



# S1 Test Report

| Grupo C1.02.08 |

| GitHub: <https://github.com/mariaescalante/Acme-L3-Do4-24.1.0> |

Joaquín Arregui Díaz (joaarrdia@alum.us.es)

| 26/05/2023 |

Versión	Descripción	Fecha
1.0	Versión inicial	25/05/2023
1.1	Versión final	26/05/2023

# Índice

Introducción: .....	2
Contenido: .....	2
Testing funcional: .....	2
Course:.....	2
Lecture:.....	3
Membership: .....	5
Pruebas de rendimiento: .....	6
Ordenador de Joaquín Arregui Diaz .....	6
Ordenador de Francisco Antonio Campos Campos .....	7
Conclusión: .....	8
Bibliografía:.....	8

## Introducción:

En este documento se detalla la información relativa al informe de testing individual del integrante Joaquín Arregui Díaz. Se explicarán brevemente los test que han sido realizadas especificando una concisa descripción. También, en el segundo capítulo se expondrá capturas del tiempo que ha tardado tanto mi pc como el de un compañero en realizar todos los test. También haremos una comparación entre los tiempos que han tardado los dos pc en realizarlos.

## Contenido:

### TESTING FUNCIONAL:

#### Course:

1. *LecturerCourseCreateTest*: En las pruebas de "Create Positive", se han evaluado diversas situaciones límite y puntos críticos que podrían causar conflictos. Se han creado cursos válidos, asegurándose de abarcar todos los casos extremos posibles. Por otro lado, en las pruebas de "Create Negative", se han introducido datos en el archivo CSV que deberían generar errores al intentar llenar los campos de creación de un lecture. Estas pruebas han sido diseñadas para asegurar que el sistema detecte y maneje correctamente los escenarios inválidos. Además de las pruebas mencionadas anteriormente, se ha incluido un caso especial llamado "test300hacking". En este caso, se ha evaluado que no sea posible crear cursos con un rol diferente al de "Lecturer", lo que ayuda a garantizar la integridad y seguridad del sistema.
2. *LecturerCourseDeleteTest*: En las pruebas de "Delete Positive", se han creado una serie de cursos que deben ser borradas cuando se ejecute el test, asegurándose de abarcar todos los casos posibles. No se ha usado test negativo. Además de las pruebas mencionadas anteriormente, se ha incluido un caso especial llamado "test300hacking". En este caso, se ha evaluado que no sea posible eliminar cursos con un rol diferente al de "Lecturer", lo que ayuda a garantizar la integridad y seguridad del sistema. Un "test301hacking" que evalúa que no se puedan eliminar cursos con el draftmode false.
3. *LecturerCourseListAltTest*: En las pruebas de "List All Positive", se han evaluado diversas situaciones límite y puntos críticos que podrían causar conflictos. Se han creado cursos válidos, asegurándose de abarcar todos los casos extremos posibles. No se ha realizado test negativo. Además de las pruebas mencionadas anteriormente, se ha incluido un caso especial llamado "test300hacking". En este caso, se ha evaluado que no sea posible listar cursos con un rol diferente al de "Lecturer", lo que ayuda a garantizar la integridad y seguridad del sistema.
4. *LecturerCourseListMineTest*: En las pruebas de "List Mine Positive", se han evaluado diversas situaciones límite y puntos críticos que podrían causar conflictos. Se han creado cursos válidos, asegurándose de abarcar todos los casos extremos posibles. No se ha realizado test negativo. Además de las pruebas mencionadas anteriormente, se ha incluido un caso especial llamado "test300hacking". En este caso, se ha evaluado que no sea posible listar cursos con un rol diferente al de "Lecturer", lo que ayuda a garantizar la integridad y seguridad del sistema.

