



Testing Report

VIRGINIA MESA PEREZ

Repositorio: <https://github.com/DP2-C1-021/Acme-ANS-C2>

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 C2.021 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 C2 de la segunda 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:

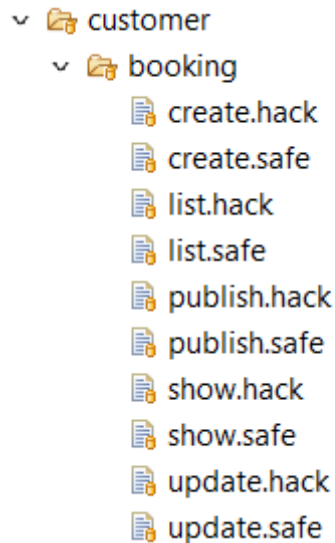


Figura 1: Archivos de testing para los bookings

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	90,9 %	964	97	1.061
CustomerBookingPublishedService.java	58,9 %	99	69	168
CustomerBookingUpdateService.java	95,1 %	349	18	367
CustomerBookingCreateService.java	97,7 %	257	6	263
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 90,9% que cubre las instrucciones a ejecutar y las excepciones esperadas.

```
@Override
public void authorise() {
    boolean authorised = super.getRequest().getPrincipal().hasRealmOfType(Customer.class);

    if (authorised && super.getRequest().hasData(CustomerBookingCreateService.FLIGHT_ID_FIELD))
        try {
            int flightId = super.getRequest().getData(CustomerBookingCreateService.FLIGHT_ID_FIELD, int.class);
            Flight flight = this.flightRepository.findFlightById(flightId);
```

Figura 3: Condición sin cubrir al completo en CustomerBookingCreateService

Esta condición se repite varias veces en el código y se debe a que, tal como está planteado el framework ahora mismo, nadie que no sea customer puede acceder



a esta función, por lo que la comprobación de si es customer sobraría. Sin embargo, con tal de evitar posibles errores en caso de que esto pueda cambiar en el futuro, he decidido dejar dicha comprobación.

- Resultados de Passengers:

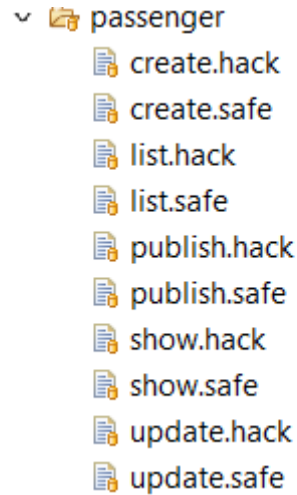


Figura 4: Archivos de testing para los passengers

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

La cobertura alcanzada ha sido un poco menor a la anterior con un 89.2% debido a los unbind de algunos archivos.

acme.features.authenticated.customer.passenger	89,2 %	638	77	715
> CustomerPassengerPublishService.java	74,6 %	106	36	142
> CustomerPassengerUpdateService.java	74,1 %	103	36	139
> CustomerPassengerListService.java	96,1 %	122	5	127
> CustomerPassengerController.java	100,0 %	36	0	36
> CustomerPassengerCreateService.java	100,0 %	115	0	115
> CustomerPassengerListAllService.java	100,0 %	57	0	57
> CustomerPassengerShowService.java	100,0 %	99	0	99

Figura 5: Cobertura de los tests en los passengers

```

@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);
}

@Override
public void unbind() {
}

```

Figura 6: Condiciones sin cubrir al completo

La causa de esta falta de cobertura está relacionada con el funcionamiento interno del framework: únicamente los usuarios autenticados con rol Customer pueden acceder a este servicio, por lo que la comprobación explícita de



hasRealmOfType(Customer.class) siempre resulta verdadera en tiempo de ejecución y no se llega a evaluar el caso contrario.

Este tipo de chequeo actúa, por tanto, como una verificación redundante, pero se ha decidido mantenerlo en el código como medida preventiva y defensiva, en caso de que el servicio se reutilice en otro entorno o que las reglas de acceso del framework cambien en futuras versiones. Así se garantiza que un usuario sin permisos no pueda explotar el servicio manipulando los datos desde el cliente.

- Resultados de BookingRecord:

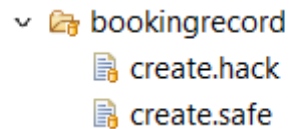


Figura 7: Archivos de testing para los bookingrecord

Los tests de bookingrecord tienen una cobertura del 94,5% siendo la cobertura más alta de mis funcionalidades.

