

Grupo: C1.015

Repositorio: <https://github.com/jormunrod/Acme-SF-24.1.1>

Testing report

Ángel García Escudero (anggaesc1@alum.us.es)

Sevilla, 27-5-2024

Contenido

1.	Resumen ejecutivo	2
2.	Tabla de revisión	3
3.	Introducción.....	4
4.	Contenidos.....	5
4.1.	Pruebas funcionales.....	5
4.1.1.	Listado de Casos de Prueba implementados	5
4.2.	Pruebas de rendimiento	9
4.2.1.	Metodología	9
4.2.2.	Resultados de las pruebas de rendimiento	9
4.2.2.1.	Gráficas de tiempos de respuesta.....	9
4.2.2.2.	Análisis del intervalo de confianza.....	12
4.2.2.3.	Contraste de hipótesis	12
5.	Conclusiones.....	13
6.	Bibliografía.....	14

1. Resumen ejecutivo

Este documento analiza las pruebas funcionales y de rendimiento realizadas en el proyecto de gestión de proyectos e historias de usuario.

Para la realización de las pruebas funcionales, se han realizado 24 casos de prueba para verificar la funcionalidad de proyectos e historias de usuario, detectando un bug que fue corregido.

En cuanto a las pruebas de rendimiento han sido realizadas utilizando el ordenador propio y otro ordenador para evaluar los tiempos de respuesta de ambos ordenadores.

2. Tabla de revisión

Nº Revisión	Fecha	Descripción
1	27/05/2024	Versión inicial

3. Introducción

Este informe presenta los resultados de las pruebas funcionales y de rendimiento realizadas en el proyecto. Las pruebas funcionales se enfocaron en verificar la correcta operación de las funcionalidades de proyectos e historias de usuario, identificando y documentando bugs relevantes. Las pruebas de rendimiento evaluaron el tiempo de respuesta del sistema en dos ordenadores distintos, proporcionando un análisis detallado con intervalos de confianza y un contraste de hipótesis. Los resultados demuestran que el otro ordenador presenta un mejor desempeño en comparación con el ordenador propio, recomendándose su uso para tareas críticas que requieren tiempos de respuesta mínimos.

4. Contenidos

4.1. Pruebas funcionales

Esta sección detalla los casos de prueba funcionales implementados para las funcionalidades del sistema, específicamente para los módulos de proyecto e historias de usuario. Cada caso de prueba se describe brevemente, incluyendo su propósito, método de ejecución y efectividad en la detección de errores. Los casos de prueba están agrupados por funcionalidad para facilitar la comprensión y organización.

4.1.1. Listado de Casos de Prueba implementados

Funcionalidad: Proyectos

- **Caso de prueba 1 (list.safe).** Prueba del listado de proyectos.
Descripción. Se comprueba que un manager pueda listar sus proyectos.
Efectividad. Se detectaron 0 bugs.
- **Caso de prueba 2 (show.safe).** Prueba de los detalles de un proyecto.
Descripción. Se comprueba que un manager pueda visualizar los detalles de sus proyectos.
Efectividad. Se detectaron 0 bugs.
- **Caso de prueba 3 (create.safe, create-2.safe).** Prueba de la creación de un proyecto.
Descripción. Se comprueba que un manager pueda crear un proyecto. Para ello, antes de la creación de un proyecto que cumpla con todas las restricciones, se han probado los casos negativos para los extremos inferiores y superiores. Posteriormente, se han creado siete proyectos adicionales para probar los casos positivos para los extremos inferiores y superiores y para las pruebas con script hacking, SQL injection y caracteres especiales.
Efectividad. Se detectaron 0 bugs.
- **Caso de prueba 4 (update.safe, update-2.safe).** Prueba de la actualización de un proyecto.
Descripción. Se comprueba que un manager pueda actualizar la información de un proyecto asociado. Para ello, antes de actualizar un proyecto que cumpla con todas las restricciones (comprobando también que este no haya sido publicado), se han probado los casos negativos para los extremos inferiores y superiores. Posteriormente, se han creado siete proyectos adicionales para probar los casos positivos para los extremos inferiores y superiores y para las pruebas con script hacking, SQL injection y caracteres especiales.
Efectividad. Se detectaron 0 bugs.
- **Caso de prueba 5 (delete.safe).** Prueba de la eliminación de un proyecto.
Descripción. Se comprueba que un manager pueda eliminar un proyecto asociado y que no haya sido publicado.
Efectividad. Se detectaron 0 bugs.
- **Caso de prueba 6 (publish.safe).** Prueba de la publicación de un proyecto.
Descripción. Se comprueba que un manager pueda publicar un proyecto asociado. Para ello, antes de actualizar un proyecto que cumpla con todas las restricciones