5. *LecturerCoursePublishTest*: En las pruebas de "Publish Positive", se han evaluado diversas situaciones límite y puntos críticos que podrían causar conflictos. Se han creado cursos válidos, asegurándose de abarcar todos los casos extremos posibles. Por otro lado, en las pruebas de "Publish Negative", se han introducido datos en el archivo CSV que deberían generar errores al intentar llenar los campos de creación de un curso. Estas pruebas han sido diseñadas para asegurar que el sistema detecte y maneje correctamente los escenarios inválidos. Además de las pruebas mencionadas anteriormente, se ha incluido un caso especial llamado "test300hacking". En este caso, se ha evaluado que no sea posible publicar cursos con un rol diferente al de "Lecturer", lo que ayuda a garantizar la integridad y seguridad del sistema. Un "test301hacking" que evalúa que no se puedan publicar cursos con el draftmode true, y por último, un "test302hacking" que hace el curso no se pueda publicar si no la ha registrado el mismo role.
6. *LecturerCourseShowTest*: En las pruebas de "Show Positive", se han evaluado diversas situaciones límite y puntos críticos que podrían causar conflictos. Se han creado cursos válidos, asegurándose de abarcar todos los casos extremos posibles. No se ha usado test negativo. Además de las pruebas mencionadas anteriormente, se ha incluido un caso especial llamado "test300hacking". En este caso, se ha evaluado que no sea posible mostrar cursos con un rol diferente al de "Lecturer", lo que ayuda a garantizar la integridad y seguridad del sistema.
7. *LecturerCourseUpdateTest*: En las pruebas de "Update Positive", se han evaluado diversas situaciones límite y puntos críticos que podrían causar conflictos. Se han creado cursos válidos, asegurándose de abarcar todos los casos extremos posibles. Por otro lado, en las pruebas de "Update Negative", se han introducido datos en el archivo CSV que deberían generar errores al intentar llenar los campos de creación de un curso. Estas pruebas han sido diseñadas para asegurar que el sistema detecte y maneje correctamente los escenarios inválidos. Además de las pruebas mencionadas anteriormente, se ha incluido un caso especial llamado "test300hacking". En este caso, se ha evaluado que no sea posible actualizar cursos con un rol diferente al de "Lecturer", lo que ayuda a garantizar la integridad y seguridad del sistema.

#### Lecture:

1. *LecturerLectureCreateTest*: En las pruebas de "Create Positive", se han evaluado diversas situaciones límite y puntos críticos que podrían causar conflictos. Se han creado conferencias válidas, asegurándose de abarcar todos los casos extremos posibles. Por otro lado, en las pruebas de "Create Negative", se han introducido datos en el archivo CSV que deberían generar errores al intentar llenar los campos de creación de una conferencia. Estas pruebas han sido diseñadas para asegurar que el sistema detecte y maneje correctamente los escenarios inválidos. Además de las pruebas mencionadas anteriormente, se ha incluido un caso especial llamado "test300hacking". En este caso, se ha evaluado que no sea posible crear conferencias con un rol diferente al de "Lecturer", lo que ayuda a garantizar la integridad y seguridad del sistema.

2. *LecturerLectureDeleteTest*: En las pruebas de "Delete Positive", se han creado una serie de conferencias que deben ser borradas cuando se ejecute el test, asegurándose de abarcar todos los casos posibles. No se ha usado test negativo. Además de las pruebas mencionadas anteriormente, se ha incluido un caso especial llamado "test3oohacking". En este caso, se ha evaluado que no sea posible eliminar conferencias con un rol diferente al de "Lecturer", lo que ayuda a garantizar la integridad y seguridad del sistema. Un "test3oihacking" que evalúa que no se puedan eliminar conferencias con el draftmode false.
3. *LecturerLectureListTest*: En las pruebas de "List Positive", se han evaluado diversas situaciones límite y puntos críticos que podrían causar conflictos. Se han creado conferencias válidas, asegurándose de abarcar todos los casos extremos posibles. No se ha realizado test negativo. Además de las pruebas mencionadas anteriormente, se ha incluido un caso especial llamado "test3oohacking". En este caso, se ha evaluado que no sea posible listar conferencias con un rol diferente al de "Lecturer", lo que ayuda a garantizar la integridad y seguridad del sistema.
4. *LecturerLecturePublishTest*: En las pruebas de "Publish Positive", se han evaluado diversas situaciones límite y puntos críticos que podrían causar conflictos. Se han creado conferencias válidas, asegurándose de abarcar todos los casos extremos posibles. Por otro lado, en las pruebas de "Publish Negative", se han introducido datos en el archivo CSV que deberían generar errores al intentar llenar los campos de creación de un curso. Estas pruebas han sido diseñadas para asegurar que el sistema detecte y maneje correctamente los escenarios inválidos. Además de las pruebas mencionadas anteriormente, se ha incluido un caso especial llamado "test3oohacking". En este caso, se ha evaluado que no sea posible publicar conferencias con un rol diferente al de "Lecturer", lo que ayuda a garantizar la integridad y seguridad del sistema. Un "test3oihacking" que evalúa que no se puedan publicar cursos con el draftmode true, y por último, un "test3o2hacking" que hace la conferencia no se pueda publicar si no la ha registrado el mismo role.
5. *LecturerLectureShowTest*: En las pruebas de "Show Positive", se han evaluado diversas situaciones límite y puntos críticos que podrían causar conflictos. Se han creado conferencias válidas, asegurándose de abarcar todos los casos extremos posibles. No se ha usado test negativo. Además de las pruebas mencionadas anteriormente, se ha incluido un caso especial llamado "test3oohacking". En este caso, se ha evaluado que no sea posible mostrar conferencias con un rol diferente al de "Lecturer", lo que ayuda a garantizar la integridad y seguridad del sistema.
6. *LecturerLectureUpdateTest*: En las pruebas de "Update Positive", se han evaluado diversas situaciones límite y puntos críticos que podrían causar conflictos. Se han creado conferencias válidas, asegurándose de abarcar todos los casos extremos posibles. Por otro lado, en las pruebas de "Update Negative", se han introducido datos en el archivo CSV que deberían generar errores al intentar llenar los campos de creación de una conferencia. Estas pruebas han sido diseñadas para asegurar que el sistema detecte y maneje correctamente los escenarios inválidos. Además de las pruebas mencionadas anteriormente, se ha incluido un caso especial llamado "test3oohacking". En este caso, se ha evaluado que no sea posible actualizar conferencias con un rol diferente al de "Lecturer", lo que ayuda a garantizar la integridad y seguridad del sistema.