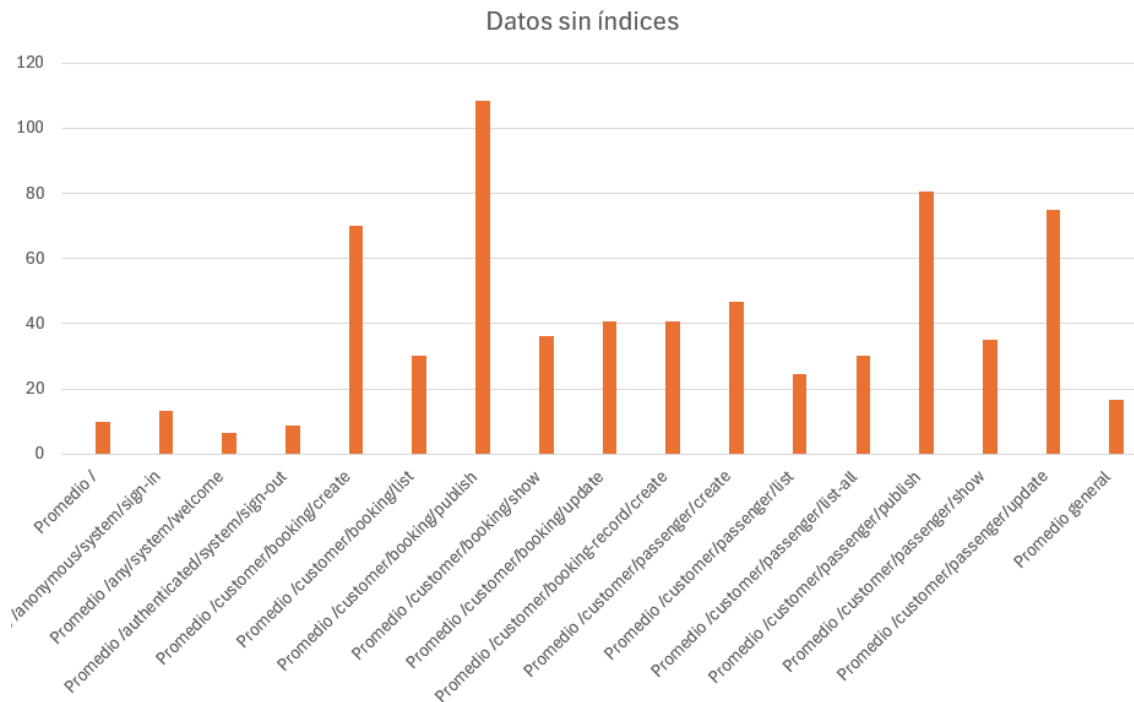
▼	acme.features.authenticated.customer.bookingrecord	94,5 %	274	16	290
>	CustomerBookingRecordCreateService.java	94,3 %	265	16	281
>	CustomerBookingRecordController.java	100,0 %	9	0	9

Figura 8: Cobertura de los tests en los bookingrecord



Pruebas de rendimiento

- Resultados sin índices



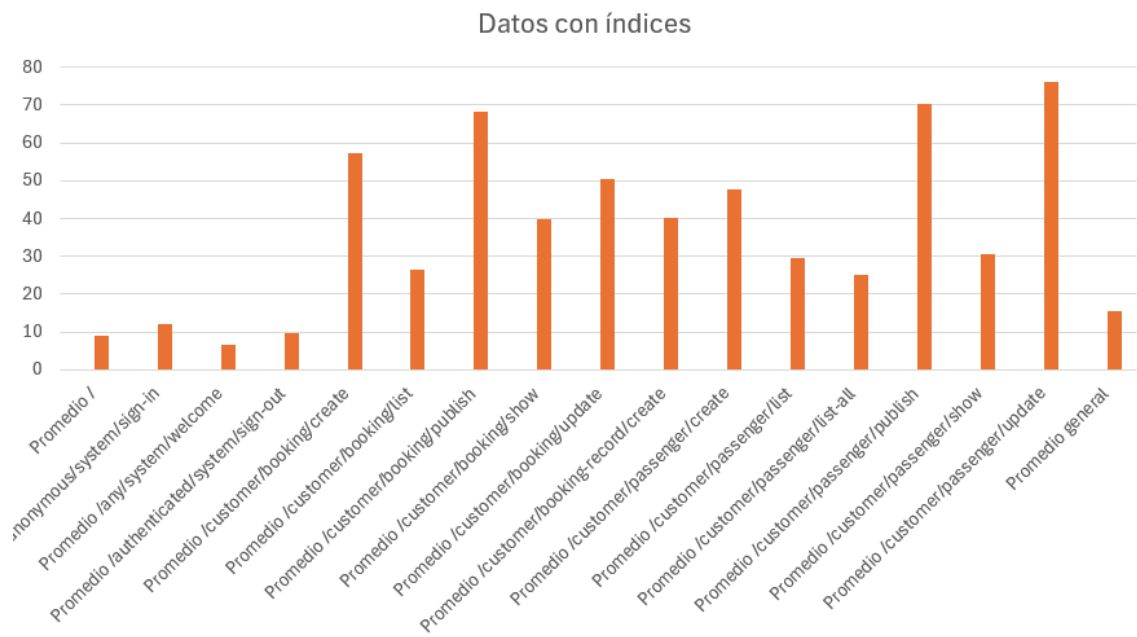
En esta tabla podemos observar los tiempos promedios de las distintas peticiones realizadas tanto a booking, passenegr como bookingrecord con los índices calculados.

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

Con Índices		Datos sin índices	
Media	15,5736299	Media	16,45946773
Error típico	0,79809659	Error típico	0,84212382
Mediana	8,1639005	Mediana	8,35715
Moda	3,6328	Moda	3,638
Desviación estándar	18,9537438	Desviación estándar	19,99933265
Varianza de la muestra	359,244403	Varianza de la muestra	399,9733063
Curtosis	25,1539176	Curtosis	17,95192868
Coeficiente de asimetría	4,07361446	Coeficiente de asimetría	3,478159062
Rango	193,752599	Rango	176,0487
Mínimo	2,8698	Mínimo	3,0489
Máximo	196,622399	Máximo	179,0976
Suma	8783,52727	Suma	9283,1398
Cuenta	564	Cuenta	564
Nivel de confianza(95,0%)	1,56761056	Nivel de confianza(95,0%)	1,654088264



Prueba z para medias de dos muestras		
	155,0564	142,935301
Media	16,21329201	15,3474103
Varianza (conocida)	399,9733063	359,244403
Observaciones	563	563
Diferencia hipotética de las medias	0	
z	0,745640725	
P(Z<=z) una cola	0,227942237	
Valor crítico de z (una cola)	1,644853627	
Valor crítico de z (dos colas)	0,455884475	
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, modificar los límites del lastCardDigits:

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

Figura 1: Estado inicial del código

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

Figura 2: Mutación 1

```

replaying - .\src\test\resources\customerbooking\create.sate...
FAILED POST /customer/booking/create (request-id="6ff4a22f-ee38-4a68-8580-c550b524cca9",
input=id=0&version=lastCardDigits=123&locatorCode=HOLA123&price=&travelClass=ECONOMY&flightId=81): Expected 'payload' to be {<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 ipsum", "selected": false, "sealed": true}, {"key": "88", "label": "لوريم ipsum", "selected": false, "sealed": true}, {"key": "89", "label": "洛伦·伊普
森", "selected": false, "sealed": true}, {"key": "90", "label": "marqueeHacked/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, lastCardDigits
&error=Must match pattern '[0-9]{4}'>, locatorCode=HOLA123, 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 ipsum", "selected": false, "sealed": true}, {"key": "88", "label": "لوريم ipsum", "selected": false, "sealed": true}, {"key": "89", "label": "洛伦·伊普
森", "selected": false, "sealed": true}, {"key": "90", "label": "marqueeHacked/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 3: Fallo detectado

Los tests fallan al cambiar este número de dígitos al no estar cumpliendo el límite de dígitos.

- Mutación 2, quitar un argumento en un `authorise` de `CustomerBookingUpdateService`:



```
@Override
public void authorise() {
    // ☑ Eliminar campos que pueden estar manipulados
    super.getRequest().getData().remove("price");
    super.getRequest().getData().remove("purchaseMoment");

    boolean status = super.getRequest().getPrincipal().hasRealmOfType(Customer.class);

    if (status && super.getRequest().hasData("id"))
        try {
            int bookingId = super.getRequest().getData("id", int.class);
            Booking booking = this.customerBookingRepository.findBookingById(bookingId);
            status = booking != null && super.getRequest().getPrincipal().hasRealm(booking.getCustomer()) && !booking.isDraftMode();
        } catch (final Throwable e) {
            status = false;
        }
    else
        status = false;

    if (status && super.getRequest().hasData("flightId"))
        try {
            int flightId = super.getRequest().getData("flightId", int.class);
            Flight flight = this.customerBookingRepository.findFlightById(flightId);
            status = flightId == 0 || flight != null && !flight.getDraftMode();
        } catch (final Throwable e) {
            status = false;
        }

    if (status && super.getRequest().hasData("travelClass"))
        try {
            TravelClass travelClass = super.getRequest().getData("travelClass", TravelClass.class);
            status = travelClass != null;
        } catch (final Throwable e) {
            status = false;
        }
}
```

Figura 4: Estado inicial del código

```
@Override
public void authorise() {
    // ☑ Eliminar campos que pueden estar manipulados
    super.getRequest().getData().remove("price");
    super.getRequest().getData().remove("purchaseMoment");

    boolean status = super.getRequest().getPrincipal().hasRealmOfType(Customer.class);

    if (status && super.getRequest().hasData("id"))
        try {
            int bookingId = super.getRequest().getData("id", int.class);
            Booking booking = this.customerBookingRepository.findBookingById(bookingId);
            status = booking != null && super.getRequest().getPrincipal().hasRealm(booking.getCustomer()) && !booking.isDraftMode();
        } catch (final Throwable e) {
            status = false;
        }
    else
        status = false;

    if (status && super.getRequest().hasData("flightId"))
        try {
            int flightId = super.getRequest().getData("flightId", int.class);
            Flight flight = this.customerBookingRepository.findFlightById(flightId);
            status = flightId == 0 || flight != null && !flight.getDraftMode();
        } catch (final Throwable e) {
            status = false;
        }
    }

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

Figura 5: Mutación 2

```
Resetting application (clear schema, populate sample, reset clock, reset randomiser).
Replaying .\src\test\resources\customer\booking\update.hack...
ERROR Exception: org.springframework.core.convert.support.ConversionUtils.invokeConverter(ConversionUtils.java:47): Failed to convert from type
[java.lang.String] to type [acme.entities.student2.TravelClass] for value [HACEKD]; nested exception is java.lang.IllegalArgumentException: No enum constant
acme.entities.student2.TravelClass.HACEKD
java.base/java.lang.Enum.valueOf(Enum.java:293): No enum constant acme.entities.student2.TravelClass.HACEKD
Loaded 223 requests from .\src\test\resources\customer\passenger\update.hack
Resetting application (clear schema, populate sample, reset clock, reset randomiser).
```

Figura 6: Fallo detectado

Falla porque al no validar en el método si el travelclass se ha modificado con f12, se puede hacer get hacking



- Mutación 3, eliminar condición en authorised de CustomerPassengerListAllService

```
@Override
public void authorize() {
    boolean isCustomer = super.getRequest().getPrincipal().hasRealmOfType(Customer.class);
    super.getResponse().setAuthorized(isCustomer);
}
```

Figura 7: Estado inicial del código

```
@Override
public void authorise() {
}
```