(comprobando también que no contenga errores fatales, que tenga asociado al menos una historia de usuario y que todas las historias de usuario asociadas hayan sido publicadas previamente), se han probado los casos negativos para los extremos inferiores y superiores. Posteriormente, se han creado siete proyectos adicionales para probar los casos positivos para los extremos inferiores y superiores y para las pruebas con script hacking, SQL injection y caracteres especiales.

Efectividad. Se detectaron 0 bugs.

- **Caso de prueba 7 (list.hack).** Prueba del listado de proyectos accediendo con otro rol distinto a manager.

Descripción. Se comprueba que un usuario logado con un rol distinto de manager no tenga autorización para mostrar el listado de proyectos.

Efectividad. Se detectaron 0 bugs.

- **Caso de prueba 8 (show.hack).** Prueba de los detalles de un proyecto accediendo con otro usuario del mismo rol y con otro rol distinto a manager.

Descripción. Se comprueba que un usuario diferente logado con el rol de manager no tenga autorización para mostrar los detalles de un proyecto asociado a otro manager. De la misma forma, se comprueba que un usuario logado con un rol distinto a manager tampoco tenga autorización.

Efectividad. Se detectaron 0 bugs.

- **Caso de prueba 9 (create.hack).** Prueba de la creación de un proyecto accediendo con otro rol distinto a manager.

Descripción. Se comprueba que un usuario logado con un rol distinto de manager no tenga autorización para crear un nuevo proyecto.

Efectividad. Se detectaron 0 bugs.

- **Caso de prueba 10 (update.hack).** Prueba de la actualización de un proyecto accediendo con otro usuario del mismo rol y con otro rol distinto a manager.

Descripción. Se comprueba que un usuario diferente logado con el rol de manager no tenga autorización para actualizar un proyecto asociado a otro manager. De la misma forma, se comprueba que un usuario logado con un rol distinto a manager tampoco tenga autorización. Así mismo, se comprueba que el propio usuario no pueda actualizar un proyecto propio que ya haya sido publicado.

Efectividad. Se detectaron 0 bugs.

- **Caso de prueba 11 (delete.hack).** Prueba de la eliminación de un proyecto accediendo con otro usuario del mismo rol y con otro rol distinto a manager.

Descripción. Se comprueba que un usuario diferente logado con el rol de manager no tenga autorización para eliminar un proyecto asociado a otro manager, tanto si ha sido publicado como si no. De la misma forma, se comprueba que un usuario logado con un rol distinto a manager tampoco tenga autorización. Así mismo, se comprueba que el propio usuario no pueda eliminar un proyecto propio que ya haya sido publicado.

Efectividad. Se detectaron 0 bugs.

- **Caso de prueba 12 (publish.hack).** Prueba de la publicación de un proyecto accediendo con otro usuario del mismo rol y con otro rol distinto a manager.

Descripción. Se comprueba que un usuario diferente logado con el rol de manager no tenga autorización para publicar un proyecto asociado a otro manager. De la misma forma, se comprueba que un usuario logado con un rol distinto a manager tampoco tenga autorización. Así mismo, se comprueba que el propio usuario no pueda publicar un proyecto propio que ya haya sido publicado.

Efectividad. Se detectaron 0 bugs.

Funcionalidad: Historias de Usuario

- **Caso de prueba 1 (list.safe).** Prueba del listado de las historias de usuario de asociadas a un proyecto propio.

Descripción. Se comprueba que un manager pueda listar las historias de usuario asociadas a un proyecto propio.

Efectividad. Se detectaron 0 bugs.

- **Caso de prueba 2 (show.safe).** Prueba de los detalles de una historia de usuario asociada a un proyecto propio.

Descripción. Se comprueba que un manager pueda visualizar los detalles de una historia de usuario asociada a un proyecto propio.

