3
Pruebas funcionales	4
Pruebas de rendimiento.....	7
Mutaciones.....	10
Conclusiones.....	14



Resumen Ejecutivo

Este es el proyecto del grupo C3.021 sobre Acme ANS, compañía ficticia especializada en organizar y controlar las operaciones de los aeropuertos y cuya logística de vuelos se gestiona con un WIS.

Para el correcto funcionamiento del sistema se debe escribir un reporte de testing donde se detallen los valores de cobertura y rendimiento del sistema.



Introducción

En este reporte se expondrán los resultados de los tests realizados durante la entrega C3 de la tercera convocatoria sobre las funcionalidades implementadas por el student 2, Virginia Mesa Pérez, sobre los requisitos 8 y 9.

Se reportarán 2 tipos de resultados, los relacionados con el testeo funcional y los relacionados con el testeo del rendimiento del sistema.

También se detallarán las 5 mutaciones realizadas sobre el código y los efectos que tienen.



Pruebas funcionales

- Resultados de bookings:

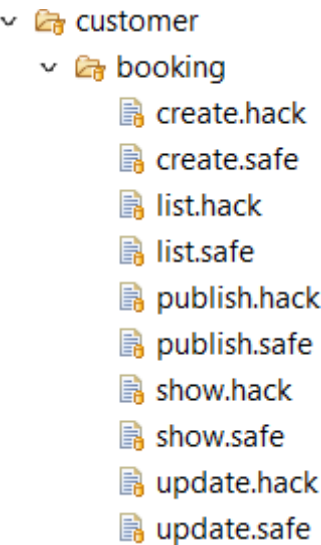


Figura 1: Archivos de testing para los bookings

Se encontraron varios bugs durante el testing, los cuales han sido solucionados, además de bastante “código espagueti” inútil en algunos casos. Los tests han ayudado a identificar código sobrante.

▼	acme.features.authenticated.customer.booking	81,5 %	936	213	1.149
>	CustomerBookingUpdateService.java	66,8 %	245	122	367
>	CustomerBookingPublishedService.java	58,9 %	99	69	168
>	CustomerBookingCreateService.java	94,9 %	333	18	351
>	CustomerBookingListService.java	96,2 %	100	4	104
>	CustomerBookingController.java	100,0 %	30	0	30
>	CustomerBookingShowService.java	100,0 %	129	0	129

Figura 2: Cobertura de los tests en los bookings

Se ha conseguido alcanzar una cobertura muy alta de la feature con un 81,5% que cubre las instrucciones a ejecutar y las excepciones esperadas.



- Resultados de Passengers:

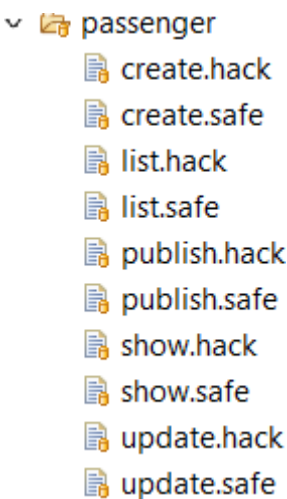


Figura 4: Archivos de testing para los passengers

En este caso no fueron encontrados bugs importantes, sino código innecesario.

La cobertura alcanzada ha sido mayor a la anterior con un 90,8%

▼	acme.features.authenticated.customer.passenger	<div><div></div></div> 90,8 %	787	80	867
>	CustomerPassengerPublishService.java	<div><div></div></div> 74,6 %	106	36	142
>	CustomerPassengerUpdateService.java	<div><div></div></div> 74,1 %	103	36	139
>	CustomerPassengerListService.java	<div><div></div></div> 96,1 %	122	5	127
>	CustomerPassengerCreateService.java	<div><div></div></div> 98,9 %	264	3	267
>	CustomerPassengerController.java	<div><div></div></div> 100,0 %	36	0	36
>	CustomerPassengerListAllService.java	<div><div></div></div> 100,0 %	57	0	57
>	CustomerPassengerShowService.java	<div><div></div></div> 100,0 %	99	0	99

Figura 5: Cobertura de los tests en los passengers



- Resultados de BookingRecord:

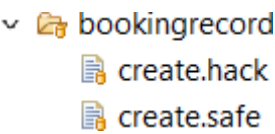


Figura 7: Archivos de testing para los bookingrecord

Lost tests de bookingrecord tienen una cobertura del 95,2% siendo la cobertura más alta de mis funcionalidades.

