Informe de Testing Grupal – D04



Grupo: *C1.010*

Repositorio: [*Acme-ANS*](https://github.com/ASoult10/Acme-ANS)

(https://github.com/ASoult10/Acme-ANS)

Miembros:

* **Alejandro Soult Toscano (**[alesoutos@alum.us.es](mailto:alesoutos@alum.us.es)**)**
* **Mario Benítez Galván (**[marbengal@alum.us.es](mailto:marbengal@alum.us.es)**)**
* **Enrique Nicolae Barac Ploae (**[enrbarplo@alum.us.es](mailto:enrbarplo@alum.us.es)**)**
* **Marta de la Calle González (**[mardegon7@alum.us.es](mailto:mardegon7@alum.us.es)**)**
* **Manuel Alcaraz Zambrano (**[manalczam@alum.us.es](mailto:manalczam@alum.us.es)**)**

Fecha: ***25/05/2025***

# Tabla de contenidos

[Tabla de contenidos 1](#_Toc197365565)

[1. Resumen Ejecutivo 2](#_Toc197365566)

[2. Tabla de versiones 3](#_Toc197365567)

[3. Introducción 4](#_Toc197365568)

[4. Testing Funcional 5](#_Toc197365569)

[5. Rendimiento de Testing 17](#_Toc197365570)

[6. Conclusiones 22](#_Toc197365571)

[7. Bibliografía 23](#_Toc197365572)

# 1. Resumen Ejecutivo

Este informe de testing documenta las pruebas realizadas para garantizar la funcionalidad y el rendimiento del proyecto Acme AirNav Solutions. Se organiza en dos capítulos:

* **Testing funcional:** Se presenta un listado de los casos de prueba implementados, agrupados por características clave del sistema. Cada caso de prueba incluye una descripción concisa y una evaluación sobre su efectividad en la detección de errores. La ejecución sistemática de estas pruebas ha sido fundamental para identificar fallos y asegurar la estabilidad de las funciones del software.
* **Rendimiento de testing:** Se analiza el tiempo de respuesta del sistema en dos computadoras distintas, proporcionando gráficos e intervalos de confianza del 95% para el tiempo de procesamiento de las solicitudes realizadas en los tests funcionales. Además, se lleva a cabo un contraste de hipótesis con un 95% de confianza para determinar cuál de los equipos ofrece un mejor rendimiento.

Gracias a este proceso de prueba exhaustivo, se ha mejorado la calidad y la fiabilidad del sistema, asegurando que cumple con los requisitos establecidos y las expectativas del cliente.

# 2. Tabla de versiones

|  |  |  |
| --- | --- | --- |
| **Fecha** | **Versión** | **Descripción** |
| 18/02/2025 | 0.0 | Creación del documento plantilla |
| 05/05/2025 | 1.0 | Preparación apartados para testing |
| 25/05/2025 | 2.0 | Realización de las pruebas y finalización del documento |

# 3. Introducción

El grupo C1.010, se encuentra en la fase inicial del proyecto de Acme AirNav Solutions, el cual representa un desafío significativo y estimulante.

La cuarta y última entrega del proyecto es una fase crítica centrada en el testing. Este momento es crucial para garantizar la calidad del producto final, ya que es fundamental probar y cubrir todas las casuísticas posibles de la aplicación, tanto positivas como negativas. Esto permitirá identificar y corregir posibles errores, asegurando que el producto cumpla fielmente con todos los requisitos. La complejidad y densidad de esta etapa requieren un enfoque detallado y un compromiso total con las revisiones y pruebas, estableciendo una base sólida para el éxito final del proyecto.

Para ello, haremos una serie de pruebas *End-To-End*, probando tanto los casos positivos, casos negativos y casos de hacking. Se probarán todas las funcionalidades correspondientes al requisito grupal 11, es decir, listar, crear y actualizar los aeropuertos.

Por tanto, haremos pruebas positivas de listado y de creación (Poniendo los valores en los límites y en valores cercanos a los límites). Además, haremos pruebas negativas con valores no permitidos, y pruebas de hacking, de forma que cualquier intento de ataque de tipo GET o POST sea detenido. Una vez hechas todas estas pruebas, las reproduciremos en dos equipos distintos, comparando sus tiempos y realizando un contraste de hipótesis con un nivel de confianza del 95% para determinar cuál es el equipo más potente

Confío en que, con esfuerzo y dedicación, lograremos entregar un producto de alta calidad que cumpla con las expectativas de nuestros clientes y profesores.