Efectividad. Se detectaron 0 bugs.

- **Caso de prueba 3 (create.safe).** Prueba de la creación de una historia de usuario.

Descripción. Se comprueba que un manager pueda crear una historia de usuario. Para ello, antes de la creación de una historia de usuario que cumpla con todas las restricciones, se han probado los casos negativos para los extremos inferiores y superiores. Posteriormente, se han creado siete historias de usuario adicionales para probar los casos positivos para los extremos inferiores y superiores y para las pruebas con script hacking, SQL injection y caracteres especiales.

Efectividad. Se detectaron 0 bugs.

- **Caso de prueba 4 (update.safe).** Prueba de la actualización de una historia de usuario.

Descripción. Se comprueba que un manager pueda actualizar la información de una historia de usuario de un proyecto asociado. Para ello, antes de actualizar una historia de usuario que cumpla con todas las restricciones (comprobando también que el proyecto y la propia historia de usuario no hayan sido publicados), se han probado los casos negativos para los extremos inferiores y superiores. Posteriormente, se han creado siete proyectos adicionales para probar los casos positivos para los extremos inferiores y superiores y para las pruebas con script hacking, SQL injection y caracteres especiales.

Efectividad. Se detectó 1 bug relativo a la autorización de la actualización de una historia de usuario que ya había sido publicada.

- **Caso de prueba 5 (delete.safe).** Prueba de la eliminación de una historia de usuario asociada a un proyecto propio.

Descripción. Se comprueba que un manager pueda eliminar una historia de usuario, que no haya sido publicada, asociada a un proyecto propio que no haya sido publicado.

Efectividad. Se detectó 1 bug relativo a la autorización de la eliminación de una historia de usuario que ya había sido publicada.

- **Caso de prueba 6 (publish.safe).** Prueba de la publicación de una historia de usuario asociada a un proyecto propio.
Descripción. Se comprueba que un manager pueda publicar una historia de usuario asociada a un proyecto propio. Para ello, antes de publicar una historia de usuario que cumpla con todas las restricciones, se han probado los casos negativos para los extremos inferiores y superiores. Posteriormente, se han creado siete historias de usuario adicionales para probar los casos positivos para los extremos inferiores y superiores y para las pruebas con script hacking, SQL injection y caracteres especiales.
Efectividad. Se detectaron 0 bugs.
- **Caso de prueba 7 (list.hack).** Prueba del listado de las historias de usuario asociadas a un proyecto accediendo con otro rol distinto a manager.
Descripción. Se comprueba que un usuario logado con un rol distinto de manager no tenga autorización para mostrar el listado de historias de usuario asociadas a un proyecto.
Efectividad. Se detectaron 0 bugs.
- **Caso de prueba 8 (show.hack).** Prueba de los detalles de una historia de usuario asociada a un proyecto accediendo con otro usuario del mismo rol y con otro rol distinto a manager.
Descripción. Se comprueba que un usuario diferente logado con el rol de manager no tenga autorización para mostrar los detalles de una historia de usuario de un proyecto asociado a otro manager. De la misma forma, se comprueba que un usuario logado con un rol distinto a manager tampoco tenga autorización.
Efectividad. Se detectaron 0 bugs.
- **Caso de prueba 9 (create.hack).** Prueba de la creación de una historia de usuario asociada a un proyecto accediendo con otro rol distinto a manager.
Descripción. Se comprueba que un usuario logado con un rol distinto de manager no tenga autorización para crear una historia de usuario asociada a un proyecto.
Efectividad. Se detectaron 0 bugs.
- **Caso de prueba 10 (update.hack).** Prueba de la actualización de una historia de usuario asociada a un proyecto accediendo con otro usuario del mismo rol y con otro rol distinto a manager.
Descripción. Se comprueba que un usuario diferente logado con el rol de manager no tenga autorización para actualizar una historia de usuario de un proyecto asociado a otro manager. De la misma forma, se comprueba que un usuario logado con un rol distinto a manager tampoco tenga autorización. Así mismo, se comprueba que el propio usuario no pueda actualizar una historia de usuario de un proyecto propio que ya haya sido publicada.
Efectividad. Se detectó 1 bug relativo a la autorización de la actualización de una historia de usuario que ya había sido publicada.
- **Caso de prueba 11 (delete.hack).** Prueba de la eliminación de una historia de usuario asociada a un proyecto accediendo con otro usuario del mismo rol y con otro rol distinto a manager.

