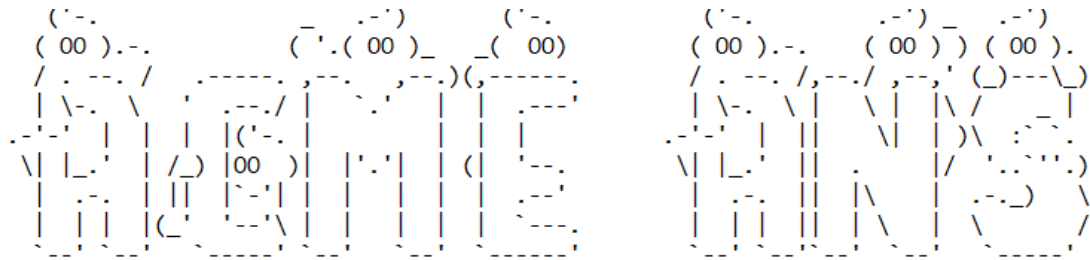


REPORTE DE TESTING Y MUTACIONES DEL CÓDIGO

Acme-ANS-D04



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

Creado por el grupo C1.022, del G1

Participantes	
Nombres	Correos
Benito Merchán, Manuel Jesús	manbenmer1@alum.us.es
Fernández Noguerol, Francisco	frafernog@alum.us.es
Gutiérrez Arazo, Beatriz	beagutara@alum.us.es
Varo Vera, Juan	juavarver@alum.us.es

18 de Mayo de 2025

Índice

Portada.....	1
Índice.....	2
Resumen ejecutivo.....	3
Tabla de revisiones.....	4
Introducción.....	5
Functional testing.....	6
Performance testing.....	8
Mutaciones en los aeropuertos.....	11
Conclusiones.....	16
Bibliografía.....	17

Resumen ejecutivo

Este es el proyecto del grupo C1.022 sobre Acme AirNav Solutions, S.A. (Acme ANS, S.A. abreviado), la cual es una compañía ficticia especializada en ayudar a aeropuertos a organizar y coordinar sus operaciones a partir de soluciones desarrolladas en software. La logística de los vuelos (la programación de los vuelos, la organización de reservas y de tripulación, etc.) se gestionan mediante el desarrollo de un WIS.

Con esto, para el correcto funcionamiento de la aplicación, se deberá escribir un reporte de testing donde se describan los valores de cobertura del código junto con los valores correspondientes al rendimiento de la aplicación, ambos, previos y posteriores a realizar mutaciones en el código.

Tabla de revisiones

Número	Fecha	Descripción
v1.0.0	18/05/2025	Versión inicial terminada del reporte

Introducción

En este reporte quedarán expuestos los resultados de los tests realizados durante el entregable D04 sobre las funcionalidades desarrolladas en grupo que tengan asociadas un requisito de conjunto de pruebas, por lo que se tendrá en cuenta tanto los conjuntos de pruebas realizados sobre el requisito 11 como del requisito 39.

Se reportarán 2 tipos de resultados, los relacionados con el testeo funcional y los relacionados con el testeo del rendimiento. A su vez también se reportarán las 5 mutaciones realizadas sobre el código y sus distintos efectos.

Este reporte está organizado de la siguiente forma:

1. **Resumen ejecutivo:** Introducción breve sobre el reporte.
2. **Tabla de revisiones:** Historial de revisiones del documento.
3. **Introducción:** Contextualización de la planificación y progreso además de su importancia.
4. **Funcional testing:** Resultados obtenidos de la fase de testing y posibles bugs encontrados.
5. **Performance testing:** Resultados obtenidos de analizar el rendimiento de los test en 2 ordenadores.
6. **Mutaciones en los aeropuertos:** Resultados de introducir bugs intencionales en el código.
7. **Conclusiones:** Resumen de los hallazgos y la importancia de este reporte.
8. **Bibliografía:** Fuentes consultadas durante la investigación.