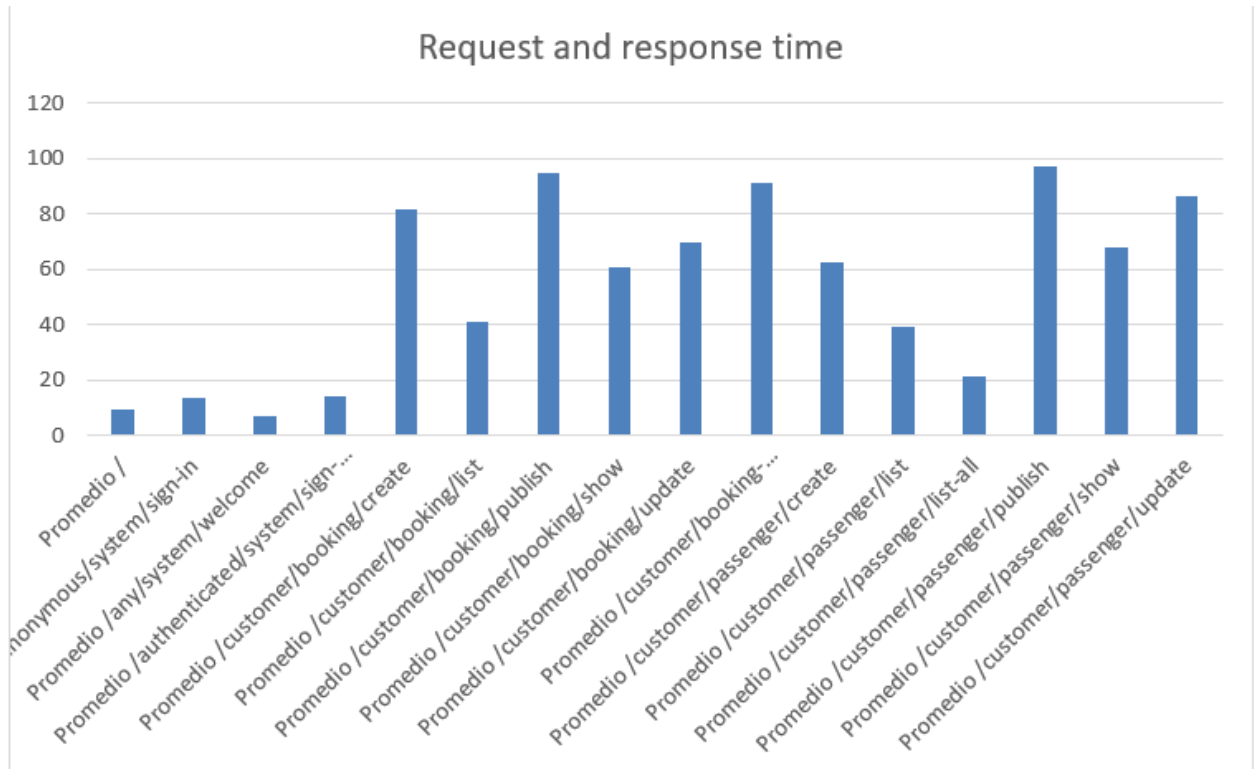
acme.features.authenticated.customer.bookingrecord	95,2 %	457	23	480
CustomerBookingRecordCreateService.java	95,1 %	448	23	471
CustomerBookingRecordController.java	100,0 %	9	0	9

