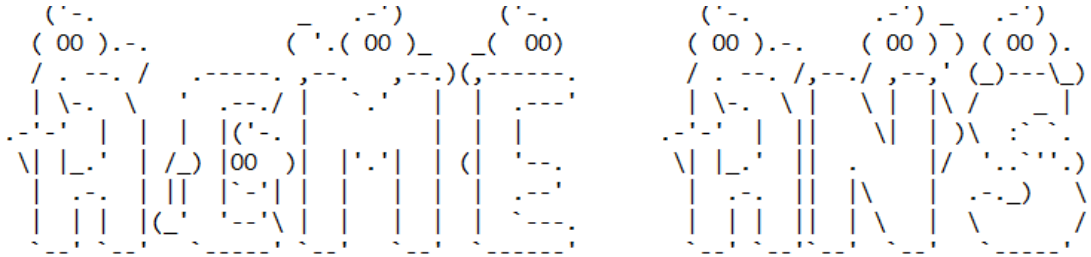


# REPORTE DE TESTING

Acme-ANS-D04



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

Creado por el grupo C1.022, del G1

Participantes	
Nombres	Correos
Fernández Noguero, Francisco	frafernog@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.....	10
Mutaciones en los requisitos 8, 9 y 29.....	15
Conclusiones.....	19
Bibliografía.....	20

## **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 requisitos 8, 9 y 29:** 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 vuelos y sus tramos:

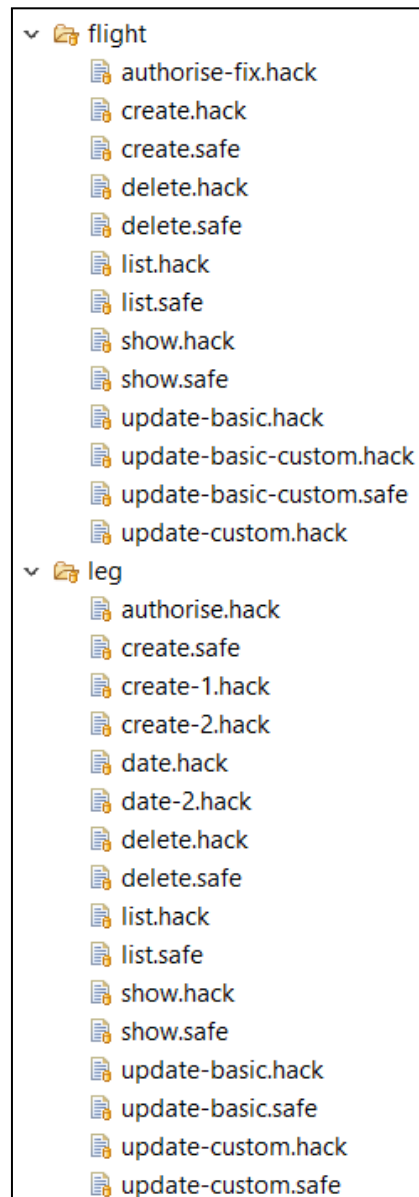


Figura 1: Archivos de testing para vuelos y sus tramos

No fue detectado ningún bug durante los tests.





Element	Coverage	Covered Instru...	Missed Instruct...	Total Instructio...
>  acme.features.manager.flight	 100,0 %	1.019	0	1.019
>  acme.features.manager.leg	 100,0 %	2.629	0	2.629

Figura 2: Cobertura de los tests de vuelos y sus tramos

Se ha logrado alcanzar un 100% de cobertura con todas las posibles ramas también analizadas.

- Resultados sobre el testeo en la previsión meteorológica:

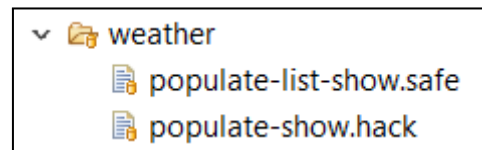


Figura 3: Archivos de testing de la previsión meteorológica

No fue detectado ningún bug durante los tests.

Element	Coverage	Covered Instru...	Missed Instruct...	Total Instructio...
▼  acme.features.any.weather	100,0 %	152	0	152
>  AnyWeatherController.java	100,0 %	14	0	14
>  AnyWeatherListService.java	100,0 %	48	0	48
>  AnyWeatherShowService.java	100,0 %	90	0	90

Figura 4: Cobertura de los tests de previsión meteorológica (features de Any)

Element	Coverage	Covered Instru...	Missed Instruct...	Total Instructio...
▼  acme.features.administrator.weath	66,4 %	99	50	149
>  AdministratorWeatherController	66,4 %	99	50	149

Figura 5: Cobertura de los tests de previsión meteorológica (features de Administrator)

Element	Coverage	Covered Instru...	Missed Instruct...	Total Instructio...
▼  acme.entities.weather	63,8 %	125	71	196
>  WeatherStatus.java	100,0 %	34	0	34
>  Weather.java	56,2 %	91	71	162

Figura 6: Cobertura de los tests de previsión meteorológica (entidad)