Functional testing

- Resultados sobre el testeo en los aeropuertos:

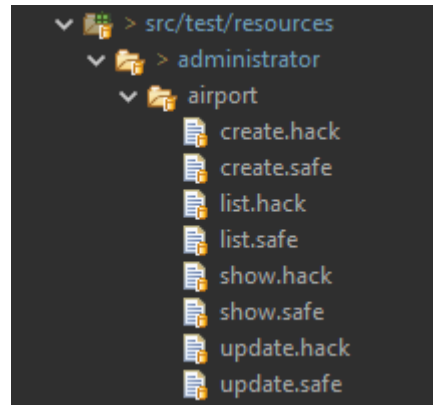


Figura 1: Archivos de testing para los aeropuertos

No fue detectado ningún bug durante los tests.

acme.features.administrator.airport	100,0 %	524	0	524
AdministratorAirportController.java	100,0 %	24	0	24
AdministratorAirportCreateService.java	100,0 %	164	0	164
AdministratorAirportListService.java	100,0 %	64	0	64
AdministratorAirportShowService.java	100,0 %	83	0	83
AdministratorAirportUpdateService.java	100,0 %	189	0	189

Figura 2: Cobertura de los tests en los aeropuertos

Se ha logrado alcanzar una cobertura total de la feature, cubriendo todas las instrucciones y posibles ramas/paths del código en cada servicio.

- Resultados sobre el testeo en los cambios de moneda:

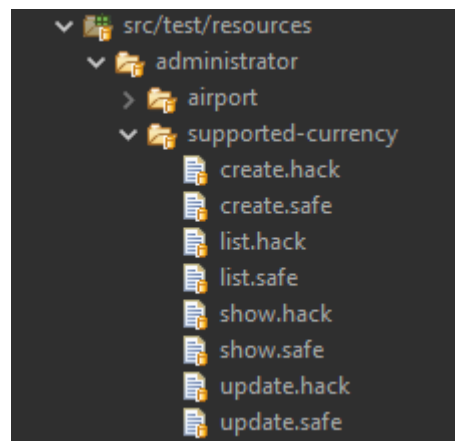
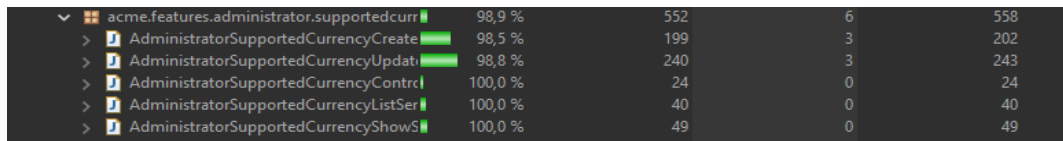


Figura 3: Archivos de testing para los cambios de moneda

Se ha logrado alcanzar una cobertura casi total de la feature, no se ha alcanzado el 100% debido a la necesidad de mockear el servicio, pero el resto de instrucciones se encuentran testeadas correctamente.



Path	Coverage	Instructions	Branches	Lines
acme.features.administrator.supportedcurr	98,9 %	552	6	558
AdministratorSupportedCurrencyCreate	98,5 %	199	3	202
AdministratorSupportedCurrencyUpdate	98,8 %	240	3	243
AdministratorSupportedCurrencyControl	100,0 %	24	0	24
AdministratorSupportedCurrencyListService	100,0 %	40	0	40
AdministratorSupportedCurrencyShowService	100,0 %	49	0	49