Figura 8: Cobertura de los tests en los bookingrecord



Pruebas de rendimiento

- Resultados sin índices



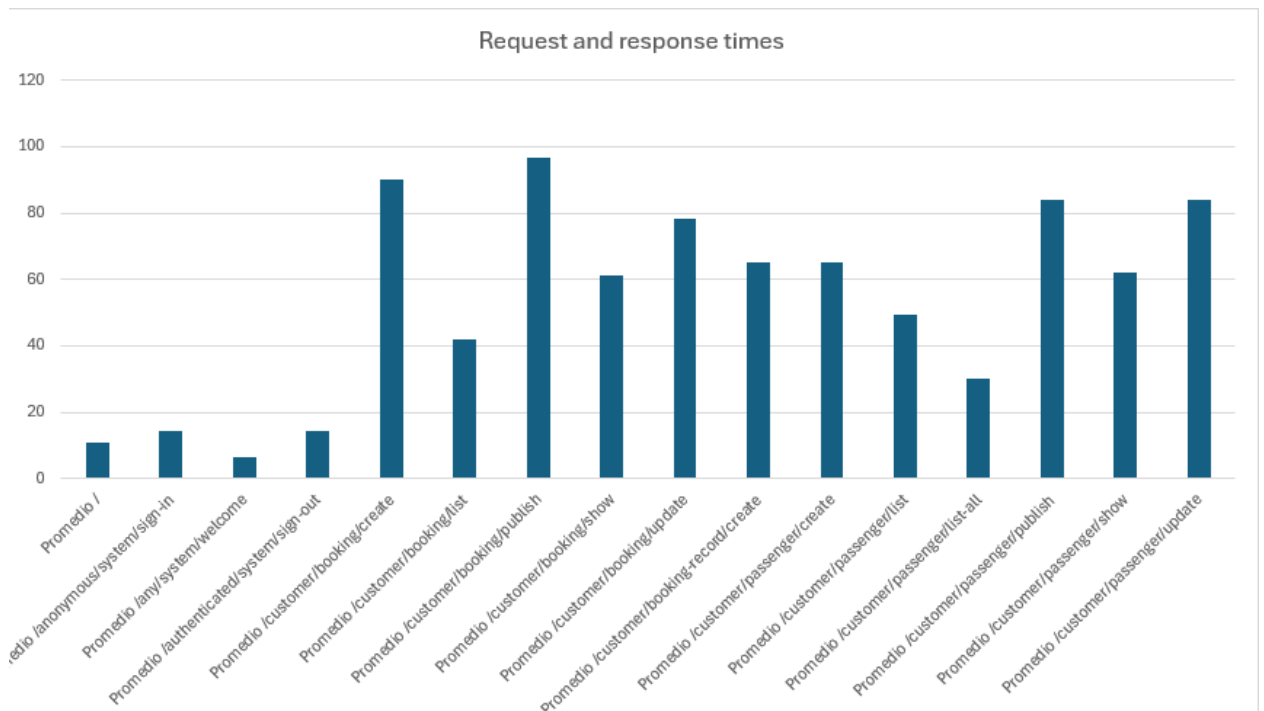
En esta tabla podemos observar los tiempos promedios de las distintas peticiones realizadas tanto a booking, passenger como bookingrecord con los índices calculados.

Las peticiones que más han tardado han sido las de publicación de datos ya que para testear todos los límites de los datos en el formulario han sido necesarias una enorme cantidad de peticiones, además de las numerosas comprobaciones de get hacking y post hacking.

Por otra parte, los lists no han requerido tantas peticiones (la gran mayoría de ellas han sido innecesarias, pues ocurrían al tratar de acceder a los bookings de estas como parte de otra prueba). Por último, también aparecen otras peticiones no tan relacionadas con el testeo de esta feature, que son las de iniciar sesión, cerrar sesión, y la página de bienvenida.