Figura 8: Mutación 3

```
[1]=FKJZAG0X0, passportNumber[19]=P2V9389, passportNumber[1]=FKJZAG0X0, passportNumber[20]=XP2MR8IA, passportNumber[21]=R9P1NYUW, passportNumber[22]=FKJZAG0X0, passportNumber[23]=FKJZAG0X0, passportNumber[24]=FKJZAG0X0, passportNumber[25]=FKJZAG0X0, passportNumber[26]=FKJZAG0X0, passportNumber[27]=FKJZAG0X0, passportNumber[28]=FKJZAG0X0, passportNumber[29]=FKJZAG0X0, passportNumber[30]=FKJZAG0X0, passportNumber[31]=FKJZAG0X0, passportNumber[32]=FKJZAG0X0, passportNumber[33]=FKJZAG0X0, passportNumber[34]=FKJZAG0X0, passportNumber[35]=FKJZAG0X0, passportNumber[31]=FKJZAG0X0, passportNumber[4]=FKJZAG0X0, passportNumber[5]=FKJZAG0X0, passportNumber[6]=FKJZAG0X0, passportNumber[7]=FKJZAG0X0, passportNumber[8]=FKJZAG0X0, passportNumber[9]=FKJZAG0X0, service=725, specialNeeds[0]=Lorem ipsum dolor sit ame, specialNeeds[10]=Lorem ipsum dolor sit ame, specialNeeds[11]=Lorem ipsum dolor sit ame, specialNeeds[12]=Lorem ipsum dolor sit ame, specialNeeds[13]=Lorem ipsum dolor sit ame, specialNeeds[14]=Lorem ipsum dolor sit ame, specialNeeds[15]=Lorem ipsum dolor sit ame, specialNeeds[16]=Lorem ipsum dolor sit ame, specialNeeds[17]=Lorem ipsum dolor sit ame, specialNeeds[18]=Lorem ipsum dolor sit ame, specialNeeds[19]=Lorem ipsum dolor sit ame, specialNeeds[20]=Lorem ipsum dolor sit ame, specialNeeds[21]=Lorem ipsum dolor sit ame, specialNeeds[22]=Lorem ipsum dolor sit ame, specialNeeds[23]=lo, specialNeeds[24]=lo, specialNeeds[25]=Lorem ipsum dolor sit amet, consectetur adipiscing, specialNeeds[26]=Lorem ipsum dolor sit amet, consectetur adipiscing, specialNeeds[27]=لوريم دبلوم, specialNeeds[28]=لوريم لوريم, specialNeeds[29]=圆|伊晋碧, specialNeeds[30]=marqueeHacked/marquee, specialNeeds[31]=<script>alert('Hacked!');</script>, specialNeeds[32]=' or 'A' = 'A', specialNeeds[33]=Lorem ipsum dolor sit ame, specialNeeds[34]=Lorem ipsum dolor sit ame, specialNeeds[35], specialNeeds[33]=Lorem ipsum dolor sit ame, specialNeeds[4]=Lorem ipsum dolor sit ame, specialNeeds[5]=Lorem ipsum dolor sit ame, specialNeeds[6]=Lorem ipsum dolor sit ame, specialNeeds[7]=Lorem ipsum dolor sit ame, specialNeeds[8]=Lorem ipsum dolor sit ame, version[0]=0, version[11]=0, version[12]=0, version[13]=0, version[14]=0, version[15]=0, version[16]=0, version[17]=0, version[18]=0, version[19]=0, version[20]=0, version[21]=0, version[22]=0, version[23]=0, version[24]=0, version[25]=0, version[26]=0, version[27]=0, version[28]=0, version[29]=0, version[30]=0, version[31]=0, version[32]=0, version[33]=0, version[34]=0, version[35]=0, version[36]=0, version[4]=0, version[5]=0, version[6]=0, version[7]=0, version[8]=0, version[9]=0'', but got ('service=725')  
FAILED GET /customer/passenger/list-all {request-id='06b61110-7a5f-4f71-9656-5544d3af9f44', input='': Expected 'status' to be '200', but got '500'. Expected 'payload' to be '{\"email\":\"acme@loreum-ipsum.org\", email[10]=@bb.es, email[11]=@bb.es, email[12]=loreum-ipsum@dolor.sit.amet.consectetur.adipiscing.elit.sed.do eiusmod.tempor.incidunt.ut.labor.e.dolor.magna.aliqua.Ut.enim.ad.minim.veniam.quis.nostrud.exercitation ullamco.labors.nisi.ut.aliquip.ex.ea.commodo.consequat.duis aute.irure.doloris.ese, email[13]=loreum-ipsum@dolor.sit.amet.consectetur.adipiscing.elit.sed.do eiusmod.tempor.incidunt.ut.labor.e.dolor.magna.aliqua.Ut.enim.ad.minim.veniam.quis.nostrud.exercitation ullamco.labors.nisi.ut.aliquip.ex.ea.commodo.consequat.duis aute.irure.dolor.es.org, email[14]=acme@loreum-ipsum.org, email[15]=acme@loreum-ipsum.org, email[16]=acme@loreum-ipsum.org, email[17]=acme@loreum-ipsum.org, email[18]=acme@loreum-ipsum.org, email[19]=acme@loreum-ipsum.org, email[21]=acme@loreum-ipsum.org, email[22]=acme@loreum-ipsum.org, email[23]=acme@loreum-ipsum.org, email[24]=acme@loreum-ipsum.org, email[25]=acme@loreum-ipsum.org, email[26]=acme@loreum-ipsum.org, email[27]=acme@loreum-ipsum.org, email[28]=acme@loreum-ipsum.org, email[29]=acme@loreum-ipsum.org, email[30]=acme@loreum-ipsum.org, email[31]=acme@loreum-ipsum.org, email[32]=acme@loreum-ipsum.org, email[33]=acme@loreum-ipsum.org, email[34]=acme@loreum-ipsum.org, email[35]=acme@loreum-ipsum.org, email[36]=acme@loreum-ipsum.org, email[37]=acme@loreum-ipsum.org, email[38]=acme@loreum-ipsum.org, email[39]=acme@loreum-ipsum.org, email[40]=acme@loreum-ipsum.org, email[41]=acme@loreum-ipsum.org, email[42]=acme@loreum-ipsum.org, email[43]=acme@loreum-ipsum.org, email[44]=acme@loreum-ipsum.org, email[45]=acme@loreum-ipsum.org, email[46]=acme@loreum-ipsum.org, email[47]=acme@loreum-ipsum.org, email[48]=acme@loreum-ipsum.org, email[49]=acme@loreum-ipsum.org, email[50]=acme@loreum-ipsum.org, email[51]=acme@loreum-ipsum.org, email[52]=acme@loreum-ipsum.org, email[53]=acme@loreum-ipsum.org, email[54]=acme@loreum-ipsum.org, email[55]=acme@loreum-ipsum.org, email[56]=acme@loreum-ipsum.org, email[57]=acme@loreum-ipsum.org, email[58]=acme@loreum-ipsum.org, email[59]=acme@loreum-ipsum.org, email[60]=acme@loreum-ipsum.org, email[