Figura 4: Cobertura de los tests en los cambios de moneda

```
@Override
public void validate(final SupportedCurrency supportedCurrency) {

    List<SupportedCurrency> supportedCurrencies = this.scr.findAllSupportedCurrencies();
    List<String> currencyNames = supportedCurrencies.stream().filter(sp -> sp.getId() != supportedCurrency.getId()).map(sp -> sp.getCurrencyName()).toList();

    super.state(!(this.scr.findDefaultSupportedCurrency().getId() == supportedCurrency.getId() && !supportedCurrency.getIsDefaultCurrency()), "isDefaultCurrency", "administrator.supported-currency");

    if (supportedCurrency.getCurrencyName() != null)
        super.state(!currencyNames.contains(supportedCurrency.getCurrencyName()), "currencyName", "administrator.supported-currency.create.already-exists-currency");
}
```

Figura 5: Camino de una condición if que no ejecuta nunca

Como se puede observar, aunque se haya alcanzado la total cobertura de instrucciones, la cobertura de ramas no era total ni en el servicio de Update ni en el de Create por la misma razón, pero gracias a esto, hemos detectado que se podría eliminar la condición ya que en ningún momento ha sido posible que esa condición no se cumpliera por razones que desconocemos, por lo que la solución ha sido eliminarla al ser inservible.

```
@Override
public void validate(final SupportedCurrency supportedCurrency) {

    List<SupportedCurrency> supportedCurrencies = this.scr.findAllSupportedCurrencies();
    List<String> currencyNames = supportedCurrencies.stream().filter(sp -> sp.getId() != supportedCurrency.getId()).map(sp -> sp.getCurrencyName()).toList();

    super.state(!(this.scr.findDefaultSupportedCurrency().getId() == supportedCurrency.getId() && !supportedCurrency.getIsDefaultCurrency()), "isDefaultCurrency", "ad");

    super.state(!currencyNames.contains(supportedCurrency.getCurrencyName()), "currencyName", "administrator.supported-currency.create.already-exists-currency");
}
```

Figura 6: Eliminación de la condición if

Performance testing

- Resultados en ordenador 1:

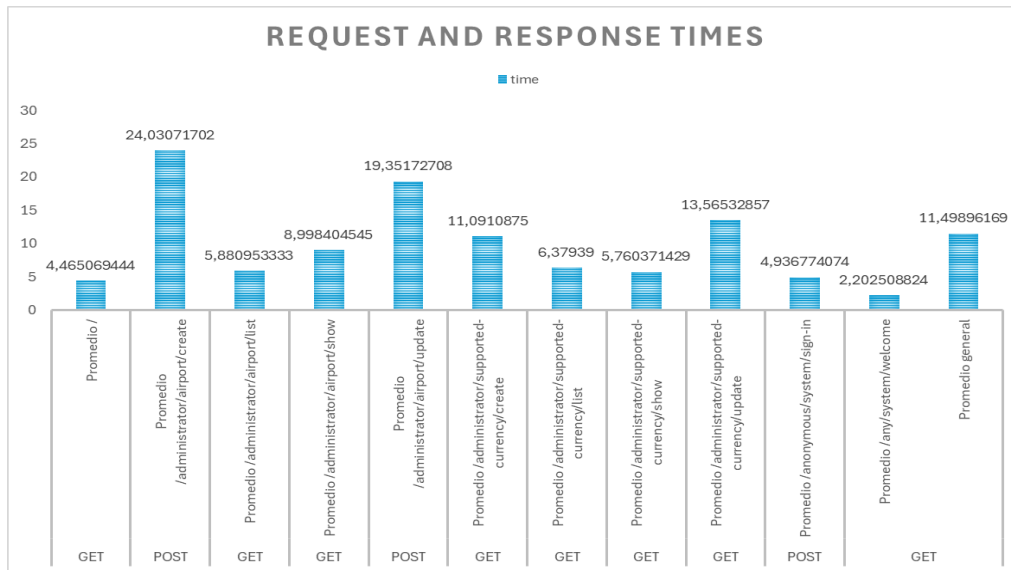


Figura 1: Tiempos para Ordenador 1 con índices

En esta tabla podemos observar los tiempos promedios de las distintas peticiones realizadas tanto en la parte de aeropuertos como en la parte de los cambios de moneda por el ordenador 1.