No se ha logrado alcanzar un 100% de cobertura debido a que ha habido que mockearlo:

```

protected ModelAndView doPopulate() {
    List<String> cities = this.repository.findAllCities();
    List<Weather> weather;
    if (SpringHelper.isRunningOn("production"))
        weather = cities.stream().map(c -> this.findWeatherOfCity(c)).filter(c -> c != null).toList();
    else
        weather = cities.stream().map(c -> this.findWeatherOfCityMocked(c)).filter(c -> c != null).toList();

    this.repository.saveAll(weather);

    ModelAndView result = new ModelAndView();
    result.setViewName("fragments/welcome");
    result.addObject("globalSuccessMessage", "acme.default.global.message.success");
    return result;
}

protected Weather findWeatherOfCity(final String city) {
    try {
        String apiKey = "2a6d2f4668851aaaa457be9f1ff57775";
        String url = "https://api.openweathermap.org/data/2.5/weather?units=metric" + "&appid=" + apiKey + "&q=" + city;
        RestTemplate api = new RestTemplate();

        ResponseEntity<WeatherPOJO> response = api.getForEntity(url, WeatherPOJO.class);
        Date currentMoment = MomentHelper.getCurrentMoment();
        return Weather.of(response.getBody(), currentMoment, city);
    } catch (final Throwable oops) {
        return null;
    }
}

```

Figura 7: Mockeo del populador

Las líneas que sí hacen la llamada real a la API se quedan sin probar, además de ciertos métodos de la entidad Weather:

```

WeatherStatus result;

if (this.wind >= BAD_WIND || this.rainPerHour >= BAD_RAIN || this.snowPerHour >= BAD_SNOW || this.temperature <= BAD_TEMP_LOW ||
    result = WeatherStatus.BAD_WEATHER;
else if (this.wind >= MID_WIND || this.snowPerHour >= MID_SNOW || this.rainPerHour >= MID_RAIN || this.temperature <= MID_TEMP_L
    result = WeatherStatus.MID_WEATHER;
else
    result = WeatherStatus.GOOD_WEATHER;

return result;
}

// Relationships -----
// Ancillary methods -----

public static Weather of(final WeatherPOJO w, final Date date, final String city) {
    Weather weather = new Weather();
    weather.setCity(city);
    weather.setForecastDate(date);
    weather.setTemperature(w.getMain() != null ? w.getMain().getTemp() : 20);
    weather.setVisibility(w.getVisibility() != null ? w.getVisibility() : 10000);
    weather.setWind(w.getWind() != null ? w.getWind().getSpeed() : 0.);
    weather.setRainPerHour(w.getRain() != null ? w.getRain().getValue() : 0.);
    weather.setSnowPerHour(w.getSnow() != null ? w.getSnow().getValue() : 0.);

    return weather;
}

```

Figura 8: Líneas no cubiertas de Weather

Además, en la feature de listar los vuelos con mal tiempo, se hace uso de un método de la clase Flight, en la cual sus ramas no pueden ser probadas al 100% debido a que hace ciertas comprobaciones con respecto al momento actual, el cual este no se puede probar:



```

@Transient
public Boolean getFlownWithBadWeather() {
    Boolean result = false;
    List<Leg> legs = this.getSortedLegs();
    if (!legs.isEmpty()) {
        result = false;
        FlightRepository repository = SpringHelper.getBean(FlightRepository.class);
        for (Leg l : legs) {
            List<Weather> listOriginWeather = repository.findWeatherByCity(l.getDepartureAirport().getCity());
            List<Weather> listDestinyWeather = repository.findWeatherByCity(l.getArrivalAirport().getCity());
            Comparator<Weather> timeComparatorOrigin = Comparator.comparing((final Weather w) -> (w.getForecastDate().getTime() - l.
            Comparator<Weather> timeComparatorDestiny = Comparator.comparing((final Weather w) -> (w.getForecastDate().getTime() - l.
            Weather originWeather = listOriginWeather.stream().min(timeComparatorOrigin).orElse(null);
            Weather destinyWeather = listDestinyWeather.stream().min(timeComparatorDestiny).orElse(null);
            if (originWeather != null && originWeather.getStatus().equals(WeatherStatus.BAD_WEATHER) || destinyWeather != null && de
                result = true;
                break;
            }
        }
    }
    return result;
}

```

Figura 9: Líneas en amarillo de Flight por no poder cambiar el momento de los tests

## Performance testing

- Resultados en ordenador 1 con índices:

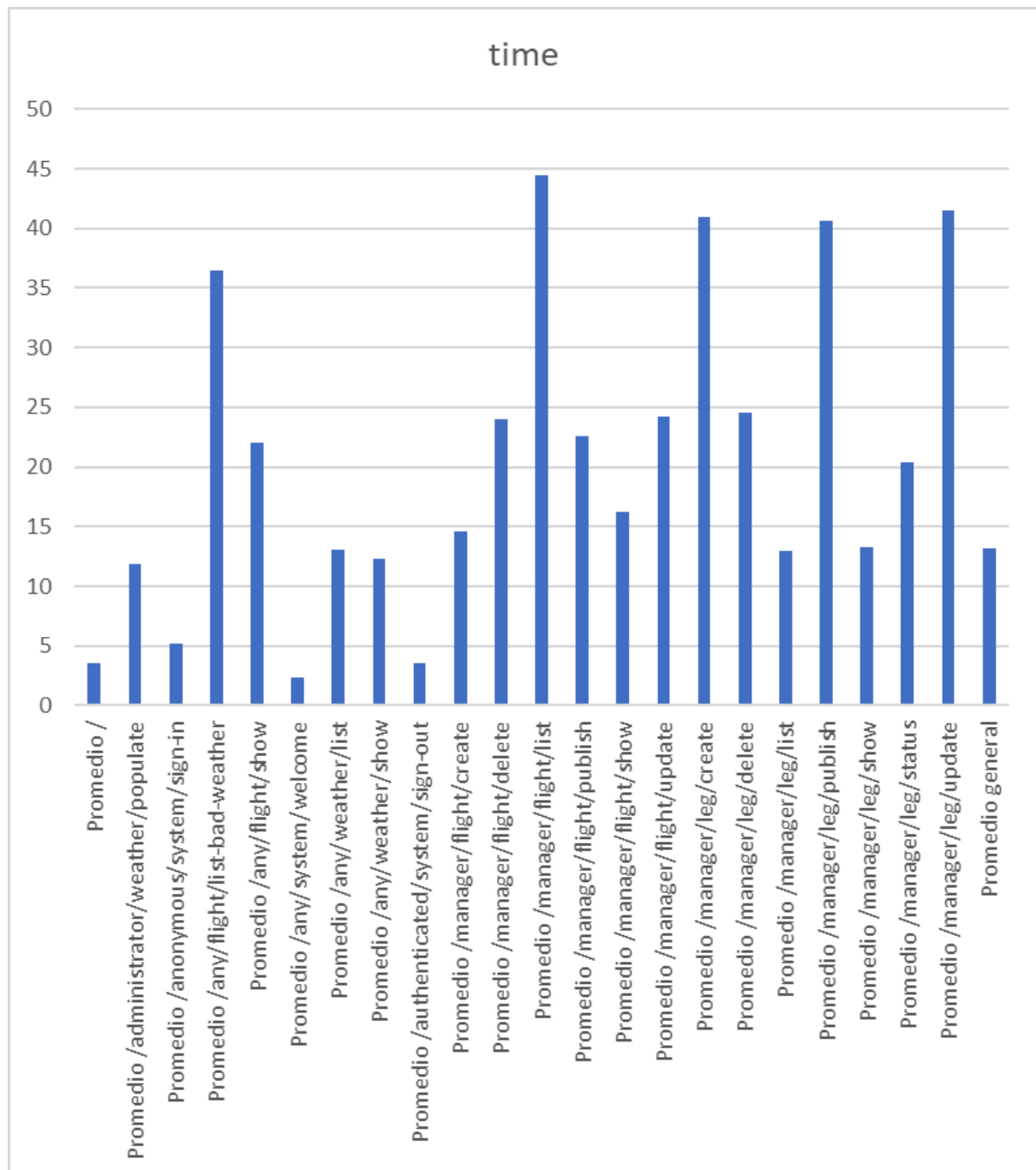


Figura 1: Tiempos para Ordenador 1 con índices

En esta tabla podemos observar los tiempos promedios de las distintas peticiones realizadas tanto en la parte de vuelos, como en la de sus tramos, como en la de previsión meteorológica.

A su vez, se puede observar como las características de modificación de tramos de vuelos y de listados de vuelos son las más costosas en tiempo. Esto es debido, principalmente, a que las restricciones de los tramos de vuelos requieren saber muchos datos sobre todos los

vuelos del sistema, es decir, hace un listado interno de los vuelos del sistema, y como vemos, el listado de los vuelos de por sí ya es bastante costoso, por lo que propicia estos tiempos en la parte de los tramos de vuelos.

En cambio, el resto de peticiones se mantienen en una buena media de tiempo siendo la menos costosa (de entre las características de los requisitos), el poblado de la previsión meteorológica.

- Resultados en ordenador 1 sin índices:

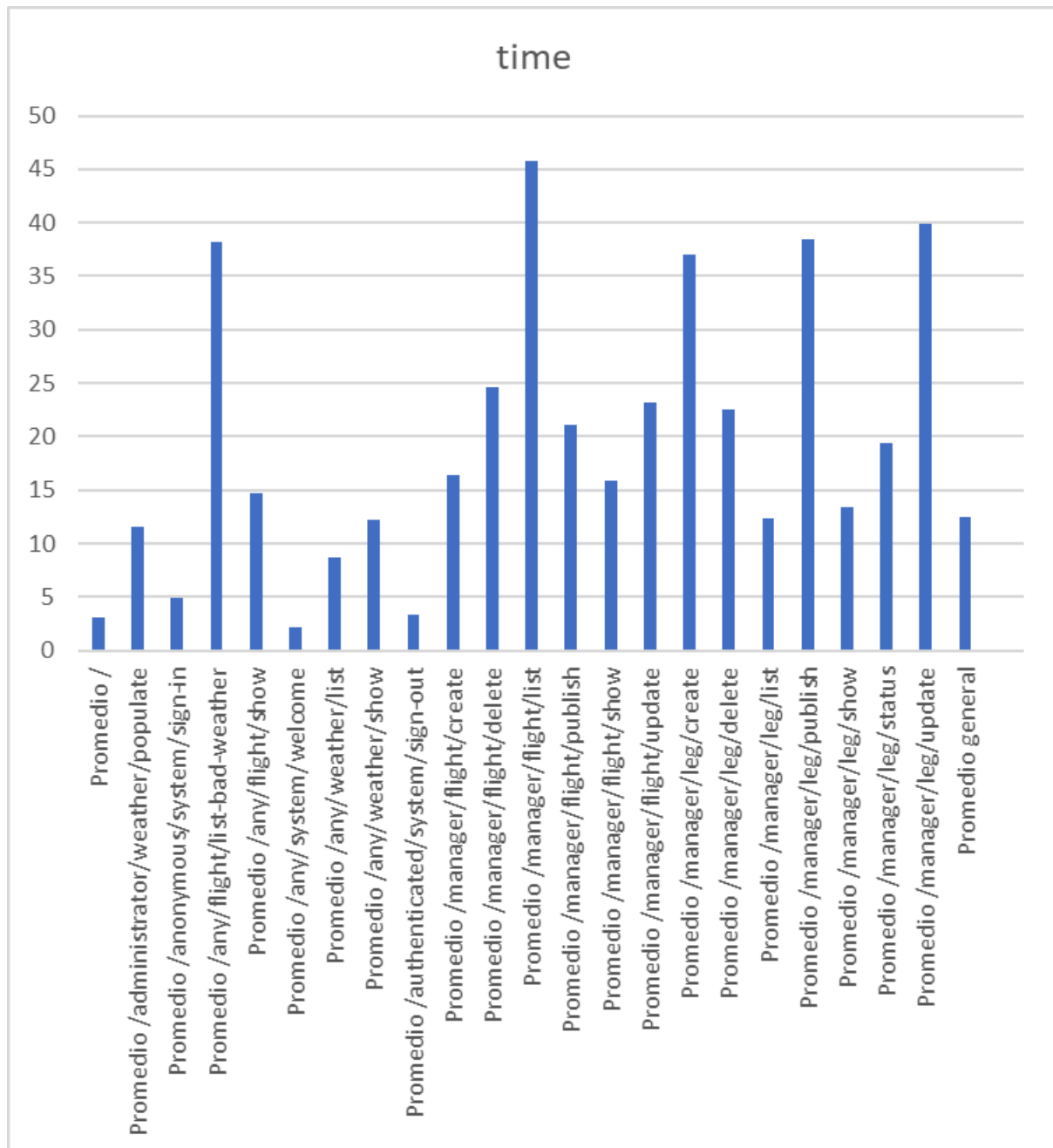


Figura 2: Tiempos para Ordenador 1 sin índices

Como se puede observar, los cambios entre usar o no índices no es muy clara, y esto es debido a que para que se refleje el rendimiento de los índices la cantidad de datos en la base de datos debe ser bastante más grande que los datos que actualmente contiene.

De hecho, en este caso, por diversos factores imposibles de controlar, el tiempo promedio en general ha sido más bajo que con el uso de índices.

Con los datos anteriores, ejecutamos el análisis z:

<i>Before</i>		<i>After</i>	
Media	14,3438549	Media	12,6397108
Error típico	0,93388164	Error típico	0,9356579
Mediana	5,3637	Mediana	4,0965
Moda	1,9236	Moda	1,5673
Desviación estándar	27,4345117	Desviación estándar	27,8505612
Varianza de la muestra	752,65243	Varianza de la muestra	775,653758
Curtosis	271,566464	Curtosis	402,667373
Coefficiente de asimetría	13,0528279	Coefficiente de asimetría	16,9147532
Rango	616,1772	Rango	691,7421
Mínimo	1,1795	Mínimo	1,0852
Máximo	617,3567	Máximo	692,8273
Suma	12378,7468	Suma	11198,7838
Cuenta	886	Cuenta	886
Nivel de confianza(95,0%)	1,83294802	Nivel de confianza(95,0%)	1,83636723
Prueba z para medias de dos muestras			
	<i>Variable 1</i>	<i>Variable 2</i>	
Media	14,3438549	12,6397108	
Varianza (conocida)	752,65243	775,653758	
Observaciones	886	886	
Diferencia hipotética de las medias	0		
z	1,28909959		
P(Z<=z) una cola	0,09868173		
Valor crítico de z (una cola)	1,64485363		
Valor crítico de z (dos colas)	0,19736347		
Valor crítico de z (dos colas)	1,95996398		

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

Con estos datos, podemos ver que el valor P es mayor que alpha, por lo que los cambios no generan una gran diferencia, de hecho, no mejoran nada.