61]=acme@loreum-ipsum.org, email[62]=acme@loreum-ipsum.org, email[63]=acme@loreum-ipsum.org, email[64]=acme@loreum-ipsum.org, email[65]=acme@loreum-ipsum.org, email[66]=acme@loreum-ipsum.org, email[67]=acme@loreum-ipsum.org, email[68]=acme@loreum-ipsum.org, email[69]=acme@loreum-ipsum.org, email[70]=acme@loreum-ipsum.org, email[71]=acme@loreum-ipsum.org, email[72]=acme@loreum-ipsum.org, email[73]=acme@loreum-ipsum.org, email[74]=acme@loreum-ipsum.org, email[75]=acme@loreum-ipsum.org, email[76]=acme@loreum-ipsum.org, email[77]=acme@loreum-ipsum.org, email[78]=acme@loreum-ipsum.org, email[79]=acme@loreum-ipsum.org, email[80]=acme@loreum-ipsum.org, email[81]=acme@loreum-ipsum.org, email[82]=acme@loreum-ipsum.org, email[83]=acme@loreum-ipsum.org, email[84]=acme@loreum-ipsum.org, email[85]=acme@loreum-ipsum.org, email[86]=acme@loreum-ipsum.org, email[87]=acme@loreum-ipsum.org, email[88]=acme@loreum-ipsum.org, email[89]=acme@loreum-ipsum.org, email[90]=acme@loreum-ipsum.org, email[91]=acme@loreum-ipsum.org, email[92]=acme@loreum-ipsum.org, email[93]=acme@loreum-ipsum.org, email[94]=acme@loreum-ipsum.org, email[95]=acme@loreum-ipsum.org, email[96]=acme@loreum-ipsum.org, email[97]=acme@loreum-ipsum.org, email[98]=acme@loreum-ipsum.org, email[99]=acme@loreum-ipsum.org, email[100]=acme@loreum-ipsum.org, email[101]=acme@loreum-ipsum.org, email[102]=acme@loreum-ipsum.org, email[103]=acme@loreum-ipsum.org, email[104]=acme@loreum-ipsum.org, email[105]=acme@loreum-ipsum.org, email[106]=acme@loreum-ipsum.org, email[107]=acme@loreum-ipsum.org, email[108]=acme@loreum-ipsum.org, email[109]=acme@loreum-ipsum.org, email[110]=acme@loreum-ipsum.org, email[111]=acme@loreum-ipsum.org, email[112]=acme@loreum-ipsum.org, email[113]=acme@loreum-ipsum.org, email[114]=acme@loreum-ipsum.org, email[115]=acme@loreum-ipsum.org, email[116]=acme@loreum-ipsum.org, email[117]=acme@loreum-ipsum.org, email[118]=acme@loreum-ipsum.org, email[119]=acme@loreum-ipsum.org, email[120]=acme@loreum-ipsum.org, email[121]=acme@loreum-ipsum.org, email[122]=acme@loreum-ipsum.org, email[123]=acme@loreum-ipsum.org, email[124]=acme@loreum-ipsum.org, email[125]=acme@loreum-ipsum.org, email[126]=acme@loreum-ipsum.org, email[127]=acme@loreum-ipsum.org, email[128]=acme@loreum-ipsum.org, email[129]=acme@loreum-ipsum.org, email[130]=acme@loreum-ipsum.org, email[131]=acme@loreum-ipsum.org, email[132]=acme@loreum-ipsum.org, email[133]=acme@loreum-ipsum.org, email[134]=acme@loreum-ipsum.org, email[135]=acme@loreum-ipsum.org, email[136]=acme@loreum-ipsum.org, email[137]=acme@loreum-ipsum.org, email[138]=acme@loreum-ipsum.org, email[139]=acme@loreum-ipsum.org, email[140]=acme@loreum-ipsum.org, email[141]=acme@loreum-ipsum.org, email[142]=acme@loreum-ipsum.org, email[143]=acme@loreum-ipsum.org, email[144]=acme@loreum-ipsum.org, email[145]=acme@loreum-ipsum.org, email[146]=acme@loreum-ipsum.org, email[147]=acme@loreum-ipsum.org, email[148]=acme@loreum-ipsum.org, email[149]=acme@loreum-ipsum.org, email[150]=acme@loreum-ipsum.org, email[151]=acme@loreum-ipsum.org, email[152]=acme@loreum-ipsum.org, email[153]=acme@loreum-ipsum.org, email[154]=acme@loreum-ipsum.org, email[155]=acme@loreum-ipsum.org, email[156]=acme@loreum-ipsum.org, email[157]=acme@loreum-ipsum.org, email[158]=acme@loreum-ipsum.org, email[159]=acme@loreum-ipsum.org, email[160]=acme@loreum-ipsum.org, email[161]=acme@loreum-ipsum.org, email[162]=acme@loreum-ipsum.org, email[163]=acme@loreum-ipsum.org, email[164]=acme@loreum-ipsum.org, email[165]=acme@loreum-ipsum.org, email[166]=acme@loreum-ipsum.org, email[167]=acme@loreum-ipsum.org, email[168]=acme@loreum-ipsum.org, email[169]=acme@loreum-ipsum.org, email[170]=acme@loreum-ipsum.org, email[171]=acme@loreum-ipsum.org, email[172]=acme@loreum-ipsum.org, email[173]=acme@loreum-ipsum.org, email[174]=acme@loreum-ipsum.org, email[175]=acme@loreum-ipsum.org, email[176]=acme@loreum-ipsum.org, email[177]=acme@loreum-ipsum.org, email[178]=acme@loreum-ipsum.org, email[179]=acme@loreum-ipsum.org, email[180]=acme@loreum-ipsum.org, email[181]=acme@loreum-ipsum.org, email[182]=acme@loreum-ipsum.org, email[183]=acme@loreum-ipsum.org, email[184]=acme@loreum-ipsum.org, email[185]=acme@loreum-ipsum.org, email[186]=acme@loreum-ipsum.org, email[187]=acme@loreum-ipsum.org, email[188]=acme@loreum-ipsum.org, email[189]=acme@loreum-ipsum.org, email[190]=acme@loreum-ipsum.org, email[191]=acme@loreum-ipsum.org, email[192]=acme@loreum-ipsum.org, email[193]=acme@loreum-ipsum.org, email[194]=acme@loreum-ipsum.org, email[195]=acme@loreum-ipsum.org, email[196]=acme@loreum-ipsum.org, email[197]=acme@loreum-ipsum.org, email[198]=acme@loreum-ipsum.org, email[199]=acme@loreum-ipsum.org, email[200]=acme@loreum-ipsum.org, email[201]=acme@loreum-ipsum.org, email[202]=acme@loreum-ipsum.org, email[203]=acme@loreum-ipsum.org, email[204]=acme@loreum-ipsum.org, email[205]=acme@loreum-ipsum.org, email[206]=acme@loreum-ipsum.org, email[207]=acme@loreum-ipsum.org, email[208]=acme
```

