Diseño y Pruebas II

Testing Report

C2.018

David Gavira Serrano

davgavser@alum.us.es

https://github.com/davidg43/Acme-SF-D 

Contenido

[Tabla de versiones 3](#_Toc171354966)

[Tabla de revisiones 3](#_Toc171354967)

[Resumen Ejecutivo 3](#_Toc171354968)

[Introducción 4](#_Toc171354969)

[Contenido 5](#_Toc171354970)

[Pruebas funcionales 5](#_Toc171354971)

[Entidad Project: 6](#_Toc171354972)

[LIST 6](#_Toc171354973)

[Show 6](#_Toc171354974)

[Update 6](#_Toc171354975)

[Create 7](#_Toc171354976)

[Delete 7](#_Toc171354977)

[Publish 8](#_Toc171354978)

[Entidad User Story: 8](#_Toc171354979)

[LIST 8](#_Toc171354980)

[Show 9](#_Toc171354981)

[Update 9](#_Toc171354982)

[Create 9](#_Toc171354983)

[Delete 10](#_Toc171354984)

[Publish 10](#_Toc171354985)

[Entidad Assignment: 10](#_Toc171354986)

[LIST 11](#_Toc171354987)

[Show 11](#_Toc171354988)

[Update 11](#_Toc171354989)

[Create 12](#_Toc171354990)

[Delete 12](#_Toc171354991)

[Cobertura de las pruebas 13](#_Toc171354992)

[Análisis de rendimiento 14](#_Toc171354993)

[Análisis sin índices 14](#_Toc171354994)

[Análisis con índices 16](#_Toc171354995)

[Contraste de hipótesis 18](#_Toc171354996)

[Prueba en equipo diferente 19](#_Toc171354997)

[Conclusión 21](#_Toc171354998)

|  |  |  |
| --- | --- | --- |
| Tabla de versiones | | |
| Versión | Fecha | Descripción |
| 1.0 | 27/05/2024 | Versión inicial del documento. |
| 2.0 | 07/07/2024 | Modificaciones D01-D04 |

# 

|  |  |  |
| --- | --- | --- |
| Tabla de revisiones | | |
| N. Revisión | Fecha | Descripción |
| 1 | 27/05/2024 | Preparado para entrega |
| 2 | 07/07/2024 | Modificaciones D01-D04 |

# Resumen Ejecutivo

El informe presente detalla el proceso de "Formal Testing" para la asignatura de "Diseño y Pruebas II", liderado por David Gavira Serrano. Se ha llevado a cabo un testeo exhaustivo de las implementaciones requeridas, siguiendo rigurosamente la metodología impartida en la asignatura. Sin embargo, lamentamos informarle que el proyecto se ha retrasó. Nos disculpamos sinceramente por cualquier inconveniente que esto pueda haber causado.

Queremos destacar que hemos trabajado intensamente para incorporar su feedback, con el objetivo de asegurar una presentación impecable de la app. A pesar de los desafíos organizativos y técnicos, hemos avanzado satisfactoriamente.

El trabajo en equipo ha sido fundamental y absolutamente necesario para la correcta realización de los requerimientos esenciales, asegurando el cumplimiento de los criterios establecidos por la evaluación. Este informe proporciona una visión integral del proceso de testing y ofrece recomendaciones para mejorar futuras actividades de prueba y desarrollo en proyectos similares.

# Introducción

El presente documento ofrece una retrospectiva detallada del Sprint "Formal Testing" para la asignatura de "Diseño y Pruebas II", dirigido por David Gavira Serrano, durante su ejecución a lo largo de un período definido. Se presenta un análisis exhaustivo de las pruebas realizadas, contrastando los resultados obtenidos con las expectativas iniciales y destacando la importancia de la metodología seguida.

Lamentamos informar que el proyecto ha sufrido retrasos. Nos disculpamos sinceramente por cualquier inconveniente que esto pueda haber causado. Sin embargo, queremos asegurarle que hemos trabajado arduamente para incorporar su feedback, y en la entrega de la segunda convocatoria, el cliente finalmente tendrá la app esperada de manera correcta.

