

Grupo: C1.015

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

Testing Report D04

Jorge Muñoz Rodríguez (jormunrod@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..... 8
 - 4.2.1. Metodología 8
 - 4.2.2. Resultados de las Pruebas de Rendimiento 8
- 5. Conclusiones..... 11
- 6. Bibliografía..... 12

1. Resumen ejecutivo

Este informe analiza las pruebas funcionales y de rendimiento realizadas en el proyecto de gestión de contratos y registros de progreso.

Pruebas Funcionales

Se implementaron 20 casos de prueba para verificar la funcionalidad de contratos y registros de progreso, detectando un total de 7 bugs. Las pruebas destacaron problemas en restricciones de presupuesto y vulnerabilidades de acceso.

Pruebas de Rendimiento

Se evaluaron los tiempos de respuesta en dos computadoras:

- Computadora A (AMD Ryzen 7, 16GB RAM): Tiempo promedio de respuesta: 8,62 ns (IC 95%: 7,88 - 9,36 ns).
- Computadora B (Intel i5, 16GB RAM): Tiempo promedio de respuesta: 10,16 ns (IC 95%: 9,25 - 11,06 ns).

Un Z-test con un nivel de confianza del 95% mostró que la Computadora A es significativamente más rápida ($p = 0,01$).

Conclusiones

Las pruebas funcionales identificaron áreas críticas de mejora, y las pruebas de rendimiento concluyeron que la Computadora A es más eficiente. Se recomienda priorizar las mejoras en seguridad y restricciones, y utilizar la Computadora A para tareas críticas.

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 contratos y registros de progreso, identificando y documentando bugs relevantes. Las pruebas de rendimiento evaluaron el tiempo de respuesta del sistema en dos computadoras distintas, proporcionando un análisis detallado con intervalos de confianza y un contraste de hipótesis. Los resultados demuestran que la Computadora A presenta un mejor desempeño en comparación con la Computadora B, 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 contratos y registros de progreso. 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: Contratos

- **Caso de Prueba 1 (list-show.safe):** Prueba de listado y detalles de contratos.
Descripción: Se verifica que un cliente puede listar sus contratos y ver sus detalles.
Efectividad: Detectó 0 bugs.
- **Caso de Prueba 2 (create.safe):** Prueba de creación de contrato.
Descripción: Se verifica que un cliente puede crear un nuevo contrato. Se comprueban todas las restricciones de los campos del formulario individualmente.
Efectividad: Detectó 1 bug relacionado con la restricción de presupuesto entre el contrato y el proyecto asociado.
- **Caso de Prueba 3 (update.safe):** Prueba de actualización de contrato.
Descripción: Se verifica que un cliente puede actualizar un contrato propio no publicado. Se comprueban todas las restricciones de los campos del formulario individualmente.
Efectividad: Detectó 0 bugs.
- **Caso de Prueba 4 (publish.safe):** Prueba de publicación de contrato.
Descripción: Se verifica que un cliente puede publicar un contrato propio aún no publicado. Se comprueban todas las restricciones de los campos del formulario individualmente, incluyendo la nueva restricción del presupuesto de los contratos publicados asociados a un mismo proyecto.
Efectividad: Detectó 1 bug relacionado con la restricción del presupuesto de los contratos publicados asociados a un mismo proyecto.
- **Caso de Prueba 5 (delete.safe):** Prueba de eliminación de un contrato.
Descripción: Se verifica que un cliente puede eliminar un contrato propio aún no publicado.
Efectividad: Detectó 0 bugs.
- **Caso de Prueba 6 (list-show.hack):** Prueba de vulnerabilidad en listado y detalles de contrato.
Descripción: Se verifica que un usuario con un rol distinto a cliente no puede listar sus contratos. Se comprueba que un usuario no propietario de un contrato no puede ver los detalles de este.
Efectividad: Detectó 1 bug relacionado con la visualización de detalles de un contrato un cliente distinto al cliente atacante.

- **Caso de Prueba 7 (create.hack):** Prueba de vulnerabilidad en creación de contrato.
Descripción: Se verifica que un usuario con un rol distinto a cliente no pueda acceder al formulario de creación de contratos. También se comprueba que el sistema no permita asociar al nuevo contrato un proyecto no publicado.
Efectividad: Detectó 1 bug relacionado con el desplegable de proyectos publicados.

- **Caso de Prueba 8 (update.hack):** Prueba de vulnerabilidad en actualización de contrato.
Descripción: Se verifica que el sistema no permita asociar al contrato un proyecto no publicado. También se comprueba que no se pueda actualizar un contrato ya publicado, ni un contrato perteneciente a otro cliente.
Efectividad: Detectó 1 bug en la actualización de un contrato ya publicado.