Solo la parte correspondiente a los aeropuertos, gracias a una cuenta rápida, nos da un promedio de 14.56545049 ya que conforma la parte más amplia de testear entre estas 2 aquí incluídas, del que gran parte se debe a la amplia variedad de peticiones distintas a probar tanto en la creación como en la actualización de los aeropuertos. Si juntamos esta parte de aeropuertos con el resto de peticiones que no son las relacionadas con la parte de las monedas nos da un promedio de ~9.823593474, un tiempo ligeramente menor a 0.01 segundos.

Nota: Tras un rápido replay de los tests sin los índices se ha decidido no aportar los valores estadísticos debido a que la diferencia de tiempo es totalmente mínima y despreciable ya que solo se cuenta con un índice en total entre las dos entidades a testear por lo que creemos que es innecesaria la comparación de esto.

- Resultados del ordenador 2:

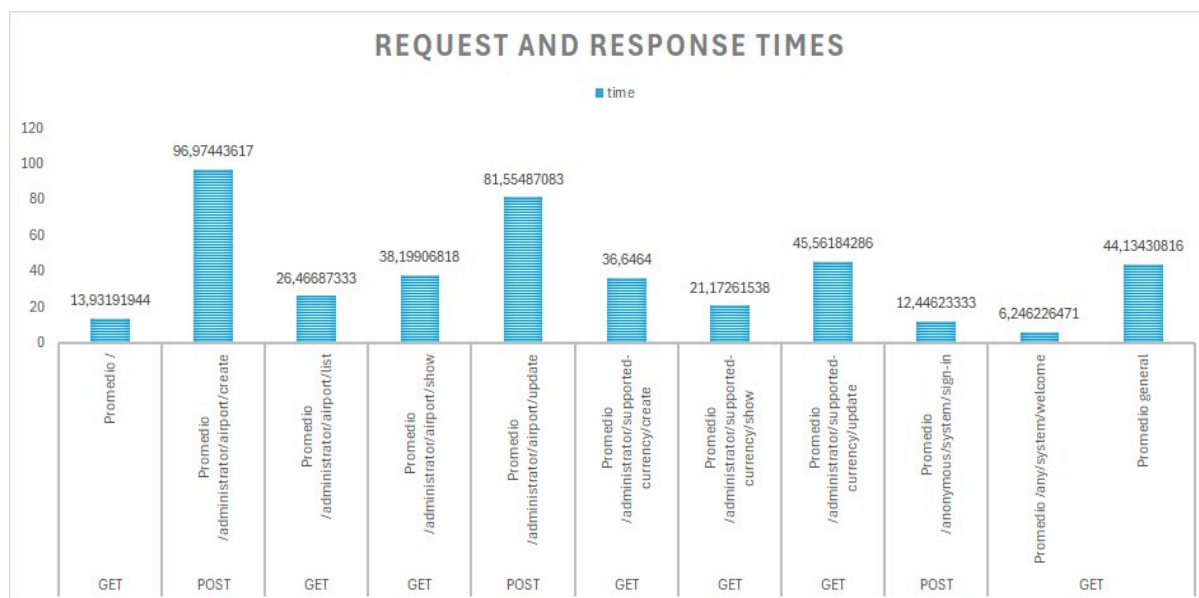


Figura 2: Tiempos para Ordenador 2 con índices

En esta tabla podemos observar los tiempos promedios de las distintas peticiones realizadas tanto en la parte de aeropuertos como en la parte de los cambios de moneda por el ordenador 2.

Cabe destacar que estos resultados fueron obtenidos con un ordenador de considerable menor potencia, por eso se puede ver el grandísimo aumento de tiempos respecto al ordenador 1. Aun así se sigue pudiendo apreciar que las operaciones de creación y actualización son las que mayor tiempo han consumido y que teniendo en cuenta la diferencia de potencia, los tiempos no han variado en proporción.