Descripción. Se comprueba que un usuario diferente logado con el rol de manager no tenga autorización para eliminar una historia de usuario de un proyecto asociado a otro manager, tanto si ha sido publicada como si no. De la misma forma, se comprueba que un usuario logado con un rol distinto a manager tampoco tenga autorización. Así mismo, se comprueba que el propio usuario no pueda eliminar una historia de usuario de un proyecto propio que ya haya sido publicado.

Efectividad. Se detectaron 0 bugs.

- **Caso de prueba 12 (publish.hack).** Prueba de la publicación de una historia de usuario asociada a un proyecto accediendo con otro usuario del mismo rol y con otro rol distinto a manager.

Descripción. Se comprueba que un usuario diferente logado con el rol de manager no tenga autorización para publicar una historia de usuario de un proyecto asociado a otro manager. De la misma forma, se comprueba que un usuario logado con un rol distinto a manager tampoco tenga autorización. Así mismo, se comprueba que el propio usuario no pueda publicar una historia de usuario de un proyecto propio que ya haya sido publicada.

Efectividad. Se detectaron 0 bugs.

4.2. Pruebas de rendimiento

En este capítulo, se presentan los resultados de las pruebas de rendimiento realizadas en el proyecto. Se ha evaluado el tiempo de respuesta del sistema al atender las solicitudes generadas durante las pruebas funcionales en dos ordenadores diferentes, de ahora en adelante con referencia “Ordenador propio” y “Otro ordenador”. Además, se incluyen análisis estadísticos con intervalos de confianza y un contraste de hipótesis para determinar cuál de los dos ordenadores es más potente.