- **Caso de Prueba 9 (publish.hack):** Prueba de vulnerabilidad en publicación de contrato.
Descripción: Se verifica que el sistema no permita asociar al contrato un proyecto no publicado. También se comprueba que no se pueda publicar un contrato ya publicado, ni un contrato perteneciente a otro cliente.
Efectividad: Detectó 0 bugs.

- **Caso de Prueba 10 (delete.hack):** Prueba de vulnerabilidad en eliminación de contrato.
Descripción: Se verifica que no se pueda eliminar un contrato ya publicado. También se comprueba que no se pueda eliminar un contrato perteneciente a otro cliente.
Efectividad: Detectó 1 bug en la eliminación de un contrato ya publicado.

Funcionalidad: Registros de progreso

- **Caso de Prueba 1 (list-show.safe):** Prueba de listado y detalles de registros de progreso.
Descripción: Se verifica que un cliente pueda listar y ver los detalles de los registros de progreso de un contrato ya publicado.
Efectividad: Detectó 0 bugs.

- **Caso de Prueba 2 (create.safe):** Prueba de creación de registro de progreso.
Descripción: Se verifica que un cliente puede crear un nuevo registro de progreso en un contrato publicado. Se comprueban todas las restricciones de los campos del formulario individualmente.
Efectividad: Detectó 0 bugs.

- **Caso de Prueba 3 (update.safe):** Prueba de actualización de registro de progreso.
Descripción: Se verifica que un cliente puede actualizar un registro de progreso no publicado perteneciente a un contrato propio publicado. Se comprueban todas las restricciones de los campos del formulario individualmente.
Efectividad: Detectó 0 bugs.

- **Caso de Prueba 4 (publish.safe):** Prueba de publicación de registro de progreso.
Descripción: Se verifica que un cliente puede publicar un registro de progreso aún no publicado perteneciente a un contrato propio publicado. Se comprueban todas las restricciones de los campos del formulario individualmente.
Efectividad: Detectó 0 bugs.
- **Caso de Prueba 5 (delete.safe):** Prueba de eliminación de un registro de progreso.
Descripción: Se verifica que un cliente puede eliminar un registro de progreso no publicado perteneciente a un contrato propio publicado.
Efectividad: Detectó 0 bugs.
- **Caso de Prueba 6 (list-show.hack):** Prueba de vulnerabilidad en listado y detalles de registro de progreso.
Descripción: Se verifica que un usuario con un rol distinto a cliente no puede listar los registros de progreso de un contrato. Se comprueba que un usuario no propietario de un contrato no puede ver los registros de progreso de este.
Efectividad: Detectó 0 bugs.
- **Caso de Prueba 7 (create.hack):** Prueba de vulnerabilidad en creación de registro de progreso.
Descripción: Se verifica que un usuario con un rol distinto a cliente no pueda acceder al formulario de creación de registro de progreso. También se comprueba que el sistema no permita asociar un registro de progreso a un contrato no publicado.
Efectividad: Detectó 1 bug relacionado con la publicación de registros de progreso en contratos no publicados.
- **Caso de Prueba 8 (update.hack):** Prueba de vulnerabilidad en actualización de registro de progreso.
Descripción: Se verifica que el sistema no permita actualizar un registro de progreso ya publicado. También se comprueba que no se pueda actualizar un registro de progreso perteneciente a un contrato de otro cliente.
Efectividad: Detectó 0 bugs.
- **Caso de Prueba 9 (publish.hack):** Prueba de vulnerabilidad en publicación de un registro de progreso.
Descripción: Se verifica que el sistema no permita publicar un registro de progreso ya publicado. También se comprueba que no se pueda publicar un registro de progreso perteneciente a un contrato de otro cliente.
Efectividad: Detectó 0 bugs.
- **Caso de Prueba 10 (delete.hack):** Prueba de vulnerabilidad en eliminación de registro de progreso.
Descripción: Se verifica que no se pueda eliminar un registro de progreso ya publicado. También se comprueba que no se pueda eliminar un registro de progreso perteneciente a un contrato de otro cliente.
Efectividad: Detectó 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 computadoras diferentes. Además, se incluyen análisis estadísticos con intervalos de confianza y un contraste de hipótesis para determinar cuál de las dos computadoras es más potente.