- Resultados en ordenador 2 con índices:

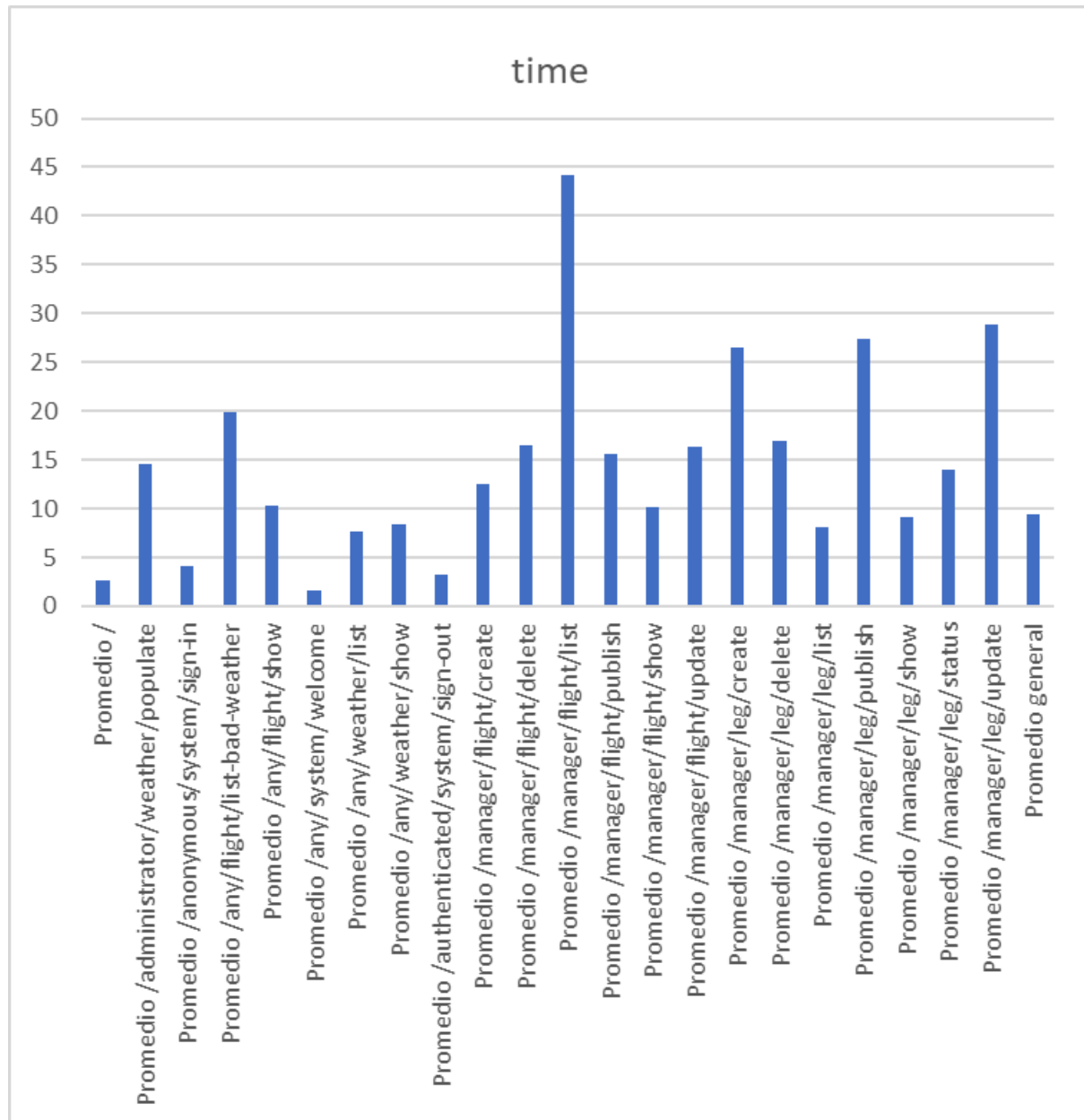


Figura 4: Tiempos para Ordenador 2 con índices

En esta tabla podemos observar los tiempos promedios de las distintas peticiones realizadas ahora por un segundo ordenador (más potente teóricamente), de donde sacamos la misma conclusión que con el ordenador 1, pero esta vez, en una escala menor ya que en general, este ordenador tarda menos en ejecutar las peticiones.

Cabe destacar el impacto que tiene en el rendimiento el listado de los vuelos, el cual sigue siendo el más alto de la tabla con diferencia, pero que aún así, no supone un gran problema.

En comparación con el ordenador 1, el promedio del tiempo en general es bastante inferior, ya que ha sufrido una bajada considerable, pero en correlación con la bajada que han tenido el resto de tiempos de la tabla.

De nuevo, con estos datos, ejecutamos el análisis-z:

<i>Before</i>		<i>After</i>	
Media	14,3438549	Media	9,51277934
Error típico	0,93388164	Error típico	0,93433825
Mediana	5,3637	Mediana	3,653
Moda	1,9236	Moda	1,556
Desviación estándar	27,4345117	Desviación estándar	27,8112807
Varianza de la muestra	752,65243	Varianza de la muestra	773,467334
Curtosis	271,566464	Curtosis	609,284446
Coficiente de asimetría	13,0528279	Coficiente de asimetría	22,6540367
Rango	616,1772	Rango	762,0449
Mínimo	1,1795	Mínimo	0,6758
Máximo	617,3567	Máximo	762,7207
Suma	12378,7468	Suma	8428,32249
Cuenta	886	Cuenta	886
Nivel de confianza(95,0%)	1,83294802	Nivel de confianza(95,0%)	1,83377721
Prueba z para medias de dos muestras			
	<i>Variable 1</i>	<i>Variable 2</i>	
Media	14,3438549	9,51277934	
Varianza (conocida)	752,652431	773,467334	
Observaciones	886	886	
Diferencia hipotética de las medias	0		
z	3,6570494		
P(Z<=z) una cola	0,00012757		
Valor crítico de z (una cola)	1,64485363		
Valor crítico de z (dos colas)	0,00025514		
Valor crítico de z (dos colas)	1,95996398		