Figura 9: Fallo detectado

Falla el test ya que no está validando si es customer el que intenta acceder a esa url.

- Mutación 4, Cambiar en CustomerBookingRecordCreateService que en lugar de no mostrar los que ya están asociados a la booking, muestre todos los asociados al customer.

```

1 @Override
2 public void unbind(final BookingRecord bookingRecord) {
3     Integer customerId = super.getRequest().getPrincipal().getActiveRealm().getId();
4     Integer bookingId = super.getRequest().getData("bookingId", int.class);
5
6     Booking booking = this.customerBookingRecordRepository.findBookingById(bookingId);
7     Collection<Passenger> availablePassengers = this.customerBookingRecordRepository.getAvailablePassengersByCustomerAndBooking(customerId, bookingId);
8
9     SelectChoices passengerChoices = SelectChoices.from(availablePassengers, "fullName", bookingRecord.getPassenger());
10    SelectChoices bookingChoices = SelectChoices.from(List.of(booking), "locatorCode", bookingRecord.getBooking());
11
12    Dataset dataset = super.unbindObject(bookingRecord, "passenger", "booking");
13    dataset.put("passengers", passengerChoices);
14    dataset.put("bookings", bookingChoices);
15    dataset.put("bookingId", bookingId);
16
17    super.getResponse().addData(dataset);
18 }

```

Figura 10: Estado inicial del código



```

@Override
public void unbind(final BookingRecord bookingRecord) {
    Integer customerId = super.getRequest().getPrincipal().getActiveRealm().getId();
    Integer bookingId = super.getRequest().getData("bookingId", int.class);

    Booking booking = this.customerBookingRecordRepository.findBookingById(bookingId);
    Collection<Passenger> availablePassengers = this.customerBookingRecordRepository.getAllPassengersByCustomer(customerId);

    SelectChoices passengerChoices = SelectChoices.from(availablePassengers, "fullName", bookingRecord.getPassenger());
    SelectChoices bookingChoices = SelectChoices.from(List.of(booking), "locatorCode", bookingRecord.getBooking());

    Dataset dataset = super.unbindObject(bookingRecord, "passenger", "booking");
    dataset.put("passengers", passengerChoices);
    dataset.put("bookings", bookingChoices);
    dataset.put("bookingId", bookingId);

    super.getResponse().addData(dataset);
}