4.2.1. Metodología

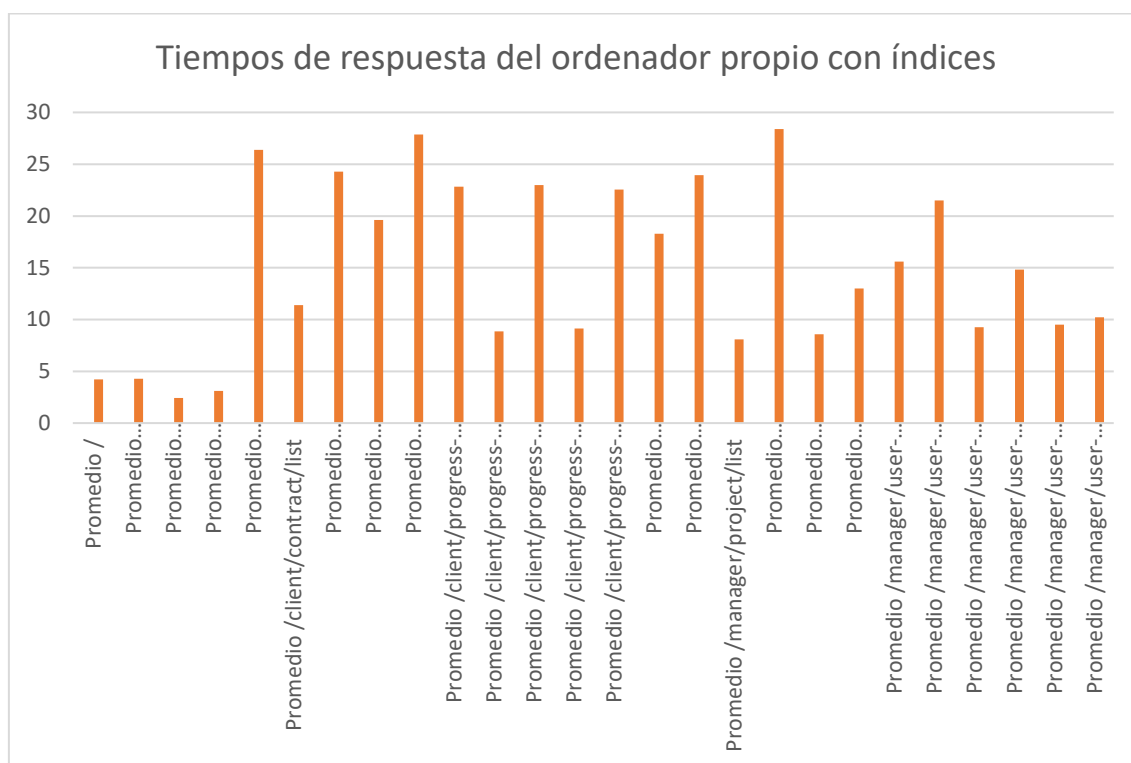
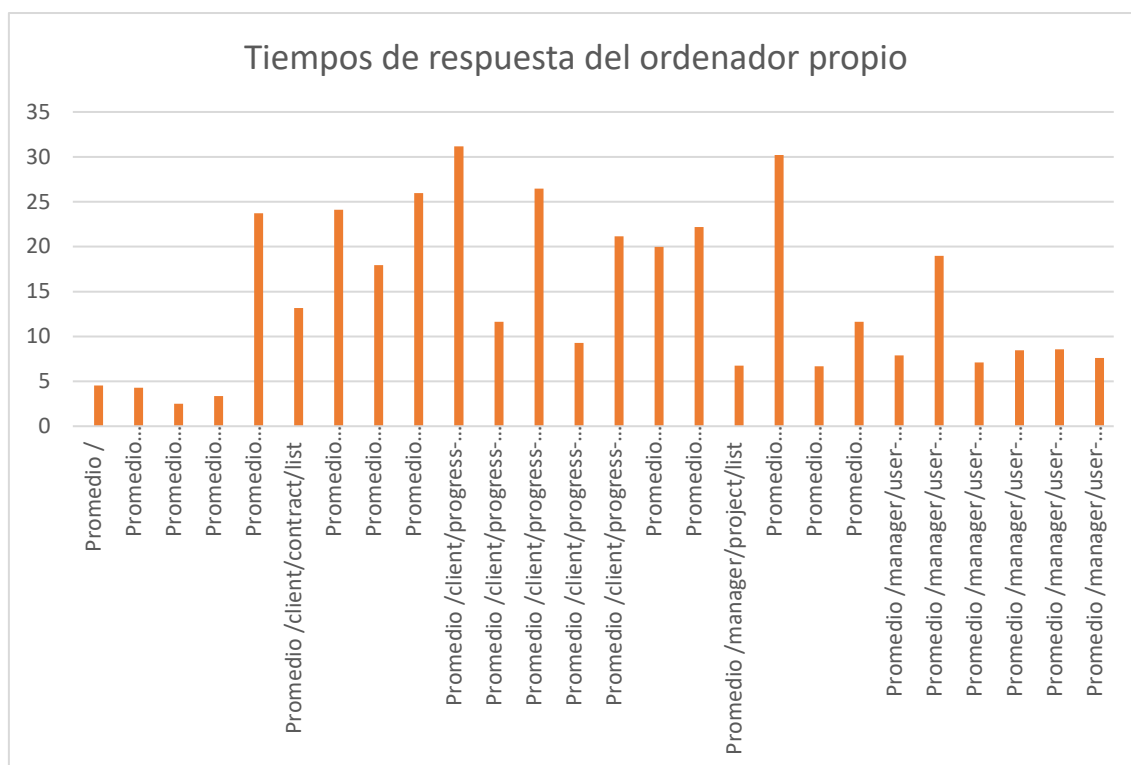
Para las pruebas de rendimiento, se han realizado las siguientes acciones:

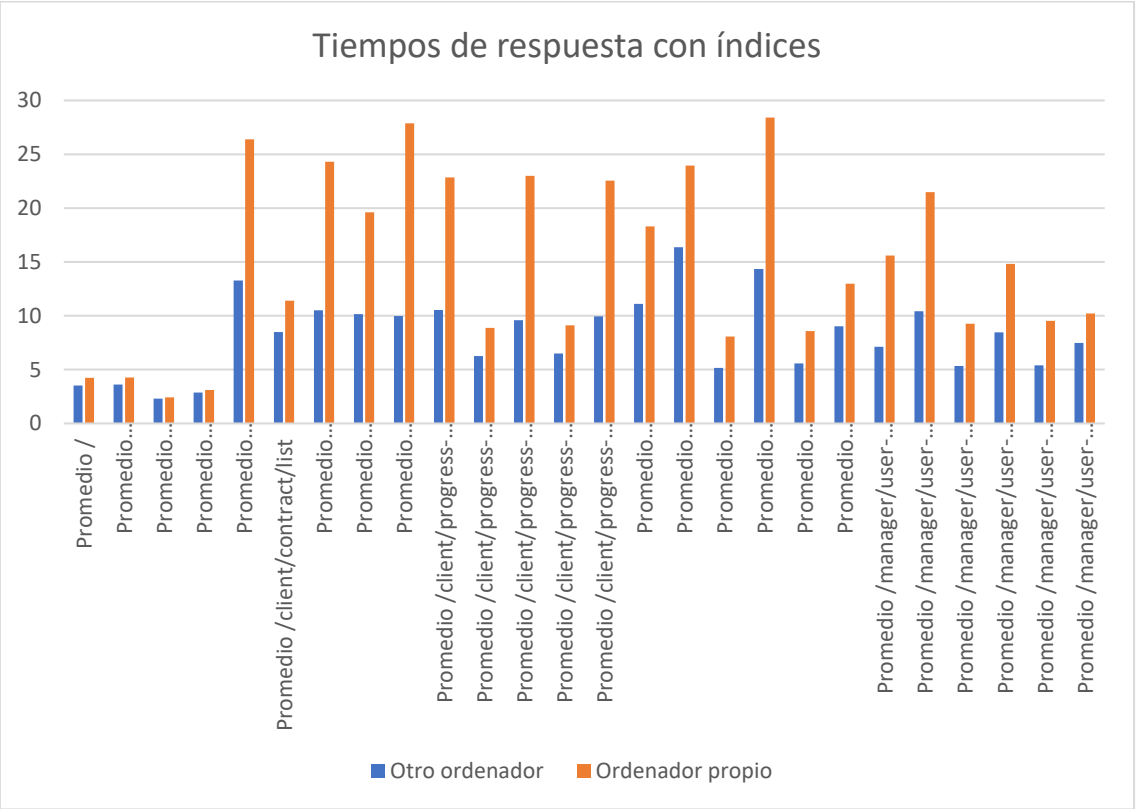
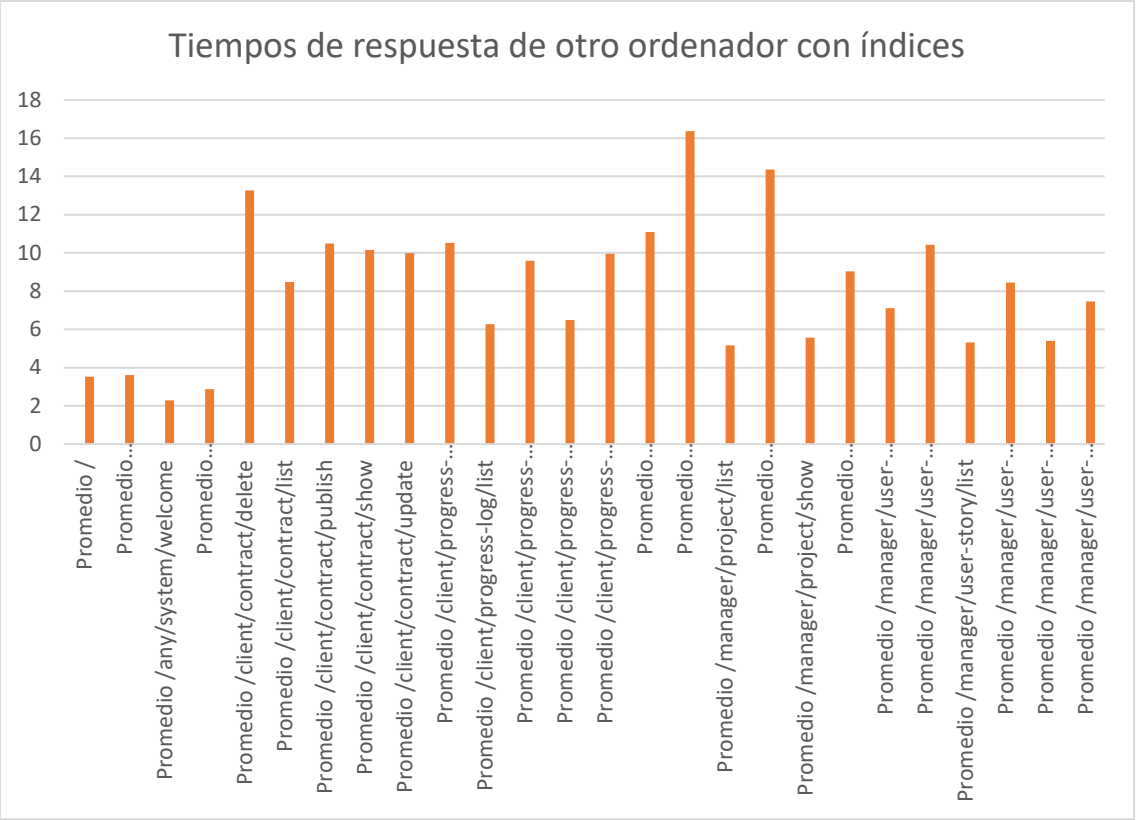
- Entorno de pruebas. El ordenador propio es un ordenador portátil con un procesador Intel i7 10750H y 32 GB de RAM. Por otro lado, el otro ordenador es un ordenador portátil con un procesador Intel i7 1165G7 y 16 GB de RAM.
- Tiempos de respuesta. Se midieron los tiempos de respuesta (wall time) en milisegundos para cada solicitud enviada durante las pruebas funcionales.
- Número de solicitudes. Se realizaron un total de 883 solicitudes en cada ordenador para garantizar la representatividad de los datos para el caso de los tiempos con índices en las consultas.