El informe aborda al detalle las diferentes pruebas realizadas, identificando hitos clave, desviaciones significativas y áreas de mejora. Además, se comentan los resultados de las diferentes pruebas realizadas y cómo el rendimiento del PC puede influir en la ejecución de las mismas. Se abordan también los conflictos y desafíos encontrados durante esta fase del proyecto, incluyendo problemas técnicos, y cómo se resolvieron para asegurar la consecución de los objetivos establecidos. Este documento tiene como objetivo proporcionar una comprensión integral del proceso de testing formal, así como ofrecer lecciones aprendidas y recomendaciones para futuras iniciativas similares.

# Contenido

## Pruebas funcionales

Se han realizado tanto pruebas positivas y negativas (<feature>.safe) como pruebas de hacking (<feature>.hack), siguiendo rigurosamente las metodologías establecidas en el documento L04-S01 – Formal Testing para la asignatura. Estas pruebas han permitido evaluar la robustez y seguridad de cada característica de la aplicación.

A continuación, clasificaremos las pruebas por entidad. Esta clasificación nos permitirá profundizar en las diferencias y particularidades de cada una de las funcionalidades, proporcionando un análisis detallado y específico para cada caso.

Se entregará junto a este documento una carpeta llamada **Coverage Info** en la que encontraremos una impresión de cada una de estas clases con la información sobre la cobertura obtenida. Con el afán de hacer este documento mucho mas liviano se tendrán las capturas con la cobertura como un PDF autónomo impreso que se podrá encontrar en una jerarquía de carpetas por entidad en la carpeta indicada.

## Entidad Project:

Se han realizado las pruebas de manera exhaustiva y además se han aportado gran variedad de datos de pruebas en los archivos “.csv” iniciales para el funcionamiento correcto de la aplicación.

Además, se ha intentado acceder de manera fraudulenta sin éxito a las diferentes URL y vistas de las diferentes funcionalidades. Esto se ha realizado en los archivos “.hack’’.

### LIST

Listar proyectos de un manager, incluyendo la posibilidad de obtener un listado vacío



Como podemos ver en el archivo anterior, se han realizado el análisis de cobertura de manera muy exitosa teniendo una cobertura de 96%. Esto nos indica que se ha completado casi en su totalidad el número de instrucciones del archivo.

Las líneas resaltadas en amarillo nos indican verificaciones de seguridad que no se han ejecutado puesto que este hecho indicaría un fallo en el flujo de uso de la aplicación.

Show

Mostrar detalles de los proyectos, ya sean publicados o no publicados.



Al igual que en el caso anterior obtenemos una cobertura casi completa, de un 97.4%

### Update

Actualizar proyectos no publicados, probando todos los campos del formulario con una variedad de valores, incluyendo valores positivos y negativos rechazados por el sistema (valores nulos, repetidos, duplicados, URLs incorrectas, etc.), asegurando que se respetan los límites establecidos.

Confirmar que el proyecto ha sido actualizado correctamente.

Verificar que se pueden ver los detalles del proyecto actualizado.



En este caso comprobamos que la cobertura ha bajado un poco, esto no es mas que el hecho de realizar varias operaciones como la descrita anteriormente. En este tipo de feature se realizan comprobaciones de seguridad más excautivas para el buen uso de la aplicación.

### Create

Crear proyectos, probando todos los campos del formulario con una variedad de valores, incluyendo valores positivos y negativos rechazados por el sistema (valores nulos, repetidos, duplicados, URLs incorrectas, etc.), asegurando que se respetan los límites establecidos.

Confirmar que el proyecto ha sido creado correctamente.

Verificar que se pueden ver los detalles del nuevo proyecto.



Aquí encontramos una cobertura muy similar a la de la funcionalidad anterior.

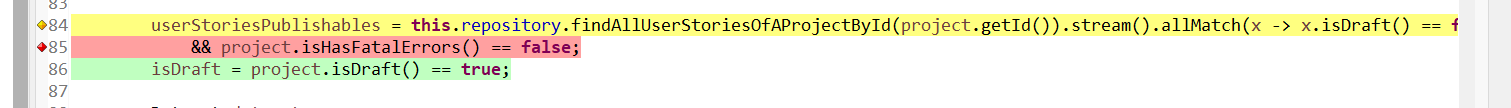
### Delete

Eliminar proyectos no publicados.

Confirmar que el proyecto ha sido eliminado correctamente.



En este caso no hemos conseguido subir mas la cobertura debido a que en:



(consultar el doc PjtDelete para verlo mejor).