- Resultados con índices



Como se puede observar los tiempos han cambiado al introducir los índices. Teniendo un promedio general menor

	before		after
Media	21,56333373	Media	20,89786758
Error típico	1,405133641	Error típico	1,312357275
Mediana	8,4338	Mediana	8,3831
Moda	#N/D	Moda	#N/D
Desviación estándar	31,04041669	Desviación estándar	28,511698
Varianza de la muestra	963,5074686	Varianza de la muestra	812,9169229
Curtosis	20,91717133	Curtosis	19,9944867
Coficiente de asimetría	3,672768059	Coficiente de asimetría	3,541582263
Rango	302,8575	Rango	267,3147
Mínimo	2,6331	Mínimo	2,4173
Máximo	305,4906	Máximo	269,732
Suma	10522,90686	Suma	9863,7935
Cuenta	488	Cuenta	472
Nivel de confianza(95,0%)	2,760872762	Nivel de confianza(95,0%)	2,578799632



Prueba z para medias de dos muestras		
	193,9077	142,5922
Media	20,10791805	20,63949321
Varianza (conocida)	963,5074686	812,9169229
Observaciones	471	471
Diferencia hipotética de las medias	0	
z	-0,273716998	
P(Z<=z) una cola	0,392151057	
Valor crítico de z (una cola)	1,644853627	
Valor crítico de z (dos colas)	0,784302114	
Valor crítico de z (dos colas)	1,959963985	

El valor crítico de z (dos colas) es bastante menor que el intervalo de confianza, lo cual significa que se pueden comparar las medias de los tiempos. Fácilmente se puede ver que con índices ha tardado menos, lo cual nos confirma lo que habíamos dicho anteriormente: el uso de índices ha optimizado los tiempos de respuestas.

Debido a la falta de posesión de un segundo ordenador los cálculos solo han podido ser realizados en un único ordenador, siendo este suficiente para indicar los análisis requeridos y las diferencias en los resultados antes y después de los índices.



Mutaciones

- Mutación 1, Intentar crear un bookingRecord con un booking en borrador

```
authorised = bookingFromQuery != null
            && bookingFromQuery.getCustomer().getId() == activeCustomerId
            && !bookingFromQuery.isDraftMode();
```

Figura 1: Estado inicial del código

```
authorised = bookingFromQuery != null
            && bookingFromQuery.getCustomer().getId() == activeCustomerId
            && bookingFromQuery.isDraftMode();
```