```

Figura 11: Mutación 4

```

Failed GET /customer/booking-record/create?bookingId=112 (request-id: '26e910ae-3673-46ec-a61e-2bdf0b2b9d06', input=''): Expected 'payload' to be
'{"bookingId":112,"bookings":[{"key":"0","label":"----","selected":true,"sealed":true}, {"key":"112","label":"890FTIV","selected":false,"sealed":true}], id=0,
passengers=[{"key":"0","label":"-----","selected":true,"sealed":true}, {"key":"142","label":"Lorem ipsum dolor sit amet, consectetur adipiscing elit. Sed do:
eiusmod tempor incididunt ut labore et dolore magna aliqua! Ut", "selected":false,"sealed":true}, {"key":"143","label":"I", "selected":false,"sealed":true},
{"key":"144","label":"Lo", "selected":false,"sealed":true}, {"key":"145","label":"Lorem ipsum dolor sit amet, consectetur adipiscing elit. Sed do: eiusmod
tempor incididunt ut labore et dolore magna aliqua! Ut enim ad minim veniam? Quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo
consequat. Duis aute irure dolor", "selected":false,"sealed":true}, {"key":"146","label":"Lorem ipsum dolor sit amet, consectetur adipiscing elit. Sed do:
eiusmod tempor incididunt ut labore et dolore magna aliqua! Ut enim ad minim veniam? Quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo
consequat. Duis aute irure dolor", "selected":false,"sealed":true}, {"key":"152","label":"." or "A" = "A", "selected":false,"sealed":true},
{"key":"153","label":"Lorem ipsum dolor sit amet, consectetur adipiscing elit. Sed do: eiusmod tempor incididunt ut labore et dolore magna aliqua!
Ut", "selected":false,"sealed":true}, {"key":"154","label":"Lorem ipsum dolor sit amet, consectetur adipiscing elit. Sed do: eiusmod tempor incididunt ut
labore et dolore magna aliqua! Ut", "selected":false,"sealed":true}, {"key":"155","label":"Lorem ipsum dolor sit amet, consectetur adipiscing elit. Sed do:
eiusmod tempor incididunt ut labore et dolore magna aliqua! Ut", "selected":false,"sealed":true}, {"key":"156","label":"Lorem ipsum dolor sit amet, consectetur
adipiscing elit. Sed do: eiusmod tempor incididunt ut labore et dolore magna aliqua! Ut", "selected":false,"sealed":true}, {"key":"157","label":"Lorem
ipsum dolor sit amet, consectetur adipiscing elit. Sed do: eiusmod tempor incididunt ut labore et dolore magna aliqua! Ut", "selected":false,"sealed":true},
{"key":"158","label":"Lorem ipsum dolor sit amet, consectetur adipiscing elit. Sed do: eiusmod tempor incididunt ut labore et dolore magna aliqua!
Ut", "selected":false,"sealed":true}, {"key":"159","label":"Lorem ipsum dolor sit amet, consectetur adipiscing elit. Sed do: eiusmod tempor incididunt ut
labore et dolore magna aliqua! Ut", "selected":false,"sealed":true}, {"key":"160","label":"Lorem ipsum dolor sit amet, consectetur adipiscing elit. Sed do:
eiusmod tempor incididunt ut labore et dolore magna aliqua! Ut", "selected":false,"sealed":true}, {"key":"161","label":"Lorem ipsum dolor sit amet, consectetur
adipiscing elit. Sed do: eiusmod tempor incididunt ut labore et dolore magna aliqua! Ut", "selected":false,"sealed":true}, {"key":"162","label":"Lorem
ipsum dolor sit amet, consectetur adipiscing elit. Sed do: eiusmod tempor incididunt ut labore et dolore magna aliqua! Ut", "selected":false,"sealed":true},
{"key":"163","label":"Lorem ipsum dolor sit amet, consectetur adipiscing elit. Sed do: eiusmod tempor incididunt ut labore et dolore magna aliqua!
Ut", "selected":false,"sealed":true}, {"key":"164","label":"Lorem ipsum dolor sit amet, consectetur adipiscing elit. Sed do: eiusmod tempor incididunt ut
labore et dolore magna aliqua! Ut", "selected":false,"sealed":true}, {"key":"165","label":"Lorem ipsum dolor sit amet, consectetur adipiscing elit. Sed do:
eiusmod tempor incididunt ut labore et dolore magna aliqua! Ut", "selected":false,"sealed":true}, {"key":"166","label":"Lorem ipsum dolor sit amet, consectetur
adipiscing elit. Sed do: eiusmod tempor incididunt ut labore et dolore magna aliqua! Ut", "selected":false,"sealed":true}, {"key":"167","label":"Lorem
ipsum dolor sit amet, consectetur adipiscing elit. Sed do: eiusmod tempor incididunt ut labore et dolore magna aliqua! Ut", "selected":false,"sealed":true},
{"key":"174","label":"Lorem ipsum dolor sit amet, consectetur adipiscing elit. Sed do: eiusmod tempor incididunt ut labore et dolore magna aliqua!
Ut", "selected":false,"sealed":true}, {"key":"175","label":"Lorem ipsum dolor sit amet, consectetur adipiscing elit. Sed do: eiusmod tempor incididunt ut
labore et dolore magna aliqua! Ut", "selected":false,"sealed":true} {"key":"176","label":"Lorem ipsum dolor sit amet, consectetur adipiscing elit. Sed do:

```

Figura 12: Fallo detectado

El test de create bookingrecord falla ya que esta renderizando más pasajeros de los que debe ya que está mostrando los que ya están asociados a esa booking tambien.

- 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 userId = super.getRequest().getPrincipal().getAccountId();

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

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

    if (isMine) {
        Collection<Passenger> passengers = this.passengerRepository.findPassengerByCustomer(userId);
        super.getBuffer().addData(passengers);
    } else {
        super.getResponse().setAuthorized(false);
    }
}
```

Figura 14: Mutación 5