4.2.1. Metodología

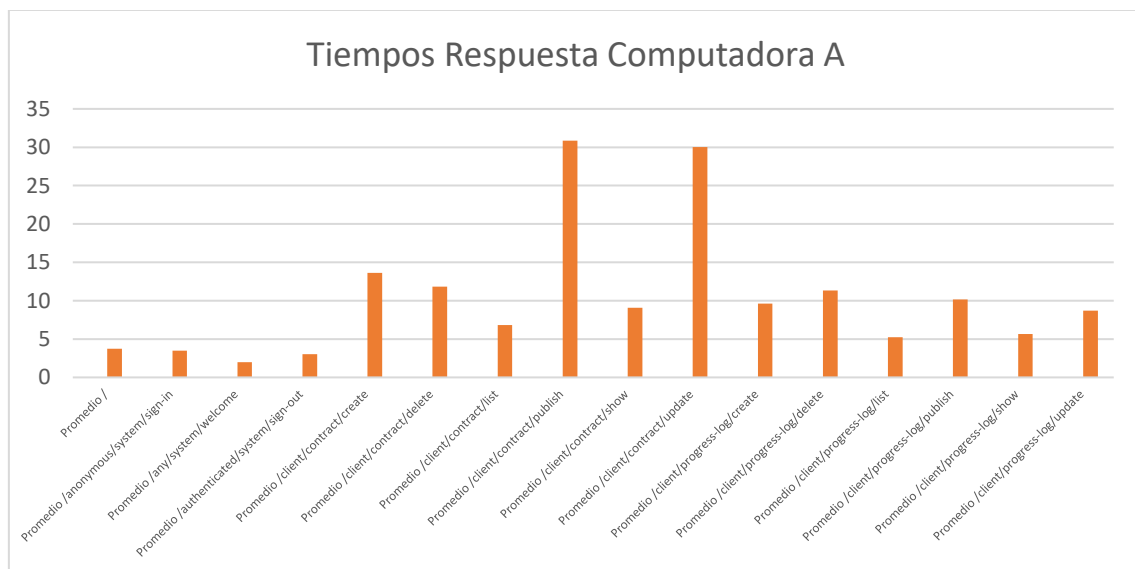
Para las pruebas de rendimiento, se llevaron a cabo las siguientes acciones:

- Entorno de Pruebas: Las pruebas se realizaron en dos computadoras con especificaciones diferentes. La Computadora A es un ordenador portátil con un procesador AMD Ryzen 7 y 16GB de RAM. La Computadora B es un ordenador portátil con un procesador Intel i5 y 16GB 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 560 solicitudes en cada computadora para garantizar la representatividad de los datos.

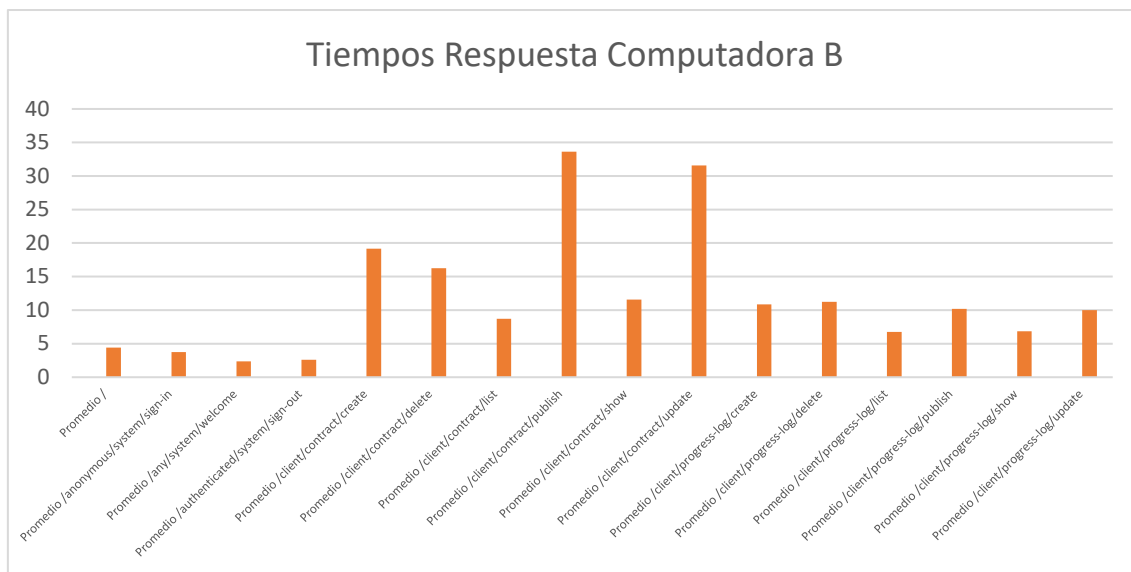
4.2.2. Resultados de las Pruebas de Rendimiento