## Membership:

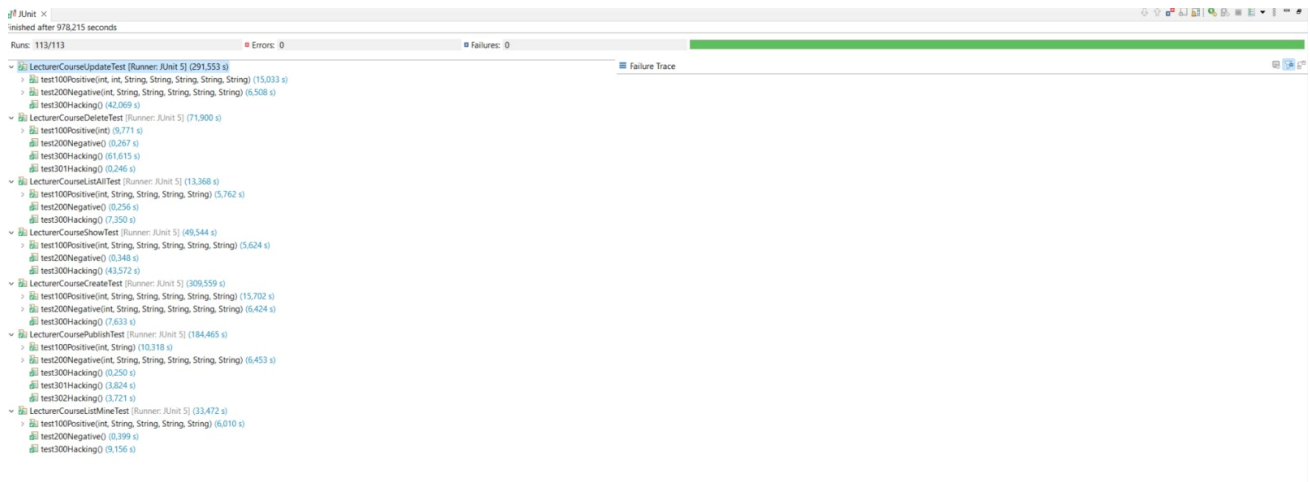
1. *LecturerMembershipCreateTest*: En las pruebas de "Create Positive", se han evaluado diversas situaciones límite y puntos críticos que podrían causar conflictos. Se han creado memberships válidas, asegurándose de abarcar todos los casos extremos posibles. Por otro lado, en las pruebas de "Create Negative", se han introducido datos en el archivo CSV que deberían generar errores al intentar llenar los campos de creación de una membership. Estas pruebas han sido diseñadas para asegurar que el sistema detecte y maneje correctamente los escenarios inválidos. Además de las pruebas mencionadas anteriormente, se ha incluido un caso especial llamado "test300hacking". En este caso, se ha evaluado que no sea posible crear memberships con un rol diferente al de "Lecturer", lo que ayuda a garantizar la integridad y seguridad del sistema.
2. *LecturerMembershipDeleteTest*: En las pruebas de "Delete Positive", se han creado una serie de memberships que deben ser borradas cuando se ejecute el test, asegurándose de abarcar todos los casos posibles. No se ha usado test negativo. Además de las pruebas mencionadas anteriormente, se ha incluido un caso especial llamado "test300hacking". En este caso, se ha evaluado que no sea posible eliminar memberships con un rol diferente al de "Lecturer", lo que ayuda a garantizar la integridad y seguridad del sistema. Un "test301hacking" que evalúa que no se puedan eliminar memberships con el draftmode false.
3. *LecturerMembershipListTest*: En las pruebas de "List Positive", se han evaluado diversas situaciones límite y puntos críticos que podrían causar conflictos. Se han creado memberships válidas, asegurándose de abarcar todos los casos extremos posibles. No se ha realizado test negativo. Además de las pruebas mencionadas anteriormente, se ha incluido un caso especial llamado "test300hacking". En este caso, se ha evaluado que no sea posible listar memberships con un rol diferente al de "Lecturer", lo que ayuda a garantizar la integridad y seguridad del sistema.
4. *LecturerMembershipShowTest*: En las pruebas de "Show Positive", se han evaluado diversas situaciones límite y puntos críticos que podrían causar conflictos. Se han creado memberships válidas, asegurándose de abarcar todos los casos extremos posibles. No se ha usado test negativo. Además de las pruebas mencionadas anteriormente, se ha incluido un caso especial llamado "test300hacking". En este caso, se ha evaluado que no sea posible mostrar memberships con un rol diferente al de "Lecturer", lo que ayuda a garantizar la integridad y seguridad del sistema.