Claramente se aprecia como no es posible ejecutar esta línea de código a la hora de borrar un proyecto debido a que estas instrucciones se ejecutan en el caso de que quisiéramos realizar alguna otra acción (publish, update) desde la vista de la funcionalidad delete. [manager/project/delete?idX]

### Publish

Actualizar y publicar proyectos, no publicados, probando todos los campos del formulario con una variedad de valores, incluyendo valores positivos y negativos rechazados por el sistema (valores nulos, repetidos, duplicados, URLs incorrectas, etc.), asegurando que se respetan los límites establecidos.

Verificar restricciones al intentar publicar un proyecto, asegurando que no se puede publicar si no cumple con las necesidades básicas para esto.

Confirmar que el proyecto ha sido publicado correctamente.

Verificar que se pueden ver los detalles del proyecto publicado.



Finalmente encontramos el método publish, quería comentar que este método y su funcionamiento fue cambiado tras las recomendaciones y valoraciones del cliente en el tablero de discusiones de la comunidad. Se ha permitido la modificación de todos los valores como si de un update se tratase, y posterior a esto se realiza la publicación ejecutándose las mismas validaciones que en las otras funcionalidades (update)

## Entidad User Story:

Se han realizado las pruebas de manera exhaustiva y además se han aportado gran variedad de datos de pruebas en los archivos “.csv” iniciales para el funcionamiento correcto de la aplicación.

Además, se ha intentado acceder de manera fraudulenta sin éxito a las diferentes URL y vistas de las diferentes funcionalidades. Esto se ha realizado en los archivos “.hack’’.

### LIST

Listar las historias de usuario de un mánager, incluyendo la posibilidad de obtener un listado vacío



Se ha obtenido unos resultados similares a sus funcionalidades homólogas en las otras entidades.

Show

Mostrar detalles de las historias de usuario, ya sean publicadas o no publicadas.



Al igual que en el caso anterior obtenemos una cobertura casi completa, de un 96.7%

### Update

Actualizar historias de usuario no publicadas, probando todos los campos del formulario con una variedad de valores, incluyendo valores positivos y negativos rechazados por el sistema (valores nulos, repetidos, duplicados, URLs incorrectas, etc.), asegurando que se respetan los límites establecidos.

Confirmar que la historia de usuario ha sido actualizada correctamente.

Verificar que se pueden ver los detalles de la historia de usuario.



En este caso comprobamos que la cobertura ha bajado un poco, esto no es más que el hecho de realizar varias operaciones como la descrita anteriormente. En este tipo de feature se realizan comprobaciones de seguridad más excautivas para el buen uso de la aplicación.

### Create

Crear historias de usuario, probando todos los campos del formulario con una variedad de valores, incluyendo valores positivos y negativos rechazados por el sistema (valores nulos, repetidos, duplicados, URLs incorrectas, etc.), asegurando que se respetan los límites establecidos.

Confirmar que la historia de usuario ha sido creado correctamente.

Verificar que se pueden ver los detalles de la nueva historias de usuario.



Aquí encontramos una cobertura muy similar a la de la funcionalidad anterior.

### Delete

Eliminar historias de usuario no publicadas.

Confirmar que la historia de usuario ha sido eliminada correctamente.



Aquí encontramos una cobertura muy similar a la de la funcionalidad anterior.

### Publish

Actualizar y publicar historias de usuario, no publicadas, probando todos los campos del formulario con una variedad de valores, incluyendo valores positivos y negativos rechazados por el sistema (valores nulos, repetidos, duplicados, URLs incorrectas, etc.), asegurando que se respetan los límites establecidos.

Verificar restricciones al intentar publicar un proyecto, asegurando que no se puede publicar si no cumple con las necesidades básicas para esto.

Confirmar que la historia de usuario ha sido publicado correctamente.

Verificar que se pueden ver los detalles de la historia de usuario publicada.



Finalmente encontramos el método publish, hemos obtenido una cobertura bastante similar a las otras funcionalidades.

## Entidad Assignment:

La entidad Assignment sirve como intermediaria para establecer una relación de muchos a muchos entre los proyectos y las historias de usuario. Esta entidad facilita la gestión de tareas y la asignación eficiente de recursos en la aplicación.