# 4. Testing Funcional

En esta sección se presentarán los casos de prueba agrupados por funcionalidad, correspondiente al requisito grupal 11. Así, se mostrará el propósito de cada caso de prueba y si ha sido útil para encontrar bugs.

## 4.1 Testing de creación de y gestión de aeropuertos

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Identificador | Propósito | Resultado | Bugs detectados | Efectividad |
| CP-01 | Listar aeropuertos | Mostrar aeropuetos | 0 | Baja |
| CP-02 | Mostrar un aeropuerto | Mostrar los detalles de un aeropuerto seleccionado | 0 | Baja |
| CP-03 | Crear un aeropuerto con datos vacíos | Se muestran por pantalla errores no bloqueantes | 0 | Baja |
| CP-04 | Crear un aeropuerto con un nombre demasiado corto | Se muestra un error bloqueante *Not Authorised* | 0 | Baja |
| CP-05 | Crear un aeropuerto con un nombre demasiado largo | Se muestra un error bloqueante *Not Authorised* | 0 | Baja |
| CP-06 | Crear un aeropuerto con un nombre válido | No muestra errores vinculados al aeropuerto | 0 | Baja |
| CP-07 | Crear un aeropuerto con un *IATA Code* demasiado largo | Se muestra un error no bloqueante | 0 | Baja |
| CP-08 | Crear un aeropuerto con un *IATA Code* demasiado corto | Se muestra un error no bloqueante | 0 | Baja |
| CP-09 | Crear un aeropuerto con un *IATA Code* que no sigue el patrón | Se muestra un error no bloqueante | 0 | Baja |
| CP-10 | Crear un aeropuerto conun *IATA Code* ya en uso | Se muestra un error no bloqueante | 0 | Baja |
| CP-11 | Crear un aeropuerto con un *IATA Code* correcto | No muestra errores vinculados al *IATA Code* | 0 | Baja |
| CP-12 | Crear un aeropuerto con un *scope* vacío | No muestra errores vinculados al *scope* | 0 | Baja |
| CP-13 | Crear un aeropuerto conun *scope* como cadena | Se muestra un error no bloqueante | 0 | Baja |
| CP-14 | Crear un aeropuerto con un *scope* correcto | No muestra ningún error vinculado al *scope* | 0 | Baja |
| CP-15 | Crear un aeropuerto con un c*ity* demasiado largo | Se muestra un error no bloqueante | 0 | Baja |
| CP-16 | Crear un aeropuerto con un *city* demasiado corto | Se muestra un error no bloqueante | 0 | Baja |
| CP-17 | Crear un aeropuerto con un *city* correcto | No muestra errores vinculados al *city* | 0 | Baja |
| CP-18 | Crear un aeropuerto con un *country* demasiado largo | Se muestra un error no bloqueante | 0 | Baja |
| CP-19 | Crear un aeropuerto con un *country* demasiado corto | Se muestra un error no bloqueante | 0 | Baja |
| CP-20 | Crear un aeropuerto con un *country* correcto | No muestra errores vinculados al *country* | 0 | Baja |
| CP-21 | Actualizar un aeropuerto con un nombre demasiado corto | Se muestra un error bloqueante *Not Authorised* | 0 | Baja |
| CP-22 | Actualizar un aeropuerto con un nombre demasiado largo | Se muestra un error bloqueante *Not Authorised* | 0 | Baja |
| CP-23 | Actualizar un aeropuerto con un nombre valido | No muestra errores vinculados al vuelo | 0 | Baja |
| CP-24 | Actualizar un aeropuerto con un *IATA Code* demasiado largo | Se muestra un error no bloqueante | 0 | Baja |
| CP-25 | Actualizar un aeropuerto con un *IATA Code* demasiado corto | Se muestra un error no bloqueante | 0 | Baja |
| CP-26 | Actualizar un aeropuerto con un *IATA Code* que no sigue el patrón | Se muestra un error no bloqueante | 0 | Baja |
| CP-27 | Actualizar un aeropuerto con un *IATA Code* correcto | No muestra errores vinculados al  *IATA Code* | 0 | Baja |
| CP-28 | Actualizar un aeropuerto con un c*ity* vacío | No muestra errores vinculados al *city* | 0 | Baja |
| CP-29 | Actualizar un aeropuerto conun *city demasiado* largo | Se muestra un error no bloqueante | 0 | Baja |
| CP-30 | Actualizar un aeropuerto con un *city* demasiado corto | Se muestra un error no bloqueante | 0 | Baja |
| CP-31 | Actualizar un aeropuerto con un *city* como cadena | Se muestra un error no bloqueante | 0 | Baja |
| CP-32 | Actualizar un aeropuerto con un *city* correcto | No muestra ningún error vinculado al *city* | 0 | Baja |
| CP-33 | Actualizar un aeropuerto conuna s*cope* no escogida | Se muestra un error no bloqueante | 0 | Baja |
| CP-34 | Actualizar una reserva conun *scope correcta* | No muestra ningún error vinculado al *scope* | 0 | Baja |
| CP-35 | Actualizar un aeropuerto con un *country* vacío | No muestra errores vinculados al *city* | 0 | Baja |
| CP-36 | Actualizar un aeropuerto conun *country demasiado* largo | Se muestra un error no bloqueante | 0 | Baja |
| CP-37 | Actualizar un aeropuerto con un *country* demasiado corto | Se muestra un error no bloqueante | 0 | Baja |
| CP-38 | Actualizar un aeropuerto con un *country* como cadena | Se muestra un error no bloqueante | 0 | Baja |
| CP-39 | Actualizar un aeropuerto con un *country* correcto | No muestra ningún error vinculado al *country* | 0 | Baja |
| CP-40 | Crear un aeropuerto con un *website* que no cumple el formato | Se muestra un error no bloqueante | 0 | Baja |
| CP-41 | Crear un aeropuerto con un *email* que no cumple el formato | Se muestra un error no bloqueante | 0 | Baja |
| CP-42 | Crear un aeropuerto con un *phone* que no cumple el formato | Se muestra un error no bloqueante | 0 | Baja |
| CP-43 | Crear un aeropuerto con un *webiste* correcto | No muestra ningún error vinculado al *website* | 0 | Baja |
| CP-44 | Crear un aeropuerto con un *email* correcto | No muestra ningún error vinculado al *email* | 0 | Baja |
| CP-45 | Crear un aeropuerto con un *phone* correcto | No muestra ningún error vinculado al *phone* | 0 | Baja |
| CP-46 | Actualizar un aeropuerto con un *website* que no cumple el formato | Se muestra un error no bloqueante | 0 | Baja |
| CP-47 | Actualizar un aeropuerto con un *email* que no cumple el formato | Se muestra un error no bloqueante | 0 | Baja |
| CP-48 | Actualizar un aeropuerto con un *phone* que no cumple el formato | Se muestra un error no bloqueante | 0 | Baja |
| CP-49 | Actualizar un aeropuerto con un *webiste* correcto | No muestra ningún error vinculado al *website* | 0 | Baja |
| CP-50 | Actualizar un aeropuerto con un *email* correcto | No muestra ningún error vinculado al *email* | 0 | Baja |
| CP-51 | Actualizar un aeropuerto con un *phone* correcto | No muestra ningún error vinculado al *phone* | 0 | Baja |