```
Falling .\src\tester\rresources(customer.booking.show.saf...  
RAILED GET /customer/passenger/list?bookingId=110 (request-id=a0d8727f-6c99-4bf2-854d-f0f78ea0fbcc), input=''): Expected 'payload' to be '(bookingId=110,  
draftMode=false, email[0]=acme@lorem-ipsum.org, email[1]=acme@lorem-ipsum.org, email[2]=acme@lorem-ipsum.org, email[3]=acme@lorem-ipsum.org, 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]=-l, fullName[3]=-lo, id[0]=-141, id  
[1]=-142, id[2]=-143, id[3]=-144, passportNumber[0]=FKJZAG0XN, passportNumber[1]=FKJZAG0XN, passportNumber[2]=FKJZAG0XN, passportNumber[3]=FKJZAG0XN,  
service=739, specialNeeds[0]-Lorem ipsum dolor sit ame, specialNeeds[1]-Lorem ipsum dolor sit ame, specialNeeds[2]-Lorem ipsum dolor sit ame, specialNeeds  
[3]-Lorem ipsum dolor sit ame, version[0]~=0, version[1]~=0, version[2]~=0, version[3]~=0)', but got '{\"bookingId\":110,\"draftMode\":false,\"email\":[0]  
acme@lorem-  
ipsum.dolor.sit.amet,\"consectetur.adipiscing.elit.Sed.do.eiusmod.tempor.incididunt.ut.labore.et.dolore.magna.aliqua.Ut.enim.ad.minim veniam quis nostrud exerciti  
onum.uillamo.laboris.nisi.ut.aliquip.ex ea.commodo.consequat Duis aute irure doloris.eu\", email:[13]-lorem-  
ipsum.dolor.sit.amet,\"consectetur.adipiscing.elit.Sed.do.eiusmod.tempor.incididunt.ut.labore.et.dolore.magna.aliqua.Ut.enim.ad.minim veniam quis nostrud exerciti  
onum.uillamo.laboris.nisi.ut.aliquip.ex ea.commodo.consequat Duis aute irure doloris.eu.org\", email:[14]=acme@lorem-ipsum.org, email:[15]=acme@lorem-ipsum.org,  
email:[16]=acme@lorem-ipsum.org, email:[17]=acme@lorem-ipsum.org, email:[18]=acme@lorem-ipsum.org, email:[19]=acme@lorem-ipsum.org, email:[1]=acme@lorem-ipsum.org,  
email:[20]=acme@lorem-ipsum.org, email:[21]=acme@lorem-ipsum.org, email:[22]=acme@lorem-ipsum.org, email:[23]=acme@lorem-ipsum.org, email:[24]=acme@lorem-  
ipsum.org, email:[25]=acme@lorem-ipsum.org, email:[26]=acme@lorem-ipsum.org, email:[27]=acme@lorem-ipsum.org, email:[28]=acme@lorem-ipsum.org, email  
:[29]=acme@lorem-ipsum.org, email:[2]=acme@lorem-ipsum.org, email:[30]=acme@lorem-ipsum.org, email:[31]=acme@lorem-ipsum.org, email:[32]=acme@lorem-ipsum.org,  
email:[33]=acme@lorem-ipsum.org, email:[34]=acme@lorem-ipsum.org, email:[3]=acme@lorem-ipsum.org, email:[4]=acme@lorem-ipsum.org, email:[5]=acme@lorem-ipsum.org,  
email:[6]=acme@lorem-ipsum.org, email:[7]=acme@lorem-ipsum.org, email:[8]=acme@lorem-ipsum.org, email:[9]=a@a-e, fullName[0]-Lorem ipsum dolor sit amet, consect  
etur adipiscing elit. Sed do eiusmod tempor incididunt ut labore et dolore magna aliqua! Ut*, fullName[10]-Lorem ipsum dolor sit amet, consectetur  
adipiscing elit. Sed do eiusmod tempor incididunt ut labore et dolore magna aliqua! Ut*, fullName[11]-Lorem ipsum dolor sit amet, consectetur adipiscing  
elit. Sed do eiusmod tempor incididunt ut labore et dolore magna aliqua! Ut*, fullName[12]-Lorem ipsum dolor sit amet, consectetur adipiscing elit. Sed do:  
eiusmod tempus incididunt ut labore et dolore magna aliqua! Ut*, fullName[13]-Lorem ipsum dolor sit amet, consectetur adipiscing elit. Sed do eiusmod tempus  
incididunt ut labore et dolore magna aliqua! Ut*, fullName[14]-Lorem ipsum dolor sit amet, consectetur adipiscing elit. Sed do eiusmod tempus incididunt ut labore  
et dolore magna aliqua! Ut*, fullName[15]-Lorem ipsum dolor sit amet, consectetur adipiscing elit. Sed do eiusmod tempus incididunt ut labore et  
dolore magna aliqua! Ut*, fullName[16]-Lorem ipsum dolor sit amet, consectetur adipiscing elit. Sed do eiusmod tempus incididunt ut labore et dolore magna  
aliqua! Ut*, fullName[17]-Lorem ipsum dolor sit amet, consectetur adipiscing elit. Sed do eiusmod tempus incididunt ut labore et dolore magna aliqua! Ut*,  
fullName[18]-Lorem ipsum dolor sit amet, consectetur adipiscing elit. Sed do eiusmod tempus incididunt ut labore et dolore magna aliqua! Ut*, fullName[19]-  
Lorem ipsum dolor sit amet, consectetur adipiscing elit. Sed do eiusmod tempus incididunt ut labore et dolore magna aliqua! Ut*, fullName[20]-Lorem ipsum dolor  
sit amet, consectetur adipiscing elit. Sed do eiusmod tempus incididunt ut labore et dolore magna aliqua! Ut*, fullName[21]-Lorem ipsum dolor sit amet,  
consectetur adipiscing elit. Sed do eiusmod tempus incididunt ut labore et dolore magna aliqua! Ut*, fullName[22]-Lorem ipsum dolor sit amet, consectetur  
adipiscing elit. Sed do eiusmod tempus incididunt ut labore et dolore magna aliqua! Ut*..fullName[23]-Lorem ipsum dolor sit amet, consectetur ad
```

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.