Debido a la necesidad de mejorar la usabilidad para el usuario y proporcionar al cliente una experiencia más fluida y eficiente, hemos ajustado el proceso de creación de estas entidades. Estos cambios buscan simplificar el flujo de trabajo y aumentar la funcionalidad y usabilidad de la aplicación final.

Se han realizado las pruebas de manera exhaustiva y además se han aportado gran variedad de datos de pruebas en los archivos “.csv” iniciales para el funcionamiento correcto de la aplicación.

Además, se ha intentado acceder de manera fraudulenta sin éxito a las diferentes URL y vistas de las diferentes funcionalidades. Esto se ha realizado en los archivos “.hack’’.

### LIST

Listar asignaciones, mostrando todas las asignaciones disponibles entre proyectos e historias de usuario, incluyendo la posibilidad de obtener un listado vacío.



Cabe destacar que el listado de los assigments se realiza tras acceder a los detalles de uno de los proyectos. Es desde aquí donde nos encontramos con un botón que nos llevará a la lista de assigment vinculados a un proyecto en específico.

Encontramos una cobertura casi completa, de 95.4%.

Show

Mostrar detalles de una asignación específica, incluyendo la información del proyecto y la historia de usuario vinculada.



Al igual que en el caso anterior obtenemos una cobertura casi completa, de un 95.8%

### Update

Actualizar asignaciones existentes, probando todos los campos del formulario con una variedad de valores, incluyendo positivos y negativos rechazados por el sistema.

Confirmar que la asignación ha sido actualizada correctamente.

Verificar que se pueden ver los detalles de la asignación actualizada.



En este caso comprobamos que la cobertura ha bajado un poco, esto no es más que el hecho de realizar varias operaciones como la descrita anteriormente. En este tipo de feature se realizan comprobaciones de seguridad más excautivas para el buen uso de la aplicación.

### Create

Crear nuevas asignaciones, probando todos los campos del formulario con una variedad de valores, incluyendo valores positivos y negativos rechazados por el sistema (valores nulos, repetidos, duplicados, etc.).

Confirmar que la asignación ha sido creada correctamente.

Verificar que se pueden ver los detalles de la nueva asignación.



Aquí encontramos una cobertura muy similar a la de la funcionalidad anterior.

### Delete

Eliminar asignaciones existentes.

Confirmar que la asignación ha sido eliminada correctamente.



Encontramos una cobertura bastante alta aunque menor que el 95%.

Para concluir este capitulo del informe, parece destacable el hecho de haber conseguido una cobertura que oscila entre el 90% y el 100%, en la mayoría de los archivos aportados se podrá observar que se ha reducido al mínimo las líneas no recorridas. Sin embargo, como se ha comentado ya en varias ocasiones, estos resultados llegan al cliente con retraso y debemos estar seguro de que la aplicación se encuentra estable y cumple con los criterios mínimos de aceptación.

### Cobertura de las pruebas

Para evitar alargar excesivamente este documento, se ha preferido aportar pruebas gráficas que comentaremos posteriormente. Además como se ha comentado antes, en las PDFs podemos encontrar la cobertura de cada uno de los archivos recorridos.

Tabla

Descripción generada automáticamente

Es importante mencionar que la cobertura es buena pero que estaría bien mejorarla para superar el 95%, para esto sería necesario realizar comprobaciones de seguridad para evitar errores graves durante el flujo de uso de la aplicación de algún otro modo. Pese a esto, se ha intentado ejecutar una gran cantidad de pruebas para que todas las líneas fueran recorridas.

## Análisis de rendimiento

En primer lugar, veremos cómo ha sido el rendimiento de las pruebas antes y después de las modificaciones de código recomendadas por el equipo docente de la asignatura. En este caso se han implementado índices en la base de datos para agilizar las consultas y reducir el tiempo para satisfacer los requisitos acordados.

### Análisis sin índices

Siguiendo los criterios recomendados se ha generado un informe sobre los tiempos de cada una de las peticiones agrupadas por su URL.

Esta medición se ha realizado después de ejecutar las pruebas realizadas sin índices en las entidades.

Tabla

Descripción generada automáticamente

Gráfico, Gráfico de barras, Gráfico en cascada

Descripción generada automáticamente

### Análisis con índices

Siguiendo los criterios recomendados se ha generado un informe sobre los tiempos de cada una de las peticiones agrupadas por su URL.