Ordenador 1		Ordenador 2	
Media	11,49896169	Media	44,13430816
Error típico	0,674862454	Error típico	2,591775554
Mediana	6,0432	Mediana	21,9314
Desviación estándar	10,90273661	Desviación estándar	43,52331269
Varianza de la muestra	118,8696657	Varianza de la muestra	1894,278748
Curtosis	4,19971135	Curtosis	2,021407474
Coefficiente de asimetría	1,616240819	Coefficiente de asimetría	1,306222137
Rango	70,3963	Rango	250,7932
Mínimo	1,1783	Mínimo	3,697
Máximo	71,5746	Máximo	254,4902
Suma	3001,229	Suma	12445,8749
Cuenta	261	Cuenta	282
Nivel de confianza(95,0%)	1,328891907	Nivel de confianza(95,0%)	5,101760094
Prueba z para medias de dos muestras			
	Ordenador 1	Ordenador 2	
Media	11,49896169	44,13430816	
Varianza (conocida)	118,8696657	1894,278748	
Observaciones	261	282	
Diferencia hipotética de las medias	0		
z	-12,18556548		
P(Z<=z) una cola	0		
Valor crítico de z (una cola)	1,644853627		
Valor crítico de z (dos colas)	0		
Valor crítico de z (dos colas)	1,959963985		

Figura 3: Valores estadísticos entre Ordenador 1 y Ordenador 2 con índices

Como se puede apreciar, hay una diferencia muy significativa en los tiempos de testeo, esto se debe principalmente a la gran diferencia de hardware entre ambos ordenadores. El ordenador 1 es un ordenador de gama alta, gaming y que se encontraba cargando en esos momentos, en cambio el ordenador 2, es un ordenador de gama media, de uso cotidiano y no estaba conectado en esos momentos. Es por esto que el análisis-z puede sugerir unos valores extraños.

Mutaciones en los aeropuertos

- Mutación 1, eliminar el authorise en AdministratorAirportCreateService

```
@Override
public void authorise() {
    super.getResponse().setAuthorised(super.getRequest().getPrincipal().hasRealmOfType(Administrator.class));
}
```

Figura 1: Estado inicial del código

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

@Override
public void load() {
    Airport airport;

    airport = new Airport();

    super.getBuffer().addData(airport);
}

@Override
public void bind(final Airport airport) {
    super.bindObject(airport, "name", "iATACode", "operationalScope", "city", "country", "website", "email", "contactNumber");
}

@Override
public void validate(final Airport object) {
    List<Airport> airports = this.aar.findAllAirports();
    List<String> airportCodes = airports.stream().map(Airport::getIATACode).toList();

    boolean confirmation = super.getRequest().getData("confirm", boolean.class);
    super.state(confirmation, "confirm", "acme.validation.confirmation.message");

    super.state(!airportCodes.contains(object.getIATACode()), "iATACode", "administrator.airport.create.already-exists");
}

@Override
public void perform(final Airport airport) {
    this.aar.save(airport);
}
```

Figura 2: Mutación 1 y efecto