# 5. Rendimiento de Testing

Se ha probado todos los casos de pruebas en dos ordenadores, mostrando en este apartado los resultados en cada uno de ellos.

## 5.1 Resultados en el primer ordenador

Para la ejecución de los casos de prueba en el primer ordenador se muestran las siguientes medias de tiempo (en milisegundos) para cada URI correspondiente a cada petición:

|  |  |
| --- | --- |
| URI | Media (ms) |
| / | 13.2813535 |
| /administrator/airport/create | 39.6492312 |
| /administrator/airport/list | 21.9990802 |
| /administrator/airport/show | 18.1107817 |
| /administrator/airport/update | 35.8081266 |
| /anonymous/system/sign-in | 8.9797 |
| / any/system/welcome | 4.309184 |
| / authenticated/system/sign-out | 4.9918598 |
| Promedio General | 31.2490774 |

Estos datos se pueden representar visualmente con una gráfica:

La estadística descriptiva que podemos sacar de estos datos es la siguiente:

|  |  |
| --- | --- |
| *Datos Estadísticos* | |
|  |  |
| Media | 31.7349449 |
| Error típico | 0.90906373 |
| Mediana | 31.0486 |
| Moda | #N/D |
| Desviación estándar | 16.8360958 |
| Varianza de la muestra | 283.454122 |
| Curtosis | 0.36447334 |
| Coeficiente de asimetría | 0.24071255 |
| Rango | 87.766101 |
| Mínimo | 2.1209 |
| Máximo | 89.887001 |
| Suma | 10885.0861 |
| Cuenta | 343 |
| Nivel de confianza(95,0%) | 1.78805985 |

Finalmente, con un nivel de confianza de *1.78805985*, podemos establecer el intervalo de confianza:

|  |  |  |
| --- | --- | --- |
| Intervalo (ms) | 29.946885 | 33.5320047 |
| Intervalo (s) | 0.02994689 | 0.033523 |

