Universidad de Sevilla  
Escuela Técnica Superior de Ingeniería Informática

**Informe de Testing D04**



Grado en Ingeniería Informática – Ingeniería de Software  
Diseño y Pruebas II

Año académico 2024 – 2025

|  |  |
| --- | --- |
| **Fecha** | **Versión** |
| 03/07/2025 | V2.0.0 |

|  |  |  |
| --- | --- | --- |
| **Laboratory group C2.005** | | |
| **Alphabetised authors** | **Role** | **Role description** |
| González Benito, Claudio – clagonben@alum.us.es | Jefe de proyecto  Desarrollador  Tester | Encargado de tomar decisiones de diseño y vigilar el correcto desarrollo.  Persona encargada de desarrollar el código  Persona encargada de realizar pruebas sobre el código |
| Ramos Vargas, Alba – albramvar1@alum.us.es | Operador  Desarrollador  Tester | Encargado de las tareas de campo, de las instalaciones y del mantenimiento de los sistemas de la empresa  Persona encargada de desarrollar el código  Persona encargada de realizar pruebas sobre el código |

# Historial de versiones

|  |  |  |
| --- | --- | --- |
| Date | Version | Description |
| 26/05/2025 | v1.0.0 | Creación del documento |
| 03/07/2025 | V2.0.0 | Actualización por nueva convocatoria |

Repositorio: <https://github.com/Manuelgithuv/Acme-ANS-C2>

Tabla de contenidos

[Historial de versiones 2](#_Toc199187864)

[Resumen ejecutivo 3](#_Toc199187865)

[Introducción 3](#_Toc199187866)

[Testing funcional 3](#_Toc199187867)

[Casos específicos 4](#_Toc199187868)

[Cobertura 6](#_Toc199187869)

[Rendimiento 6](#_Toc199187870)

[Conclusiones 8](#_Toc199187871)

# Resumen ejecutivo

El objetivo del informe es estudiar el testing hecho para los requisitos grupales del proyecto. En él se especificarán los casos, los resultados esperados y cualquier discrepancia con estos. También se verá la cobertura conseguida y se comparará el rendimiento en dos casos diferentes.

# Introducción

En este informe se especificarán los cases de prueba que se han desarrollado, es decir, listar, enseñar, crear, actualizar, borrar y publicar, para todos los módulos grupales.

Se detallará qué debe estudiar cada caso de prueba de forma general y de forma particular, especificando las validaciones específicas y cuándo se espera una excepción. Además, se considerará y estudiará la cobertura que produces estos tests.

Finalmente, en el segundo capítulo de este informe se estudiará el rendimiento, considerando dos ordenadores

# Testing funcional

Los casos genéricos que se van a considerar son:

* Lista: el usuario debe recibir una lista de la entidad estudiada.
* Enseñar: el usuario debe poder ver toda la información relevante de la entidad en cuestión.
* Crear: el usuario debe poder crear una entidad, siguiendo las validaciones necesarias.
* Actualizar: el usuario debe poder actualizar una entidad preexistente, siguiendo las validaciones necesarias.

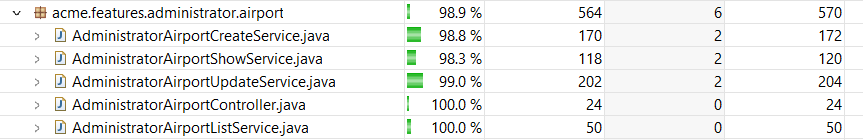
Adicionalmente, se considerarán dos tipos de tests. Los tests ‘legales’ deben contener sólo acciones legales y nunca podrán obtener una excepción. Los tests ‘ilegales’ tienen el objetivo de obtener estas excepciones, específicamente las 500 Acceso no autorizado.