4.2.2.1. Gráficas de Tiempos de Respuesta

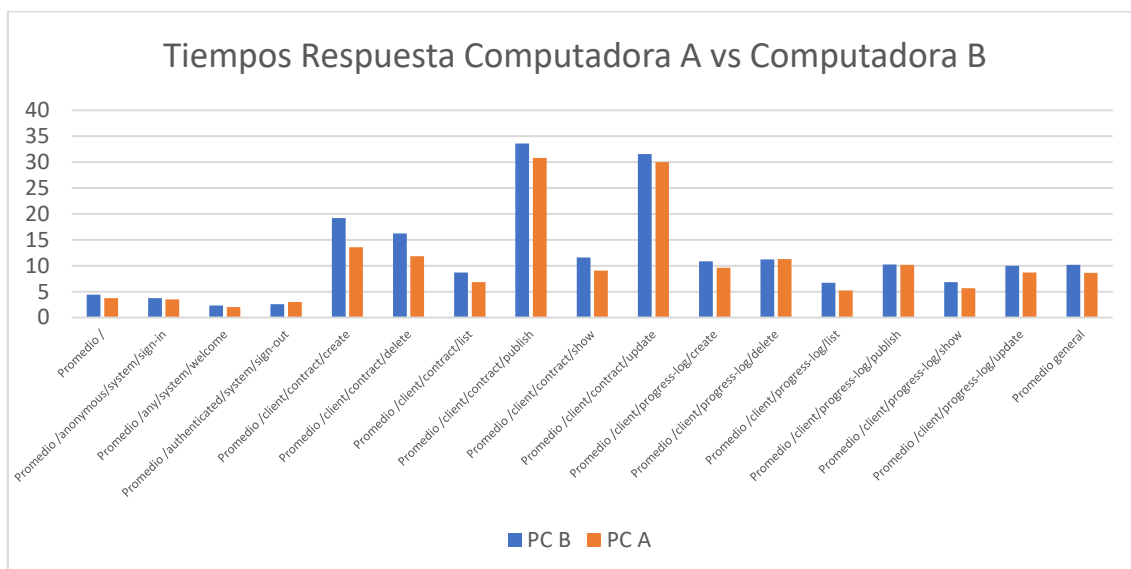
Las siguientes gráficas muestran los tiempos de respuesta registrados en ambas computadoras:



Gráfica 1 - Tiempos de Respuesta en la Computadora A



Gráfica 2 - Tiempos de Respuesta en la Computadora B



Gráfica 3 - Tiempos de Respuesta en la Computadora A vs Computadora B

4.2.2.2. Análisis de Intervalo de Confianza

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

Computadora A:

- Tiempo de respuesta promedio: 8,62 ms
- Intervalo de confianza del 95%: [7,88 - 9,36] ms

Computadora B:

- Tiempo de respuesta promedio: 10,16 ms
- Intervalo de confianza del 95%: [9,25 - 11,06] ms

4.2.2.3. Contraste de Hipótesis

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

Hipótesis:

- H_0 (Hipótesis nula): No hay diferencia significativa en el rendimiento entre la Computadora A y la Computadora B.
- H_1 (Hipótesis alternativa): La Computadora A es significativamente más rápida que la Computadora B.

Resultados del Z-test:

- Se compararon las medias de los tiempos de respuesta de ambas computadoras.
- El valor p obtenido fue 0'01, lo cuál es menor al nivel de significancia 0'05.

Dado que el valor p es menor a 0.05, rechazamos la hipótesis nula H_0 y aceptamos la hipótesis alternativa H_1 . Por lo tanto, concluimos que la Computadora A es significativamente más rápida que la Computadora B en términos de tiempo de respuesta.

Conclusión:

Las pruebas de rendimiento muestran que la Computadora A tiene un mejor desempeño en comparación con la Computadora B, con tiempos de respuesta más rápidos y una menor variabilidad. Estos resultados son consistentes con las especificaciones de hardware de cada computadora. En base a estos hallazgos, se recomienda utilizar la Computadora A para tareas críticas que requieran 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 detectaron varios bugs que afectan tanto la creación como la actualización y eliminación de contratos y registros de progreso, particularmente en escenarios de vulnerabilidad. Las pruebas de rendimiento mostraron que la Computadora A es significativamente más rápida que la Computadora B, con un tiempo de respuesta promedio más bajo y menor variabilidad. Estos resultados respaldan la recomendación de utilizar la Computadora A para operaciones que demanden alta eficiencia y baja latencia.

6. Bibliografía

Intencionalmente blanco.