## 5.2 Resultados en el segundo ordenador

Para la ejecución de los casos de prueba en el segundo ordenador se muestran las siguientes medias de tiempo para cada URI correspondiente a cada petición:

|  |  |
| --- | --- |
| URI | Media (ms) |
| / | 15.7085867 |
| /administrator/airport/create | 59.7793382 |
| /administrator/airport/list | 28.63147 |
| /administrator/airport/show | 26.3581727 |
| /administrator/airport/update | 48.7647808 |
| /anonymous/system/sign-in | 11.18018 |
| / any/system/welcome | 5.04402 |
| / authenticated/system/sign-out | 5.86346 |
| Promedio General | 44.7168663 |

Estos datos se pueden representar visualmente con una gráfica:

La estadística descriptiva que podemos sacar de estos datos es la siguiente:

|  |  |
| --- | --- |
| *Datos Estadísticos* | |
|  |  |
| Media | 45.5294589 |
| Error típico | 1.56781834 |
| Mediana | 41.1439 |
| Moda | #N/D |
| Desviación estándar | 29.036402 |
| Varianza de la muestra | 843.112643 |
| Curtosis | 6.49753554 |
| Coeficiente de asimetría | 1.59579002 |
| Rango | 219.1789 |
| Mínimo | 2.4964 |
| Máximo | 221.6753 |
| Suma | 15616.6044 |
| Cuenta | 343 |
| Nivel de confianza(95,0%) | 3.08378054 |

Finalmente, con un nivel de confianza de *3.08378054*, podemos establecer el intervalo de confianza:

|  |  |  |
| --- | --- | --- |
| Interval(ms) | 42.4456784 | 48.6132395 |
| Interval(s) | 0.04244568 | 0.04861324 |

## 5.3 Comparación Final

Haciendo una prueba Z, podemos ver si estas diferencias son comparables o no:

|  |  |  |
| --- | --- | --- |
|  | *Primer Ordenador* | *Segundo Ordenador* |
| Media | 30.9625501 | 44.281198 |
| Varianza (conocida) | 283.414522 | 843.112643 |
| Observaciones | 359 | 359 |
| Diferencia hipotética de las medias | 0 |  |
| z | -7.51846002 |  |
| P(Z<=z) una cola | 2.7756E-14 |  |
| Valor crítico de z (una cola) | 1.64485363 |  |
| P(Z<=z) dos colas | 5.5511E-14 |  |
| Valor crítico de z (dos colas) | 1.95996398 |  |

Podemos ver que el valor de P es menor que 0.05. Esto quiere decir que la diferencia de rendimiento entre un ordenador y otro es suficientemente significativa y, por lo tanto, podemos comparar los dos valores.

# 6. Conclusiones

La fase de pruebas representa un componente fundamental en el ciclo de desarrollo de cualquier sistema informático. Mediante evaluaciones tanto funcionales como de rendimiento, se puede comprobar que la aplicación cumple con los requerimientos esperados, brindando así estabilidad, eficiencia y una experiencia satisfactoria al usuario. Llevar a cabo estas pruebas de forma estructurada permite localizar fallos, corregir imperfecciones y asegurar un funcionamiento consistente en distintos escenarios de uso.

Sin embargo, evaluar solo el funcionamiento y la velocidad del sistema no basta para garantizar su confiabilidad y seguridad. En un contexto donde las amenazas informáticas son cada vez más frecuentes, resulta indispensable incluir pruebas orientadas a la seguridad y a la resistencia frente a ataques. Estas evaluaciones permiten descubrir fallos de seguridad que podrían poner en riesgo la integridad del sistema y la privacidad de los datos. Adoptar medidas preventivas y simular intentos de intrusión ayuda a fortalecer la protección del software, minimizando posibles impactos negativos para la organización y sus usuarios.

En consecuencia, el proceso de pruebas debe entenderse como una actividad integral que abarca no solo aspectos técnicos, sino también la protección y robustez del sistema. Un enfoque combinado que contemple funcionalidad, desempeño y seguridad es clave para entregar una solución confiable, eficiente y protegida frente a amenazas, fomentando la confianza y satisfacción del usuario final.

# 7. Bibliografía

* [*06 – Annexes.dock*](https://ev.us.es/bbcswebdav/pid-5227970-dt-content-rid-62285040_1/xid-62285040_1)– Project Statement, Asignatura de Diseño y Pruebas II del grado de Ingeniería Informática del Software, Universidad de Sevilla.