## Casos específicos

|  |  |  |  |
| --- | --- | --- | --- |
| Caso |  | Resultado esperado | Bugs detectados |
| Listar | **LEGAL** | El sistema debe enseñar una lista de los aeropuertos a los administradores. | Ningún bug detectado |
| **ILEGAL** | Cuando un usuario sin el rol de administrador intente acceder al listado, debe recibir una excepción de no autorizado. | Ningún bug detectado |
| Enseñar | **LEGAL** | Desde el listado de aeropuertos, el usuario debe poder acceder a una vista detallada de los aeropuertos. | Ningún bug detectado |
| **ILEGAL** | El sistema debe mostrar un error 500 Acceso no autorizado en los siguientes casos:   * El usuario intenta acceder a una entidad con id no correspondiente a un aeropuerto * El usuario no especifica un id para mostrar | Ningún bug detectado |
| Crear | **LEGAL** | Un administrador debe poder crear una entidad aeropuerto con las siguientes consideraciones:   * Los campos de nombre, código IATA, dominio de operaciones, ciudad y país no deben ser nulos. * El nombre debe ser menor de 50 caracteres. * El código IATA debe estar formado de 3 caracteres alfabéticos en mayúsculas. * La ciudad y el país debe ser menor de 50 caracteres. * El teléfono de contacto debe seguir el patrón en expresión regular "^\+?\d{6,15}$" | Ningún bug detectado |
| **ILEGAL** | El sistema debe devolver un error de validación para los siguientes intentos de hackeo:   * Seleccionar un dominio de operaciones que no existe.   El sistema debe devolver una excepción 500 Acceso no autorizado en los siguientes casos:   * Un usuario sin rol de administrador intenta acceder al formulario de creación. | Ningún bug detectado |
| Actualizar | **LEGAL** | Un administrador debe poder actualizar un aeropuerto preexistente siguiendo los siguientes requisitos:   * Los campos de nombre, código IATA, dominio de operaciones, ciudad y país no deben ser nulos. * El nombre debe ser menor de 50 caracteres. * El código IATA debe estar formado de 3 caracteres alfabéticos en mayúsculas. * La ciudad y el país debe ser menor de 50 caracteres. * El teléfono de contacto debe seguir el patrón en expresión regular "^\+?\d{6,15}$" | Ningún bug detectado |
| **ILEGAL** | El sistema debe devolver un error de validación para los siguientes intentos de hackeo:   * Seleccionar un dominio de operaciones que no existe.   El sistema debe devolver una excepción 500 Acceso no autorizado en los siguientes casos:   * Un usuario sin rol de administrador intenta hacer cualquier petición de actualización. * El usuario no da un id para actualizar. * El usuario intenta acceder a un id de una entidad que no es un aeropuerto. | Ningún bug detectado |

## Cobertura

La cobertura conseguida con las pruebas anteriores es:



Como podemos ver, la cobertura proporcionada por las pruebas es casi perfecta, siendo la menor un 98.3% para el servicio de ‘show’.

# Rendimiento

Hemos estudiado las diferencias de rendimiento de dos ordenadores de miembros del equipo. En este apartado estudiaremos si existen diferencias notables entre estos dos dispositivos. A continuación, vemos las gráficas que obtenemos:

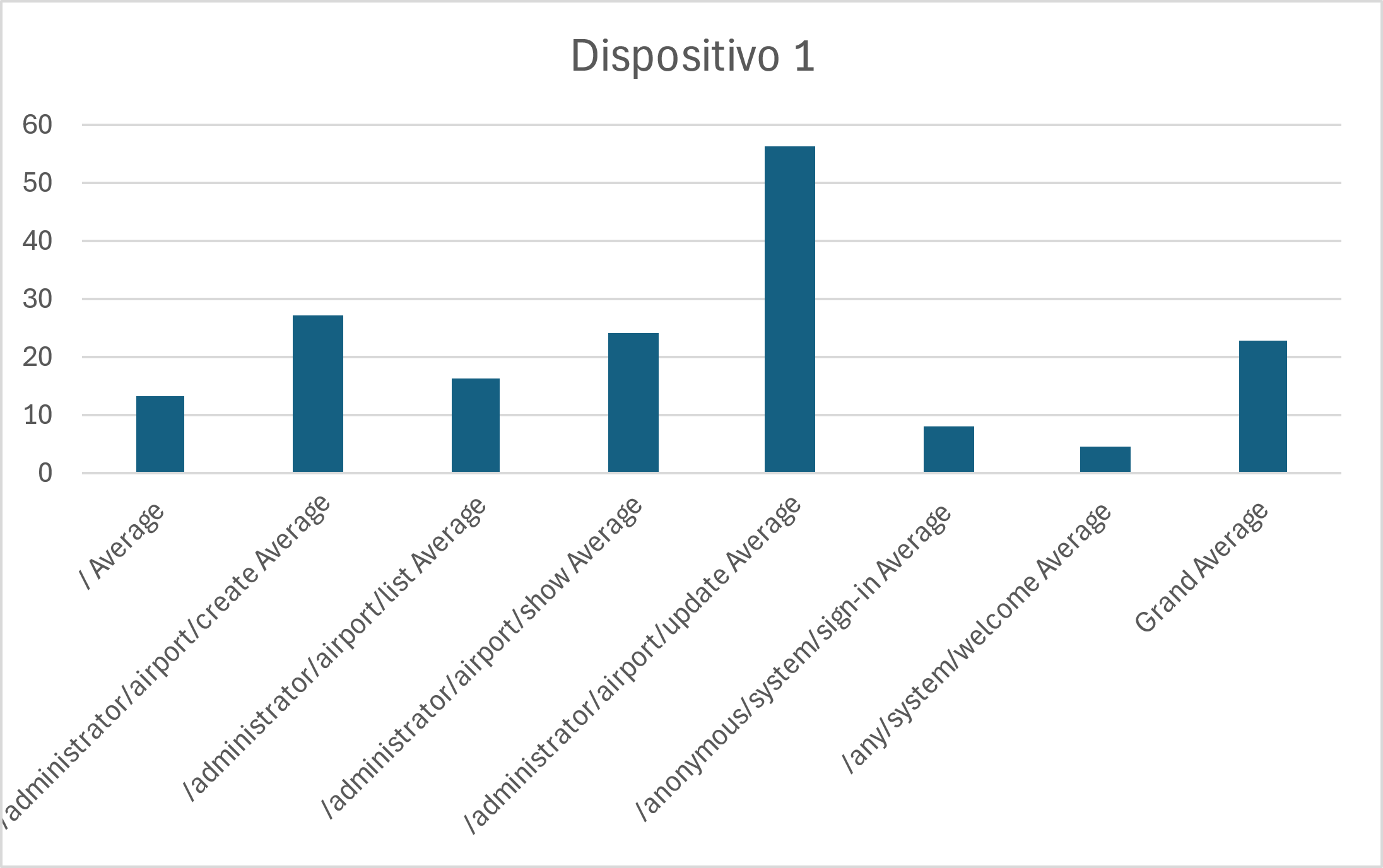


Figura 1. Prueba 1

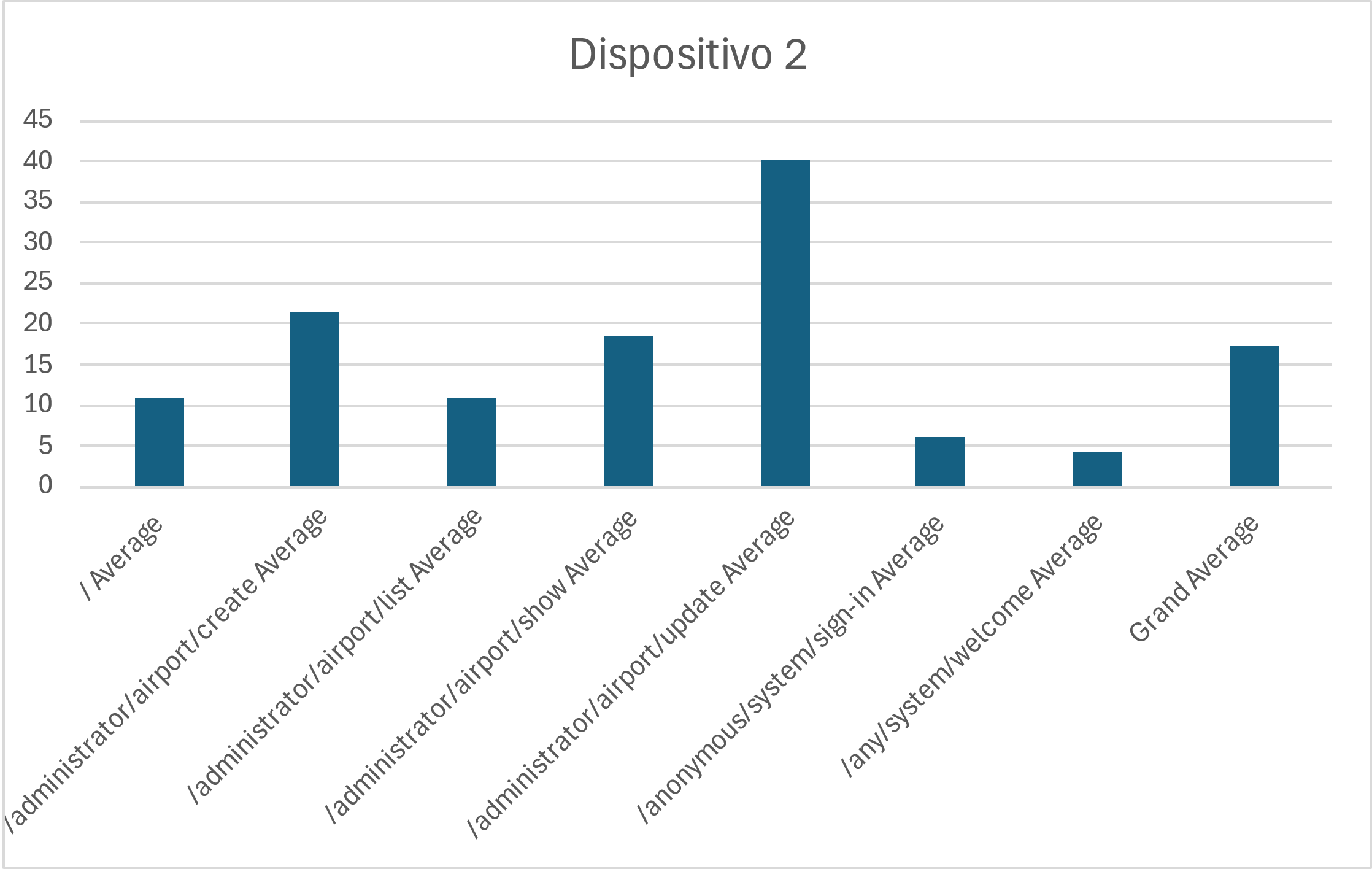


Figura 2. Prueba 2

Podemos ver que, aunque la mayoría de los casos de prueba se mantienen constantes, el caso de actualización decrece de unos 57.5 (aproximado) hasta unos 40. Sin embargo, para ver una comparación más exacta estudiaremos el nivel de significancia y el valor crítico de z.

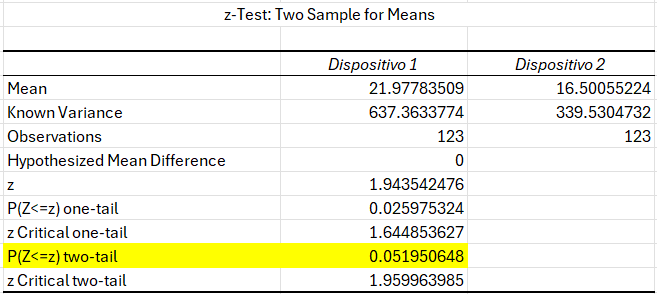


Figura 3. Comparación de valores mediante z-test

En nuestra evaluación del rendimiento, observamos que hemos obtenido un valor de **0.051950648** para un nivel de significancia α de **0.95**. Siguiendo la metodología estudiada en clase, al encontrarse este valor de z es tan cercano a α no podemos decir que haya habido una mejora significativa.

# Conclusiones

En este proceso de testing hemos definido pruebas específicas para cada funcionalidad requerida por el cliente. El entorno de pruebas utilizado nos proporciona además una gran cantidad de datos que podemos estudiar para comprobar la evolución del rendimiento.