```
FAILED GET /administrator/airport/create (request-id="da5fda4f-2e24-48ba-9640-2b5d74a0ab16", input=""): Expected 'status' to be '200', but got '500'. Expected 'payload' to be '{confirm=false, id=0, operationalScope=[{"key":"0", "label":"----", "selected":true, "sealed":true}, {"key":"INTERNATIONAL", "label":"INTERNATIONAL", "selected":false, "sealed":true}, {"key":"DOMESTIC", "label":"DOMESTIC", "selected":false, "sealed":true}, {"key":"REGIONAL", "label":"REGIONAL", "selected":false, "sealed":true}], service=159, version=0}', but got '{service=159}'.
FAILED POST /administrator/airport/create (request-id="46f96c07-27f5-43c5-8da5-faa3b86aa9c6", input="id=0&version=0&name=&iATACode=&operationalScope=0&city=&country=&website=&email=&contactNumber=&confirm=false"): Expected 'status' to be '200', but got '500'. Expected 'payload' to be '{city=, city$error=May not be null., confirm=false, confirm$error=Must confirm this., contactNumber=, country=, country$error=May not be null., email=, iATACode=, iATACode$error=May not be null., id=0, name=, name$error=May not be null., operationalScope=[{"key":"0", "label":"----", "selected":true, "sealed":true}, {"key":"INTERNATIONAL", "label":"INTERNATIONAL", "selected":false, "sealed":true}, {"key":"DOMESTIC", "label":"DOMESTIC", "selected":false, "sealed":true}, {"key":"REGIONAL", "label":"REGIONAL", "selected":false, "sealed":true}], operationalScope$error=May not be null., service=159, version=0, website=}', but got '{service=159}'.
```

Figura 3: Fallo detectado

El sistema no permite realizar ningún tipo de operación create ya que se le deniega el acceso con un 500 automáticamente.

- Mutación 2, cambiar los rangos máximos de strings en ciertos atributos

```

@Mandatory
@ValidString(max = 50)
@Automapped
private String          name;

@Mandatory
@ValidString(min = 3, max = 3, pattern = "[A-Z]{3}")
@Column(unique = true)
private String          iATACode;

@Mandatory
@Enumerated(EnumType.STRING)
@Automapped
private OperationalScope operationalScope;

@Mandatory
@ValidString(max = 50)
@Automapped
private String          city;

@Mandatory
@ValidString(max = 50)
@Automapped
private String          country;

```

Figura 4: Estado inicial del código

```

@Mandatory
@ValidString(max = 49)
@Automapped
private String          name;

@Mandatory
@ValidString(min = 3, max = 3, pattern = "[A-Z]{3}")
@Column(unique = true)
private String          iATACode;

@Mandatory
@Enumerated(EnumType.STRING)
@Automapped
private OperationalScope operationalScope;

@Mandatory
@ValidString(max = 49)
@Automapped
private String          city;

@Mandatory
@ValidString(max = 49)
@Automapped
private String          country;

```

Figura 5: Mutación 2

```

Resetting application (clear schema, populate sample, reset clock, reset randomiser).
ERROR Validating 'airport-06' ... FAILED. 'name': Length must be between 0 and 49. 'city': Length must be between 0 and 49. 'country': Length must be between 0 and 49.
Replaying .\src\test\resources\administrator\airport\create.hack...
Loaded 34 requests from .\src\test\resources\administrator\airport\list.hack
Resetting application (clear schema, populate sample, reset clock, reset randomiser).
ERROR Validating 'airport-06' ... FAILED. 'city': Length must be between 0 and 49. 'country': Length must be between 0 and 49. 'name': Length must be between 0 and 49.
Replaying .\src\test\resources\administrator\airport\list.hack...
Loaded 154 requests from .\src\test\resources\administrator\airport\show.hack
Resetting application (clear schema, populate sample, reset clock, reset randomiser).
ERROR Validating 'airport-06' ... FAILED. 'name': Length must be between 0 and 49. 'country': Length must be between 0 and 49. 'city': Length must be between 0 and 49.

```

Figura 6: Fallo detectado

- Mutación 3, cambiar condición de un validador en AdministratorAirportCreateService:

Se espera que la operación de confirmación funcione, pero al confirmarse, se provoca un fallo.

