**Testing report**

**Acme Ans C3**

**Grupal**

**Group C3.X01**

**Student:**

* Olga Cantalejo Gómez ([olgcangom@alum.us.es](mailto:olgcangom@alum.us.es))

**Repository:** [**https://github.com/olgacangom/Acme-ANS-C3**](https://github.com/olgacangom/Acme-ANS-C3)

**Date:** 15/10/2025

Contenido

[1. Resumen ejecutivo 3](#_Toc211457645)

[2. Tabla de versiones 4](#_Toc211457646)

[3. Introducción 5](#_Toc211457647)

[4. Testing functional 6](#_Toc211457648)

[5. Rendimiento del testing 7](#_Toc211457649)

[5.1 Resultados sin índices 7](#_Toc211457650)

[5.2 Resultados con índices 8](#_Toc211457651)

[5.3 Comparación final 9](#_Toc211457652)

[6. Conclusión 10](#_Toc211457653)

[7. Bibliografía 11](#_Toc211457654)

# Resumen ejecutivo

Este informe expone los resultados obtenidos tras realizar pruebas funcionales y de rendimiento sobre las operaciones de gestión aeroportuaria destinadas a los administradores, enmarcadas dentro del requisito funcional número 11 del sistema. Las funcionalidades evaluadas comprenden la consulta del listado de aeropuertos (list), la visualización de sus características (show), así como la creación (create) y modificación de estos registros (update), acciones que requieren confirmación explícita por parte del usuario.

Las pruebas funcionales se han planteado con un enfoque integral, abarcando tanto escenarios válidos como situaciones de error y accesos no autorizados, con el objetivo de verificar la solidez, seguridad y fiabilidad de la aplicación. Para ello, se han considerado casos positivos, negativos y de tipo hacking, siguiendo una metodología formal de verificación y validación del comportamiento esperado del sistema.

Asimismo, se han ejecutado pruebas de rendimiento, registrando los tiempos de respuesta y analizando los resultados mediante técnicas estadísticas. Se ha calculado un intervalo de confianza del 95% para comparar el funcionamiento del sistema en diversos entornos y detectar posibles diferencias significativas en términos de eficiencia

# Tabla de versiones

|  |  |  |
| --- | --- | --- |
| **Fecha** | **Versión** | **Descripción** |
| 09/10/2025 | 0.0 | Creación del documento plantilla |
| 15/10/2025 | 1.0 | Actualización del documento |

# Introducción

Como integrante del grupo C3.X01, afronto la cuarta y última fase del proyecto **Acme-ANS**, centrada en el proceso de testing. Esta etapa representa un punto crítico en el desarrollo, ya que permite validar la calidad, seguridad y fiabilidad del sistema antes de su posible despliegue en un entorno real. El objetivo principal es verificar de forma exhaustiva las funcionalidades asignadas al rol de administrador (administrator), especialmente aquellas relacionadas con la gestión de aeropuertos (airport), tal como se especifica en el Requisito Funcional Nº 11.

El informe se estructura en dos secciones principales: una dedicada al análisis de las pruebas funcionales y otra enfocada en la evaluación del rendimiento. Para garantizar una validación completa, se han diseñado distintos tipos de pruebas que abarcan casos positivos (utilizando valores límite y cercanos al límite), negativos (con entradas no válidas) y simulaciones de ataques (para interceptar intentos de acceso no autorizado mediante peticiones GET y POST).

# Testing functional

Las pruebas se realizaron siguiendo la metodología descripta en las dispositivas teóricas, con el objetivo de alcanzar el mayor nivel de cobertura possible.

* **Create:** Se testearon múltiples combinaciones de atributos al crear nuevos aeropuertos, incluyendo valores inválidos, con el fin de comprobar que se generaban los mensajes de error esperados. En el apartado de hacking, se intentó crear registros sin autenticación, reutilizar identificadores ya existentes, entre otros casos. La funcionalidad demostró un comportamiento correcto en todos los intentos, obteniendo una cobertura del 100%.
* **List:** Se verificó la funcionalidad de visualización del listado de aeropuertos, asegurando que el sistema respondiera correctamente ante accesos válidos. En las pruebas de seguridad, se simularon intentos de acceso no autorizado: desde usuarios no autenticados, usuarios incorrectos dentro de un realm válido, y realms inválidos. Todos los casos fueron gestionados correctamente por el sistema, alcanzando una cobertura del 100%.
* **Show:** Esta funcionalidad fue evaluada mediante la consulta de información detallada de un aeropuerto específico. Para validar la robustez del sistema, se realizaron pruebas de acceso indebido, como el uso de credenciales de roles no autorizados y la consulta de aeropuertos inexistentes. El sistema respondió adecuadamente en todos los escenarios, logrando una cobertura del 100%.
* **Update:** Las pruebas de modificación incluyeron variaciones en los atributos, tanto válidas como inválidas, para verificar la gestión de errores. También se evaluaron intentos de actualización sin autenticación, sobre aeropuertos inexistentes y con usuarios sin permisos administrativos. El sistema manejó correctamente cada situación, alcanzando una cobertura del 100%.

Gráfico, Gráfico de barras

El contenido generado por IA puede ser incorrecto.

# Rendimiento del testing

## 5.1 Resultados sin índices

Interfaz de usuario gráfica, Texto, Aplicación

El contenido generado por IA puede ser incorrecto.

Gráfico, Gráfico de barras

El contenido generado por IA puede ser incorrecto.Estos datos se aprecian con major claridad en la siguiente gráfica.

## 5.2 Resultados con índices

Se ha añadido a la entidad Airport, el siguiente índice:

@Table(indexes = {

@Index(columnList = "iataCode"), @Index(columnList = "email")

})

Interfaz de usuario gráfica, Aplicación, Tabla

El contenido generado por IA puede ser incorrecto.

Estos datos se aprecian con major claridad en la siguiente gráfica

Gráfico, Gráfico de barras, Gráfico en cascada

El contenido generado por IA puede ser incorrecto.

## Comparación final

Tabla

El contenido generado por IA puede ser incorrecto.

Tabla

El contenido generado por IA puede ser incorrecto.

# Conclusión

La fase de pruebas constituye un pilar esencial en el desarrollo de sistemas informáticos, ya que permite verificar que la aplicación cumple con los requisitos funcionales y de rendimiento esperados. En este estudio comparativo, se han analizado los tiempos de respuesta con y sin índices, aplicando métricas estadísticas y una prueba Z para determinar la significancia de las diferencias observadas.

El estadístico obtenido fue z = 0.0329, con un valor p (dos colas) = 0.9737, muy superior al nivel de significación α = 0.05.

Dado que |z| < 1.9599 y p > 0.05, no se rechaza la hipótesis nula.

Por tanto, no se observan diferencias estadísticamente significativas entre los tiempos antes y después de las modificaciones.

Los cambios aplicados no produjeron una mejora real en el rendimiento del sistema.

Estas conclusiones refuerzan la importancia de realizar pruebas estructuradas y cuantificables, no solo para validar la funcionalidad del sistema, sino también para identificar variaciones de rendimiento que puedan afectar la experiencia del usuario. Además, este tipo de análisis permite tomar decisiones informadas sobre la infraestructura tecnológica más adecuada para garantizar eficiencia, estabilidad y escalabilidad en distintos escenarios de uso.

# Bibliografía

Intencionalmente en blanco.