## PRUEBAS DE RENDIMIENTO:

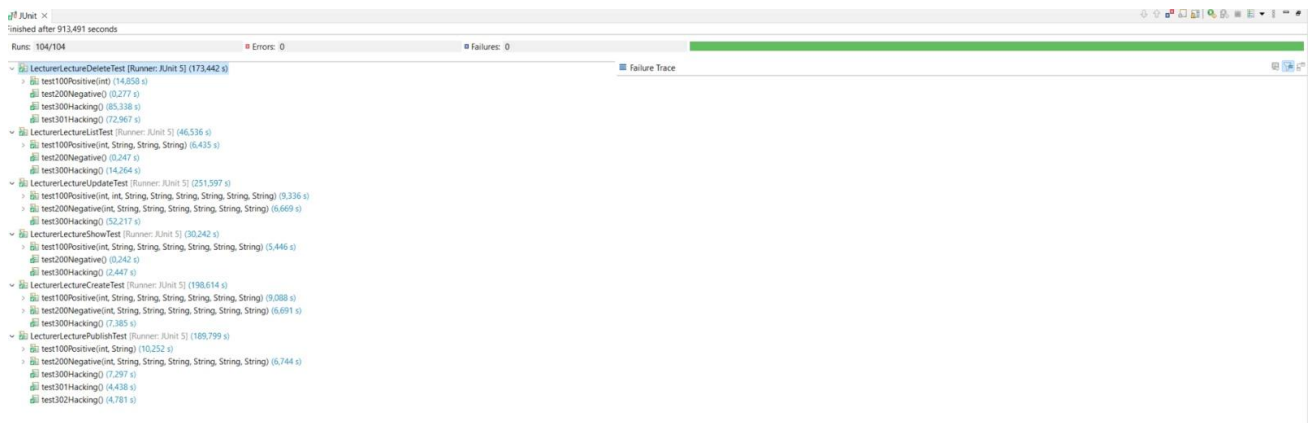
En este capitulo se mostraran una serie de capturas de las pruebas realizadas, tanto en mi ordenador, como en el de un compañero. Al final realizaré una breve conclusión de los resultados obtenidos.