```
@Override
public void validate(final Airport object) {

    List<Airport> airports = this.aar.findAllAirports();
    List<String> airportCodes = airports.stream().map(Airport::getIATACode).toList();

    boolean confirmation = super.getRequest().getData("confirm", boolean.class);
    super.state(confirmation, "confirm", "acme.validation.confirmation.message");

    super.state(!airportCodes.contains(object.getIATACode()), "iATACode", "administrator.airport.create.already-exists");

}
```

Figura 7: Estado inicial del código

```
@Override
public void validate(final Airport object) {

    List<Airport> airports = this.aar.findAllAirports();
    List<String> airportCodes = airports.stream().map(Airport::getIATACode).toList();

    boolean confirmation = super.getRequest().getData("confirm", boolean.class);
    super.state(confirmation, "confirm", "acme.validation.confirmation.message");

    super.state(!airportCodes.contains(object.getIATACode()), "iATACode", "administrator.airport.create.already-exists");

}
```

Figura 8: Mutación 3

```
FAILED POST /administrator/airport/create (request-id="544dc392-84b1-4e4f-aea0-446adb6f41b4", input="id=0&version=0&name=Aeropuerto
+creado&iATACode=ABC&operationalScope=INTERNATIONAL&city=Ciudad&country=Pa%C3%ADs&website=https%3A%2F%2Fwww.website.com&email=testing
%40hotmail.com&contactNumber=12345678901234&confirm=true"): Expected 'payload' to be '{service=160}', but got '{city=Ciudad, confirm=false,
confirm$error=Must confirm this., contactNumber=12345678901234, country=País, email=testing@hotmail.com, iATACode=ABC, iATACode$error=this IATA
code already exists., id=0, name=Aeropuerto creado, operationalScope=[{"key":"0","label":"---","selected":false,"sealed":true},
{"key":"INTERNATIONAL","label":"INTERNATIONAL","selected":true,"sealed":true},
{"key":"DOMESTIC","label":"DOMESTIC","selected":false,"sealed":true}, {"key":"REGIONAL","label":"REGIONAL","selected":false,"sealed":true}],
service=160, version=0, website=https://www.website.com}'.
Loaded 66 requests from .\src\test\resources\administrator\airport\list.safe
Resetting application (clear schema, populate sample, reset clock, reset randomiser).
Replaying .\src\test\resources\administrator\airport\list.safe...
Loaded 330 requests from .\src\test\resources\administrator\airport\show.safe
```

Figura 9: Fallo detectado

- Mutación 4, eliminar una validación en AdministratorAirportCreateService:

```
@Override
public void validate(final Airport object) {

    List<Airport> airports = this.aar.findAllAirports();
    List<String> airportCodes = airports.stream().map(Airport::getIATACode).toList();

    boolean confirmation = super.getRequest().getData("confirm", boolean.class);
    super.state(confirmation, "confirm", "acme.validation.confirmation.message");

    super.state(!airportCodes.contains(object.getIATACode()), "iATACode", "administrator.airport.create.already-exists");

}
```

Figura 10: Estado inicial del código

```

@Override
public void validate(final Airport object) {

    List<Airport> airports = this.aar.findAllAirports();
    List<String> airportCodes = airports.stream().map(Airport::getIATACode).toList();

    boolean confirmation = super.getRequest().getData("confirm", boolean.class);
    super.state(confirmation, "confirm", "acme.validation.confirmation.message");|
}

```

Figura 11: Mutación 4