Esta medición se ha realizado después de ejecutar las pruebas realizadas con índices en las entidades.

|  |  |  |
| --- | --- | --- |
|  |  |  |

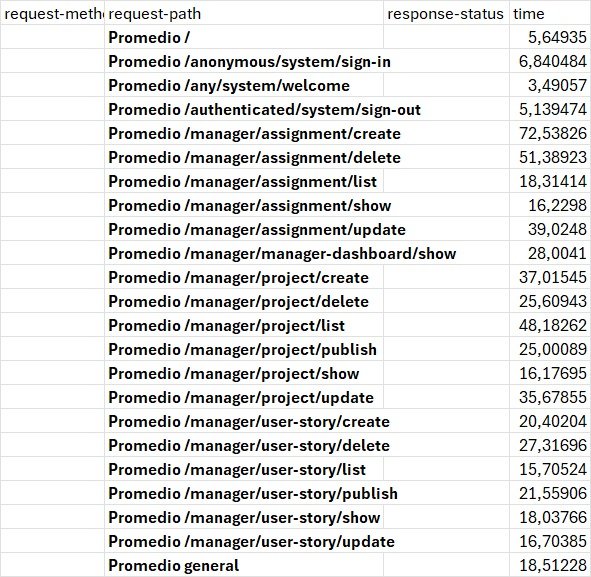


Imagen que contiene Gráfico en cascada

Descripción generada automáticamente

### Contraste de hipótesis

Interfaz de usuario gráfica, Tabla

Descripción generada automáticamente

Tabla, Excel

Descripción generada automáticamente

Como hemos estudiado en el contexto de la asignatura, es posible afirmar que no hay cambios medibles ya que en un intervalo de confianza del 95% el p-valor calculado es mayor que “1-95% = 0.05”.