### Ordenador de Joaquín Arregui Diaz:

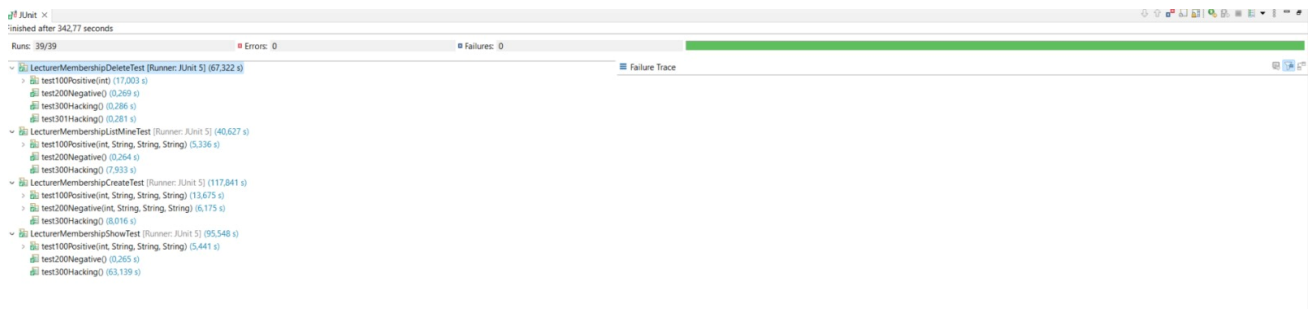
#### Tests de Course:



#### Tests de Lecture:

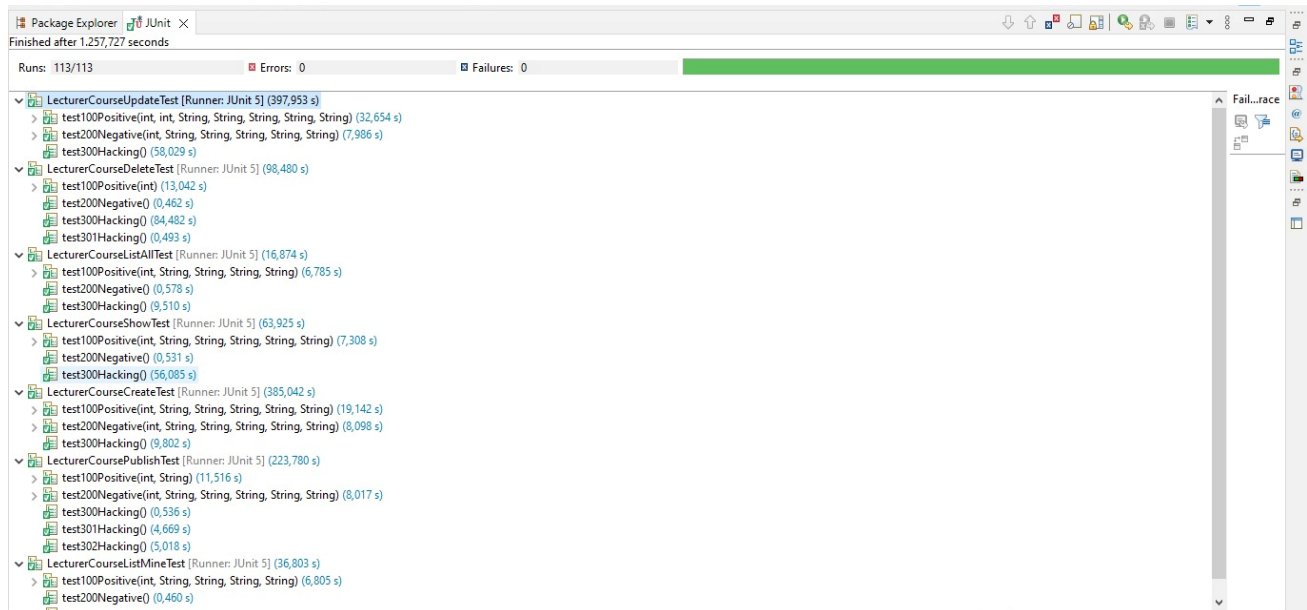


#### Tests de Membership:

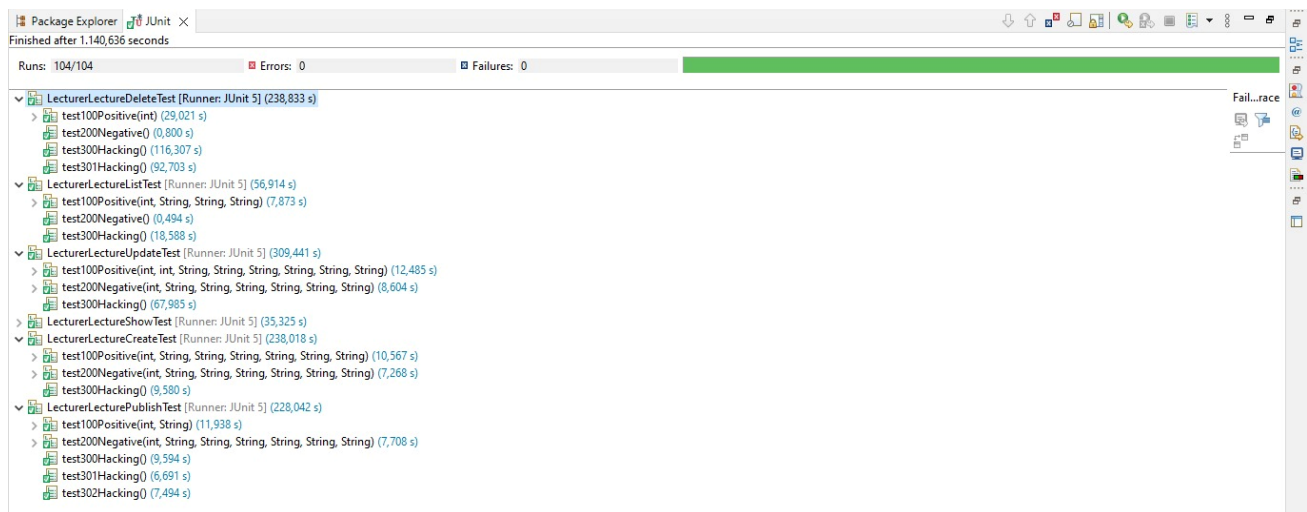


## Ordenador de Francisco Antonio Campos Campos:

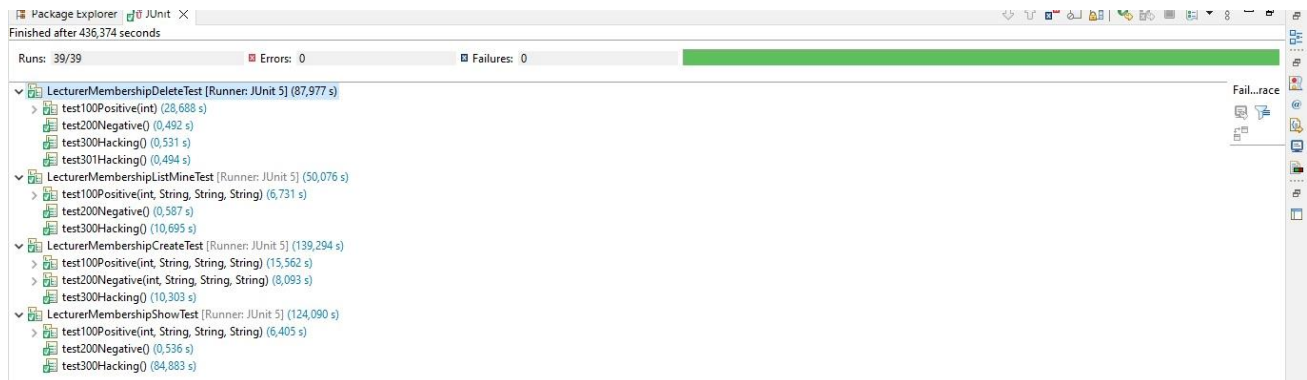
### Tests de Course:



### Tests de Lecture:



### Tests de Membership:





Al analizar ambos equipos, se observa que los resultados de las pruebas fueron similares, ya que ambos arrojaron resultados positivos en todos los tests. No obstante, es importante destacar la diferencia de tiempo en la ejecución de las pruebas, siendo el equipo mi equipo aproximadamente 7 minutos más rápido. Esta disparidad de tiempo se debe a la diferencia en la antigüedad de los equipos: mi ordenador es del año 2022, mientras que el de Joaquín es del año 2014. A pesar de esta disparidad, ambos equipos realizaron las pruebas de manera adecuada y obtuvieron los mismos resultados.

## Conclusión:

En conclusión, este informe de pruebas ha sido muy exhaustivo y ha demostrado un rendimiento sobresaliente en todos los aspectos evaluados. Se han incluido numerosos casos positivos, especialmente en los extremos, que representan situaciones conflictivas y desafiantes. Sin embargo, el sistema ha superado todos estos casos de prueba de manera correcta, lo que destaca su capacidad para manejar escenarios complejos de manera exitosa. En resumen, el sistema ha demostrado una buena capacidad de respuesta para superar los casos más difíciles, en los que es fácil fallar, y ha pasado todas las pruebas de manera exitosa. Estos resultados reflejan una buena calidad del sistema evaluado en este informe de pruebas.

## Bibliografía:

- Annexes.docx de la Enseñanza Virtual de la Universidad de Sevilla.