Figura 2: Mutación 1

```
Loaded 251 requests from .\src\test\resources\customer\bookingrecord\create.safe
Resetting application (clear schema, populate sample, reset clock, reset randomiser).
Replaying .\src\test\resources\customer\bookingrecord\create.safe...
FAILED POST /customer/booking-record/create (request-id="ec816c91-044e-4ff2-a037-11721d0880ec",
input="id=0&version=0&booking=121&passenger=136&bookingId=121"): Expected 'status' to be '200', but got '500'.
FAILED GET /customer/passenger/list?bookingId=121 (request-id="98d752f3-8fe7-4e96-93fe-39d66e9bfff28", input=""): Expected 'payload' to be
'{bookingId=121, draftMode=false, email[0]=acme@lorem-ipsum.org, email[1]=acme@lorem-ipsum.org, email[2]=prueba@gmail.com, fullName[0]=Lorem ipsum
dolor sit amet, consectetur adipiscing elit. Sed do eiusmod tempor incididunt ut labore et dolore magna aliqua! Ut*, fullName[1]=Lorem ipsum dolor
sit amet, consectetur adipiscing elit. Sed do eiusmod tempor incididunt ut labore et dolore magna aliqua! Ut*, fullName[2]=prueba, id[0]=135, id
[1]=136, id[2]=769, passportNumber[0]=FKJZAG0XM, passportNumber[1]=FKJZAG0XM, passportNumber[2]=FKJZAG0V, service=615, specialNeeds[0]=Lorem ipsum
dolor sit amet, specialNeeds[1]=Lorem ipsum dolor sit amet, specialNeeds[2]=, version[0]=0, version[1]=0, version[2]=0}', but got '{bookingId=121,
draftMode=false, email[0]=acme@lorem-ipsum.org, email[1]=prueba@gmail.com, fullName[0]=Lorem ipsum dolor sit amet, consectetur adipiscing elit. Se
do eiusmod tempor incididunt ut labore et dolore magna aliqua! Ut*, fullName[1]=prueba, id[0]=135, id[1]=769, passportNumber[0]=FKJZAG0XM,
passportNumber[1]=FKJZAG0V, service=615, specialNeeds[0]=Lorem ipsum dolor sit amet, specialNeeds[1]=, version[0]=0, version[1]=0}'.
FAILED GET /customer/booking/show?id=121 (request-id="aed73cd9-5885-44a1-a0e5-963dc8c62520", input=""): Expected 'payload' to be '{bookingId=121,
draftMode=false, flightId=76, flights=[{"key": "0", "label": "----", "selected": false, "sealed": true}, {"key": "75", "label": "Lorem ipsum dolor sit
amet", "selected": false, "sealed": true}, {"key": "76", "label": "L", "selected": true, "sealed": true}, {"key": "77", "label": "Lo", "selected": false, "sealed": true}, {"key": "78", "label": "Lorem ipsum dolor sit amet, consectetur adipiscing", "selected": false, "sealed": true}, {"key": "79", "label": "Lorem ipsum dolor sit
amet, consectetur adipiscing", "selected": false, "sealed": true}, {"key": "80", "label": "Lorem ipsum dolor sit amet, consectetur
adipiscing", "selected": false, "sealed": true}, {"key": "81", "label": "لوريم إيبسوم", "selected": false, "sealed": true}, {"key": "82", "label": "لوريم
إيبسوم", "selected": false, "sealed": true}, {"key": "83", "label": "洛伦·伊普森", "selected": false, "sealed": true},
```

Figura 3: Fallo detectado



- **Mutación 2 — Aceptar desajuste entre booking (form) y bookingId (query)**

Qué cambia: en el POST, el id del select booking ya no tiene que coincidir con el bookingId de la URL.

Objetivo: los hack tests que fuerzan un booking distinto del de la query deberían fallar → mutación “killed”.

```
authorised = authorised
    && bookingFromQuery != null
    && bookingIdFromForm == bookingFromQuery.getId()
    && bookingFromQuery.getCustomer().getId() == activeCustomerId
    && !bookingFromQuery.isDraftMode();
```

Figura 4: Estado inicial del código

```
authorised = authorised
    && bookingFromQuery != null
    && bookingIdFromForm != bookingFromQuery.getId()
    && bookingFromQuery.getCustomer().getId() == activeCustomerId
    && !bookingFromQuery.isDraftMode();
```

Figura 5: Mutación 2

```
Replaying .\src\test\resources\customer\bookingrecord\create.hack...
ERROR Exception: acme.client.components.views.SelectChoices.getSelected(SelectChoices.java:133): Cannot invoke
"acme.client.components.views.SelectChoice.isSelected()" because "this.selected" is null
FAILED POST /customer/booking-record/create (request-id="da284fdc-a13c-4c96-9815-2f2438043838",
input="id=0&version=0&booking=133&passenger=136&bookingId=121"): Expected 'payload' to be '{service=615}', but got '{booking$error=La reserva no
pertenece a su cuenta.. No puede añadir pasajeros a una reserva ya publicada.., service=615}'.
Loaded 207 requests from .\src\test\resources\customer\passenger\create.hack
```

Figura 6: Fallo detectado



- Mutación 3, modificar los límites del lastCardDigits:

```
@Optional
@ValidString(pattern = "[0-9]{4}")
@Automapped
private String lastCardDigits;
```