Figura 5: Valores estadísticos entre Ordenador 1 y Ordenador 2 con índices

En este caso, como el valor P es mucho inferior a alpha, podemos decir que ha mejorado bastante el rendimiento, pero claro, no hemos hecho modificaciones en el código, sino que hemos ejecutado el mismo código en otro ordenador más potente.

## Mutaciones en los requisitos 8, 9 y 29

- Mutación 1, modificar el rol del authorise en la actualización de vuelos.

Código cambiado:

```
@Override
public void authorise() {
    boolean status;
    int flightId;
    Flight flight;

    flightId = super.getRequest().getData("id", int.class);
    flight = this.fr.findFlightById(flightId);
    status = flight != null && super.getRequest().getPrinci

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

Figura 1: Mutación 1 en el código

Error encontrado tras los tests:

FAILED GET /manager/flight/update?id=60 (request-id="dcfdd8a0-2756-4775-943b-b76145c59e2c", input=""): Expected 'status' to be '500', but got '200'. Expected 'payload' to be '{service=160}', but got '{cost=EUR 1,000.00, description=This is a really long flight, destiny=Barcelona, id=60, isDraftMode=false, needsSelfTransfer=true, numberOfLayovers=1, origin=London, scheduledArrival=2024/12/10 09:00, scheduledDeparture=2024/03/12 06:00, service=160, tag=LONG FLIGHT, version=0}'.

Figura 2: Respuesta de los tests a Mutación 1

- Mutación 2, poner a Optional un atributo obligatorio en la entidad de Leg.

Código cambiado:

```
@ValidString(min = 4, max = 4, pattern = "\\d{4}")
@Optional
@Column(unique = true)
private String uniqueIdentifier;
```

Figura 3: Mutación 2 en el código

## Error encontrado tras los tests:

```
Replaying .\src\test\resources\manager\leg\create-1.hack...
FAILED POST /manager/leg/create (request-id="7be5d67d-7614-43eb-9674-656a4b9bd0f2",
input="id=0&version=0&uniqueIdentifier=&scheduledDeparture=&scheduledArrival=&status=0&departureAirport=0&arrivalAirport=0&aircraft=0&flight=0"): Expected 'payload' to be '{aircraft$error=May not be null., arrivalAirport$error=May not be null., arrivalIATACodes=[{"key":"0","label":"----","selected":true,"sealed":true}, {"key":"35","label":"LHR","selected":false,"sealed":true}, {"key":"36","label":"BCN","selected":false,"sealed":true}, {"key":"37","label":"CDG","selected":false,"sealed":true}, {"key":"38","label":"LAX","selected":false,"sealed":true}, {"key":"39","label":"JFK","selected":false,"sealed":true}, {"key":"40","label":"XXX","selected":false,"sealed":true}, {"key":"41","label":"YYY","selected":false,"sealed":true}, {"key":"42","label":"ZZZ","selected":false,"sealed":true}, {"key":"43","label":"ZZA","selected":false,"sealed":true}], departureAirport$error=May not be null., departureIATACodes=[{"key":"0","label":"----","selected":true,"sealed":true}, {"key":"35","label":"LHR","selected":false,"sealed":true}, {"key":"36","label":"BCN","selected":false,"sealed":true}, {"key":"37","label":"CDG","selected":false,"sealed":true}, {"key":"38","label":"LAX","selected":false,"sealed":true}, {"key":"39","label":"JFK","selected":false,"sealed":true}, {"key":"40","label":"XXX","selected":false,"sealed":true}, {"key":"41","label":"YYY","selected":false,"sealed":true}, {"key":"42","label":"ZZZ","selected":false,"sealed":true}, {"key":"43","label":"ZZA","selected":false,"sealed":true}], flight$error=May not be null., flightChoices=[{"key":"0","label":"----","selected":true,"sealed":true}, {"key":"62","label":"62","selected":false,"sealed":true}, {"key":"63","label":"63","selected":false,"sealed":true}], id=0, isDraftMode=true, registrationNumbers=[{"key":"0","label":"----","selected":true,"sealed":true}, {"key":"68","label":"ABC123-XYZ789-ULR-2025-TEST999","selected":false,"sealed":true}, {"key":"69","label":"DEF456-SAFE-MODEL-2025","selected":false,"sealed":true}], scheduledArrival=, scheduledArrival$error=May not be null., scheduledDeparture=, scheduledDeparture$error=May not be null., service=158, status=0, status$error=May not be null., statuses=[{"key":"0","label":"----","selected":true,"sealed":true}, {"key":"ON_TIME","label":"ON_TIME","selected":false,"sealed":true}, {"key":"DELAYED","label":"DELAYED","selected":false,"sealed":true}, {"key":"CANCELLED","label":"CANCELLED","selected":false,"sealed":true}, {"key":"LANDED","label":"LANDED","selected":false,"sealed":true}], uniqueIdentifier=, uniqueIdentifier$error=May not be null., version=0}', but got '{aircraft$error=May not be null., arrivalAirport$error=May not be null., arrivalIATACodes=[{"key":"0","label":"----","selected":true,"sealed":true}, {"key":"35","label":"LHR","selected":false,"sealed":true}, {"key":"36","label":"BCN","selected":false,"sealed":true}, {"key":"37","label":"CDG","selected":false,"sealed":true}, {"key":"38","label":"LAX","selected":false,"sealed":true}, {"key":"39","label":"JFK","selected":false,"sealed":true}, {"key":"40","label":"XXX","selected":false,"sealed":true}, {"key":"41","label":"YYY","selected":false,"sealed":true}, {"key":"42","label":"ZZZ","selected":false,"sealed":true},
```

Figura 4: Respuesta de los tests a Mutación 2 (recortado)

- Mutación 3, eliminar validación de los identificadores únicos de los tramos de vuelos.

Código modificado:

```
/*
if (!leg.getUniqueIdentifier().equals(""))
    super.state(!legIds.contains(leg.getUniqueIdentifier()), "uniqueIdentifier", "manager.leg.create.not-unique-identifier");
*/
```

Figura 5: Mutación 3 en el código



## Error encontrado tras los tests:

```
Replaying .\src\test\resources\manager\leg\create-1.hack...
FAILED POST /manager/leg/create (request-id="bc0994cc-6803-4fa7-83e9-51be1d2b192a",
input="id=0&version=0&uniqueIdentifier=1001&scheduledDeparture=&scheduledArrival=&status=0&departureAirport=0&arrivalAirport=0&aircraft=0&flight=0"):
Expected 'payload' to be '{aircraft$error=May not be null., arrivalAirport$error=May not be null., arrivalIATACodes=
[{"key":"0","label":"----","selected":true,"sealed":true}, {"key":"35","label":"LHR","selected":false,"sealed":true},
{"key":"36","label":"BCN","selected":false,"sealed":true}, {"key":"37","label":"CDG","selected":false,"sealed":true},
{"key":"38","label":"LAX","selected":false,"sealed":true}, {"key":"39","label":"JFK","selected":false,"sealed":true},
{"key":"40","label":"XXX","selected":false,"sealed":true}, {"key":"41","label":"YYY","selected":false,"sealed":true},
{"key":"42","label":"ZZZ","selected":false,"sealed":true}, {"key":"43","label":"ZZA","selected":false,"sealed":true}], departureAirport$error=May not
be null., departureIATACodes=[{"key":"0","label":"----","selected":true,"sealed":true}, {"key":"35","label":"LHR","selected":false,"sealed":true},
{"key":"36","label":"BCN","selected":false,"sealed":true}, {"key":"37","label":"CDG","selected":false,"sealed":true},
{"key":"38","label":"LAX","selected":false,"sealed":true}, {"key":"39","label":"JFK","selected":false,"sealed":true},
{"key":"40","label":"XXX","selected":false,"sealed":true}, {"key":"41","label":"YYY","selected":false,"sealed":true},
{"key":"42","label":"ZZZ","selected":false,"sealed":true}, {"key":"43","label":"ZZA","selected":false,"sealed":true}], flight$error=May not be null.,
flightChoices=[{"key":"0","label":"----","selected":true,"sealed":true}, {"key":"62","label":"62","selected":false,"sealed":true},
{"key":"63","label":"63","selected":false,"sealed":true}], id=0, isDraftMode=true, registrationNumbers=
[{"key":"0","label":"----","selected":true,"sealed":true}, {"key":"68","label":"ABC123-XYZ789-ULR-2025-TEST999","selected":false,"sealed":true},
{"key":"69","label":"DEF456-SAFE-MODEL-2025","selected":false,"sealed":true}], scheduledArrival=, scheduledArrival$error=May not be null.,
scheduledDeparture=, scheduledDeparture$error=May not be null., service=160, status=0, status$error=May not be null., statuses=
[{"key":"0","label":"----","selected":true,"sealed":true}, {"key":"ON_TIME","label":"ON_TIME","selected":false,"sealed":true},
{"key":"DELAYED","label":"DELAYED","selected":false,"sealed":true}, {"key":"CANCELLED","label":"CANCELLED","selected":false,"sealed":true},
{"key":"LANDED","label":"LANDED","selected":false,"sealed":true}], uniqueIdentifier=1001, uniqueIdentifier$error=This identifier is already in use.,
version=0}', but got '{aircraft$error=May not be null., arrivalAirport$error=May not be null., arrivalIATACodes=
[{"key":"0","label":"----","selected":true,"sealed":true}, {"key":"35","label":"LHR","selected":false,"sealed":true},
```

Figura 6: Respuesta de los tests a Mutación 3 (recortado)

- Mutación 4. no permitir la población de las previsiones meteorológicas.

## Código modificado:

```
@GetMapping("/administrator/weather/populate")
public ModelAndView populateInitial() {
    Assert.state(false, "acme.default.error.not-authorised");//PrincipalHelper.get().hasRealmOfType(Administrator.class), "acme.default.error.not-authorised");

    return this.doPopulate();
}
```

Figura 7: Mutación 4 en el código

## Error encontrado tras los tests:

```
Replaying .\src\test\resources\any\weather\populate-list-show.safe...
FAILED GET /administrator/weather/populate (request-id="65fb6089-8fef-49d0-ab51-e0e79d6e2d89", input=""): Expected 'status' to be '200', but got '500'. Expected
'payload' to be '{service=159}', but got '{}'.

```

Figura 8: Respuesta de los tests a Mutación 4

-

- Mutación 5. permitir POST hacking de los Select Choices de Legs.

Código modificado:

```
@Override
public void authorise() {

    boolean status;

    Manager manager = (Manager) super.getRequest().getPrincipal().getRealmOfType(Manager.class);
    Integer aircraftId = super.getRequest().getData("aircraft", int.class, 0);
    Integer arrivalId = super.getRequest().getData("arrivalAirport", int.class, 0);
    Integer departureId = super.getRequest().getData("departureAirport", int.class, 0);
    Integer flightId = super.getRequest().getData("flight", int.class, 0);
    List<Integer> airports = this.lr.findAllAirports().stream().map(a -> a.getId()).toList();
    List<Integer> aircrafts = this.lr.findAllAircraftsByAirlineId(manager.getAirlineManaging().getId()).stream().map(a -> a.getId()).toList();
    List<Integer> flights = this.lr.findAllFlightsEditableByManagerId(manager.getId()).stream().map(f -> f.getId()).toList();
    /*
    status =
        (aircraftId == 0 || aircrafts.contains(aircraftId)) &&
        (arrivalId == 0 || airports.contains(arrivalId)) &&
        (departureId == 0 || airports.contains(departureId)) &&
        (flightId == 0 || flights.contains(flightId));
    */
    status = true;

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

Figura 9: Mutación 5 en el código

Error encontrado en los tests:

```
FAILED POST /manager/leg/create (request-id="eecb4785-8ebd-4d56-a793-f0aadffe4520",
input="id=0&version=0&uniqueIdentifier=&scheduledDeparture=&scheduledArrival=&status=ON_TIME&departureAirport=30&arrivalAirport=0&aircraft=0&flight=0"): Expected 'status' to be '500', but got '200'. Expected 'payload' to be '{service=159}', but got '{aircraft
$error=May not be null., arrivalAirport$error=May not be null., arrivalIATACodes=
[{"key":"0","label":"----","selected":true,"sealed":true}, {"key":"35","label":"LHR","selected":false,"sealed":true},
{"key":"36","label":"BCN","selected":false,"sealed":true}, {"key":"37","label":"CDG","selected":false,"sealed":true},
{"key":"38","label":"LAX","selected":false,"sealed":true}, {"key":"39","label":"JFK","selected":false,"sealed":true},
{"key":"40","label":"XXX","selected":false,"sealed":true}, {"key":"41","label":"YYY","selected":false,"sealed":true},
{"key":"42","label":"ZZZ","selected":false,"sealed":true}, {"key":"43","label":"ZZA","selected":false,"sealed":true}], departureAirport
$error=May not be null., departureIATACodes=[{"key":"0","label":"----","selected":true,"sealed":true},
{"key":"35","label":"LHR","selected":false,"sealed":true}, {"key":"36","label":"BCN","selected":false,"sealed":true},
{"key":"37","label":"CDG","selected":false,"sealed":true}, {"key":"38","label":"LAX","selected":false,"sealed":true},
{"key":"39","label":"JFK","selected":false,"sealed":true}, {"key":"40","label":"XXX","selected":false,"sealed":true},
{"key":"41","label":"YYY","selected":false,"sealed":true}, {"key":"42","label":"ZZZ","selected":false,"sealed":true},
{"key":"43","label":"ZZA","selected":false,"sealed":true}], flight$error=May not be null., flightChoices=
[{"key":"0","label":"----","selected":true,"sealed":true}, {"key":"62","label":"62","selected":false,"sealed":true},
{"key":"63","label":"63","selected":false,"sealed":true}], id=0, isDraftMode=true, registrationNumbers=
[{"key":"0","label":"----","selected":true,"sealed":true}, {"key":"68","label":"ABC123-XYZ789-ULR-2025-
TEST999","selected":false,"sealed":true}, {"key":"69","label":"DEF456-SAFE-MODEL-2025","selected":false,"sealed":true}],
scheduledArrival=, scheduledArrival$error=May not be null., scheduledDeparture=, scheduledDeparture$error=May not be null.,
service=159, status=ON_TIME, statuses=[{"key":"0","label":"----","selected":false,"sealed":true},
{"key":"ON_TIME","label":"ON_TIME","selected":true,"sealed":true}, {"key":"DELAYED","label":"DELAYED","selected":false,"sealed":true},
```

Figura 10: Respuesta de los tests a Mutación 5 (recortado)

## **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, hasta que se le hicieron las mutaciones, 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