```

FAILED POST /administrator/airport/create (request-id="80d4fae2-a62c-49f5-a532-3f9b653ad8bd",
input="id=0&version=0&name=&iATACode=BCN&operationalScope=0&city=&country=&website=&email=&contactNumber=&confirm=false"): Expected 'payload' to
be '{city=, city$error=May not be null., confirm=false, confirm$error=Must confirm this., contactNumber=, country=, country$error=May not be
null., email=, iATACode=BCN, iATACode$error=This IATA code already exists., id=0, name=, name$error=May not be null., operationalScope=
[{"key":"0","label":"----","selected":true,"sealed":true},{key:"INTERNATIONAL","label":"INTERNATIONAL","selected":false,"sealed":true},
{"key":"DOMESTIC","label":"DOMESTIC","selected":false,"sealed":true},{key:"REGIONAL","label":"REGIONAL","selected":false,"sealed":true}],
operationalScope$error=May not be null., service=158, version=0, website=}', but got '{city=, city$error=May not be null., confirm=false, confirm
$error=Must confirm this., contactNumber=, country=, country$error=May not be null., email=, iATACode=BCN, id=0, name=, name$error=May not be
null., operationalScope=[{"key":"0","label":"----","selected":true,"sealed":true},
{"key":"INTERNATIONAL","label":"INTERNATIONAL","selected":false,"sealed":true},
{"key":"DOMESTIC","label":"DOMESTIC","selected":false,"sealed":true},{key:"REGIONAL","label":"REGIONAL","selected":false,"sealed":true}],

```

Figura 12: Fallo detectado

El sistema permite IATACodes duplicados gracias a este cambio.

- Mutación 5, cambiar la obligatoriedad de ciertos atributos:

```
@Optional
@ValidUrl
@Automapped
private String website;

@Optional
@ValidEmail
@Automapped
private String email;

@Optional
@ValidString(pattern = "^\\+?\\d{6,15}$")
@Automapped
private String contactNumber;
```

Figura 13: Estado inicial del código

```
@Mandatory
@ValidUrl
@Automapped
private String website;

@Mandatory
@ValidEmail
@Automapped
private String email;

@Mandatory
@ValidString(pattern = "^\\+?\\d{6,15}$")
@Automapped
private String contactNumber;
```

Figura 14: Mutación 5

```
FAILED POST /administrator/airport/create (request-id="95657d49-6663-428f-9c04-fc0edb1372b5", input="id=0&version=0&name=&iATACode=%D5%B4%E1%83%9D%
%E5%A4%A9%E5%AF%AB.%C3%81AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA&operationalScope=0&city=&country=&website=&email=&contactNumber=&confirm=false"): Expected
'payload' to be '{city=, city$error=May not be null., confirm=false, confirm$error=Must confirm this., contactNumber=, country=, country$error=May
not be null., email=, iATACode=天爲.AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA, iATACode$error=Length must be between 3 and 3., id=0, name=,
name$error=May not be null., operationalScope=[{"key":"0","label":"----","selected":true,"sealed":true},
{"key":"INTERNATIONAL","label":"INTERNATIONAL","selected":false,"sealed":true},
{"key":"DOMESTIC","label":"DOMESTIC","selected":false,"sealed":true},{key":"REGIONAL","label":"REGIONAL","selected":false,"sealed":true}],
operationalScope$error=May not be null., service=159, version=0, website=}', but got '{city=, city$error=May not be null., confirm=false, confirm
$error=Must confirm this., contactNumber=, contactNumber$error=May not be null., country=, country$error=May not be null., email=, email$error=May
not be null., iATACode=天爲.AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA, iATACode$error=Length must be between 3 and 3., id=0, name=, name
$error=May not be null., operationalScope=[{"key":"0","label":"----","selected":true,"sealed":true},
{"key":"INTERNATIONAL","label":"INTERNATIONAL","selected":false,"sealed":true},
{"key":"DOMESTIC","label":"DOMESTIC","selected":false,"sealed":true},{key":"REGIONAL","label":"REGIONAL","selected":false,"sealed":true}],
operationalScope$error=May not be null., service=159, version=0, website=, website$error=May not be null.}'.
```

Figura 15: Fallo detectado

Conclusiones

Como conclusión de este reporte remarcar que debido a la exhaustiva búsqueda de fallos antes de realizar los tests, estos no fueron capaces de encontrar ningún bug y pasaron perfectamente, por otro lado es claro que la velocidad de ejecución de los tests depende en gran medida del hardware con el que se ejecuten por lo que sería injusto una comparación entre ambos ordenadores.

Bibliografía

Intencionalmente en blanco