Figura 7: Estado inicial del código

```
@Optional
@ValidString(pattern = "[0-9]{5}$")
@Automapped
private String lastCardDigits;
```

Figura 8: Mutación 3

```

playing .\src\test\resources\customerbooking\create.sale...
FAILED POST /customer/booking/create (request id="6ff4a22f-ec38-4a68-8580-c59b0524cca9"):
Input=id=0&version=0&lastCardDigits=123&locatorCode=MOJAI123&price=&travelClass=ECONOMY&flightId=81): Expected 'payload' to be '{flightId=81, flights=[{"key":"80", "label":"----", "selected":false, "sealed":true}, {"key":"81", "label":"Lorem ipsum dolor sit amet, selected":true, "sealed":true}, {"key":"82", "label":"L", "selected":false, "sealed":true}, {"key":"83", "label":"lo, "selected":false, "sealed":true}, {"key":"84", "label":"Lorem ipsum dolor sit amet, consectetur adipiscing", "selected":false, "sealed":true}, {"key":"85", "label":"Lorem ipsum dolor sit amet, consectetur adipiscing", "selected":false, "sealed":true}, {"key":"86", "label":"Lorem ipsum dolor sit amet, consectetur adipiscing", "selected":false, "sealed":true}, {"key":"87", "label":"لوريم إيبس", "selected":false, "sealed":true}, {"key":"88", "label":"لوريم إيبس", "selected":false, "sealed":true}, {"key":"89", "label":"洛伦·伊普森", "selected":false, "sealed":true}, {"key":"90", "label":"<marquee>Hacked</marquee>", "selected":false, "sealed":true}, {"key":"91", "label":"<script>alert('Hacked!');</script>", "selected":false, "sealed":true}, {"key":"92", "label":"' or 'A' = 'A', "selected":false, "sealed":true}, {"key":"93", "label":"L", "selected":false, "sealed":true}, {"key":"94", "label":"L", "selected":false, "sealed":true}, {"key":"95", "label":"L", "selected":false, "sealed":true}, {"key":"96", "label":"L", "selected":false, "sealed":true}, {"key":"97", "label":"L", "selected":false, "sealed":true}, {"key":"98", "label":"L", "selected":false, "sealed":true}, {"key":"99", "label":"L", "selected":false, "sealed":true}, {"key":"100", "label":"L", "selected":false, "sealed":true}, {"key":"101", "label":"L", "selected":false, "sealed":true}, {"key":"102", "label":"L", "selected":false, "sealed":true}, {"key":"103", "label":"L", "selected":false, "sealed":true}, {"key":"104", "label":"L", "selected":false, "sealed":true}, {"key":"105", "label":"L", "selected":false, "sealed":true}, {"key":"106", "label":"L", "selected":false, "sealed":true}], id=0, lastCardDigits=123, lastCardDigitsError=Must match pattern '^[0-9]{4}$', locatorCode=MOJAI123, price=EUR 500.00, purchaseMoment=2025/01/01 00:00, service=739, travelClass=[{"key":"0", "label":"----", "selected":false, "sealed":true}, {"key":"ECONOMY", "label":"ECONOMY", "selected":true, "sealed":true}, {"key":"BUSINESS", "label":"BUSINESS", "selected":false, "sealed":true}], version=0}, but got '{flightId=81, flights=[{"key":"0", "label":"----", "selected":false, "sealed":true}, {"key":"81", "label":"Lorem ipsum dolor sit amet, selected":true, "sealed":true}, {"key":"82", "label":"L", "selected":false, "sealed":true}, {"key":"83", "label":"lo, "selected":false, "sealed":true}, {"key":"84", "label":"Lorem ipsum dolor sit amet, consectetur adipiscing", "selected":false, "sealed":true}, {"key":"85", "label":"Lorem ipsum dolor sit amet, consectetur adipiscing", "selected":false, "sealed":true}, {"key":"86", "label":"Lorem ipsum dolor sit amet, consectetur adipiscing", "selected":false, "sealed":true}, {"key":"87", "label":"لوريم إيبس", "selected":false, "sealed":true}, {"key":"88", "label":"لوريم إيبس", "selected":false, "sealed":true}, {"key":"89", "label":"洛伦·伊普森", "selected":false, "sealed":true}, {"key":"90", "label":"<marquee>Hacked</marquee>", "selected":false, "sealed":true}, {"key":"91", "label":"<script>alert('Hacked!');</script>", "selected":false, "sealed":true}, {"key":"92", "label":"' or 'A' = 'A', "selected":false, "sealed":true}, {"key":"93", "label":"L", "selected":false, "sealed":true}, {"key":"94", "label":"L", "selected":false, "sealed":true}, {"key":"95", "label":"L", "selected":false, "sealed":true}, {"key":"96", "label":"L", "selected":false, "sealed":true}, {"key":"97", "label":"L", "selected":false, "sealed":true}, {"key":"98", "label":"L", "selected":false, "sealed":true}, {"key":"99", "label":"L", "selected":false, "sealed":true}, {"key":"100", "label":"L", "selected":false, "sealed":true}