4.2.2. Resultados de las pruebas de rendimiento

4.2.2.1. Gráficas de tiempos de respuesta

Las siguientes gráficas muestran los tiempos de respuesta registrados en ambos ordenadores:





4.2.2.2. Análisis del intervalo de confianza

Se ha calculado el intervalo de confianza del 95% para los tiempos de respuesta en ambos ordenadores:

Ordenador propio sin índices.

- Tiempo de respuesta promedio: 9.476 ms
- Intervalo de confianza del 95%: [8.785 – 10.167] ms

Ordenador propio con índices.

- Tiempo de respuesta promedio: 11.318 ms
- Intervalo de confianza del 95%: [10.663 – 11.974] ms

Otro ordenador con índices.

- Tiempo de respuesta promedio: 11.318 ms
- Intervalo de confianza del 95%: [10.663 – 11.974] ms

4.2.2.3. Contraste de hipótesis

Para determinar cuál de los dos ordenadores es más potente, se realizó un Z-test con un nivel de confianza del 95%.

Hipótesis.

- H0 (Hipótesis nula). No hay diferencia significativa en el rendimiento entre ambos ordenadores.
- H1 (Hipótesis alternativa). El otro ordenador es significativamente más rápido que el propio ordenador.

Resultados del Z-test.

- Se compararon las medias de los tiempos de respuesta de ambos ordenadores.
- El valor p obtenido fue 0.0005, lo cuál es inferior al nivel de significancia del 0.05.

Dado que el valor p es menor a 0.05, rechazamos la hipótesis H0 y aceptamos la hipótesis H1. Por lo tanto, concluimos que el otro ordenador es significativamente más rápido que el ordenador propio en términos de respuesta.

Conclusión.

Las pruebas de rendimiento muestran que el otro ordenador tiene un mejor desempeño en comparación con el ordenador propio, con tiempos de respuesta más rápidos y una menor variabilidad. Estos resultados son consistentes con las especificaciones de hardware de cada ordenador. En base a estos resultados, se recomienda utilizar el otro ordenador para tareas críticas que requieren tiempos de respuesta mínimos.

5. Conclusiones

Las pruebas realizadas revelan importantes hallazgos tanto en términos de funcionalidad como de rendimiento del sistema. En las pruebas funcionales, se detectó un bug que afectaba la actualización de historias de usuario, particularmente en escenarios de vulnerabilidad. Las pruebas de rendimiento mostraron que el otro ordenador es significativamente más rápido que el ordenador propio, con un tiempo de respuesta promedio más bajo y menor variabilidad. Estos resultados respaldan la recomendación de utilizar el otro ordenador para operaciones que demanden alta eficiencia y baja latencia.

6. Bibliografía

Intencionalmente en blanco.