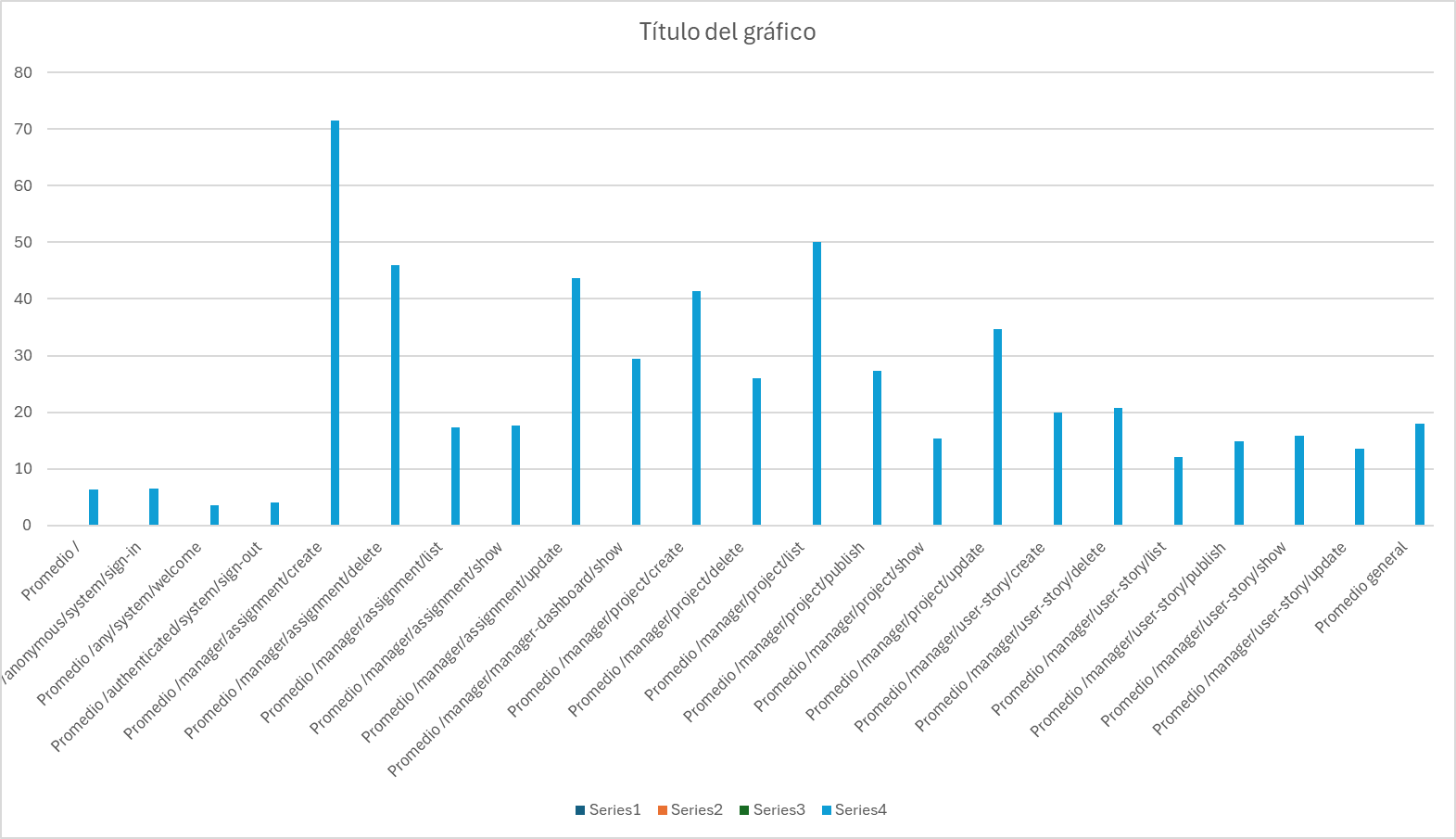
p=0.7253  
p>0.05

### Prueba en equipo diferente

Esta prueba se ha realizado en el PC de Luis Garcia Parras, el Student 2 del proyecto, esto se ha realizado evitando tener otro programa alguno en paralelo tratando de intentar conseguir las condiciones ideales.



|  |  |
| --- | --- |
| *PC-Luis* | |
|  |  |
| Media | 18,0069582 |
| Error típico | 0,70598865 |
| Mediana | 11,11 |
| Moda | 2,3185 |
| Desviación estándar | 22,4255604 |
| Varianza de la muestra | 502,905759 |
| Curtosis | 42,6483964 |
| Coeficiente de asimetría | 4,8453726 |
| Rango | 280,579 |
| Mínimo | 1,9936 |
| Máximo | 282,5726 |
| Suma | 18169,0208 |
| Cuenta | 1009 |
| Nivel de confianza(95,0%) | 1,3853758 |



Hemos de indicar que estas mediciones en el PC-Luis se han realizado sin los índices en la base de datos, por lo tanto, para compararlos con los nuestros en el mismo estado deberíamos llevar a cabo el mismo proceso que anteriormente.

Tabla

Descripción generada automáticamente

Por segunda vez, es posible afirmar que no hay cambios medibles ya que en un intervalo de confianza del 95% el p-valor calculado es mayor que “1-95% = 0.05”.

p=0.845821  
p>0.05

# Conclusión

El proceso de "Formal Testing" realizado en el marco de la asignatura "Diseño y Pruebas II", bajo la dirección de David Gavira Serrano, ha sido esencial para garantizar la calidad y fiabilidad del producto desarrollado. A través de una metodología rigurosa y un enfoque exhaustivo, hemos podido identificar y corregir errores críticos que podrían haber afectado negativamente la experiencia del usuario y la reputación del producto.

Sentimos el retraso requerido para cumplir con los criterios de aceptación del cliente. En este caso, creemos haber cumplido con los requisitos y estamos deseando conocer su feedback.

La colaboración en equipo ha sido fundamental para cumplir con los requisitos y superar los desafíos técnicos y organizativos. Las pruebas realizadas, tanto funcionales como de hacking, han cubierto una amplia gama de operaciones, asegurando que el sistema responda adecuadamente bajo diversas condiciones. La implementación de índices en la base de datos y la utilización de herramientas estadísticas, como el Z-Test y el intervalo de confianza, han permitido evaluar de manera objetiva el rendimiento y la estabilidad de la aplicación, confirmando que las mejoras realizadas son significativas y no producto del azar.

En definitiva, el testing formal ha sido una piedra angular en el desarrollo del proyecto, asegurando que el producto final sea robusto, fiable y capaz de satisfacer las expectativas y necesidades del usuario. Este nivel de escrutinio no solo mejora la calidad del producto, sino que también genera una mayor confianza entre los usuarios y clientes, proporcionando una base sólida para futuros desarrollos y pruebas.