```

Figura 9: Fallo detectado



- Mutación 4, autorizar a cualquiera en CustomerPassengerShowService

```
@Override
public void authorise() {
    int id;
    Passenger passenger;
    int customerId = super.getRequest().getPrincipal().getActiveRealm().getUserAccount().getId();

    id = super.getRequest().getData("id", int.class);
    passenger = this.repository.findPassengerById(id);

    boolean status = passenger.getCustomer().getUserAccount().getId() == customerId && super.getRequest().getPrincipal().hasRealmOfType(Customer.class);

    super.getResponse().setAuthorised(status);
}
```

Figura 10: Estado inicial del código

```
@Override
public void authorise() {
    super.getResponse().setAuthorised(true);
}
```

Figura 11: Mutación 4

```
Resetting application (clear schema, populate sample, reset clock, reset randomiser).
Replaying .\src\test\resources\customer\passenger\show.hack...
FAILED GET /customer/passenger/show?id=144 (request-id="4beeb4ce-b3ed-4718-b62c-a78af127256b", input=""): Expected 'status' to be '500', but got '200'. Expected 'payload' to be '{service=605}', but got '{dateOfBirth=2012/07/02 00:00, draftMode=true, email=acme@lorem-ipsum.org, fullName=marqueeHacked<marquee>, id=144, passportNumber=FKJZAG0XM, service=605, specialNeeds=Lorem ipsum dolor sit ame, version=0}'.
FAILED GET /customer/passenger/show?id=181 (request-id="2aa85d0c-03a7-4dae-8935-736202402fda", input=""): Expected 'status' to be '500', but got '200'. Expected 'payload' to be '{service=609}', but got '{dateOfBirth=2012/07/02 00:00, draftMode=false, email=acme@lorem-ipsum.org, fullName=Lorem ipsum dolor sit amet, consectetur adipiscing elit. Sed do eiusmod tempor incididunt ut labore et dolore magna aliqua! Ut*, id=181, passportNumber=FKJZAG0XM, service=609, specialNeeds=Lorem ipsum dolor sit ame, version=0}'.
FAILED GET /customer/passenger/show?id=137 (request-id="3ccde20-1b03-4e0a-b6e8-033211a0b675", input=""): Expected 'status' to be '500', but got '200'. Expected 'payload' to be '{service=603}', but got '{dateOfBirth=2012/07/02 00:00, draftMode=true, email=acme@lorem-ipsum.org, fullName=L, id=137, passportNumber=FKJZAG0XM, service=603, specialNeeds=Lorem ipsum dolor sit ame, version=0}'.
FAILED GET /customer/passenger/show?id=181 (request-id="a543f53c-c87b-4a17-adb2-ffa98909fd50", input=""): Expected 'status' to be '500', but got '200'. Expected 'payload' to be '{service=607}', but got '{dateOfBirth=2012/07/02 00:00, draftMode=false, email=acme@lorem-ipsum.org, fullName=Lorem ipsum dolor sit amet, consectetur adipiscing elit. Sed do eiusmod tempor incididunt ut labore et dolore magna aliqua! Ut*, id=181, passportNumber=FKJZAG0XM, service=607, specialNeeds=Lorem ipsum dolor sit ame, version=0}'.
FAILED GET /customer/passenger/show?id=137 (request-id="62047ed2-d888-455d-90fe-2387a150af56", input=""): Expected 'status' to be '500', but got '200'. Expected 'payload' to be '{service=618}', but got '{dateOfBirth=2012/07/02 00:00, draftMode=true, email=acme@lorem-ipsum.org, fullName=L, id=137, passportNumber=FKJZAG0XM, service=618, specialNeeds=Lorem ipsum dolor sit ame, version=0}'.
FAILED GET /customer/passenger/show?id=144 (request-id="baeb6975-be4d-4a68-8de8-71439beb0c86", input=""): Expected 'status' to be '500', but got '200'. Expected 'payload' to be '{service=604}', but got '{dateOfBirth=2012/07/02 00:00, draftMode=true, email=acme@lorem-ipsum.org, fullName=marqueeHacked<marquee>, id=144, passportNumber=FKJZAG0XM, service=604, specialNeeds=Lorem ipsum dolor sit ame, version=0}'.
Loaded 367 requests from .\src\test\resources\customer\hack\update.hack
```

Figura 12: Fallo detectado



- Mutación 5, en el CustomerPassengerListService, sustituir la llamada para mostrar solo los pasajeros de una reserva por mostrar todos los pasajeros de ese customer:

```
@Override
public void load() {
    int bookingId = super.getRequest().getData(CustomerPassengerListService.BOOKING_ID, int.class);
    int userAccountId = super.getRequest().getPrincipal().getAccountId();

    Booking booking = this.bookingRepository.findBookingById(bookingId);
    boolean isMine = booking.getCustomer().getUserAccount().getId() == userAccountId;

    super.getResponse().addGlobal(CustomerPassengerListService.BOOKING_ID, bookingId);
    super.getResponse().addGlobal("draftMode", booking.isDraftMode());

    if (isMine) {
        Collection<Passenger> passengers = this.bookingRepository.findPassengersByBooking(bookingId);
        super.getBuffer().addData(passengers);
    } else
        super.getResponse().setAuthorised(false);
}
```

Figura 13: Estado inicial del código

```
@Override
public void load() {
    int bookingId = super.getRequest().getData(CustomerPassengerListService.BOOKING_ID, int.class);
    int userAccountId = super.getRequest().getPrincipal().getAccountId();

    Booking booking = this.bookingRepository.findBookingById(bookingId);
    boolean isMine = booking.getCustomer().getUserAccount().getId() == userAccountId;

    super.getResponse().addGlobal(CustomerPassengerListService.BOOKING_ID, bookingId);
    super.getResponse().addGlobal("draftMode", booking.isDraftMode());

    if (isMine) {
        Collection<Passenger> passengers = this.repo.findPassengersByCustomer(booking.getCustomer().getId());
        super.getBuffer().addData(passengers);
    } else
        super.getResponse().setAuthorised(false);
}
```

Figura 14: Mutación 5

[illegible]

Figura 15: Fallo detectado



Como se puede ver, al tratar de listar solo los pasajeros de la reserva, se obtienen más datos de los esperados.

Conclusiones

Para concluir, no se han llegado a encontrar fallos significativos en el código durante los tests. Sin embargo, sí se han podido realizar varias optimizaciones de código, tanto mejorar el que era necesario en la aplicación como borrar líneas que sobraban.