

Reporte de Testing Grupal



Grupo: C2.039

Repositorio: <https://github.com/pabalcber/C2.039-Acme-SF>

Integrantes:

Nombre	Apellidos	Correo Corporativo
Sheng	Chen	sheche1@alum.us.es
Jun	Yao	junyao@alum.us.es

Tabla de versiones:

Fecha	Versión	Descripción de los cambios	Sprint
24/05/2024	1.0	Creación del documento y redacción de sus contenidos	4
07/07/2024	2.0	Terminar el documento	4

1.Índice

1.Índice	3
2. Resumen Ejecutivo.....	4
3.Tabla de revisiones	5
4. Introducción	6
5.Pruebas Funcionales	7
5.1. Banners	7
7.Conclusión.....	9
8.Bibliografía.....	9

2. Resumen Ejecutivo

Este informe de pruebas resume los resultados de las pruebas realizadas en el proyecto del grupo. En el apartado de pruebas funcionales, se presentan los casos de prueba implementados, agrupados por funcionalidad, con una evaluación de su efectividad en la detección de errores. Por otro lado, el apartado de pruebas de rendimiento proporciona gráficos y un intervalo de confianza del 95% para el tiempo de ejecución en diferentes computadoras, junto con un contraste de hipótesis sobre su potencia relativa. La metodología utilizada fue rigurosa, apoyada en herramientas como Eclipse y una planificación sistemática. Este informe busca ofrecer una visión clara de la calidad y el rendimiento de nuestro software.

3.Tabla de revisiones

Número de revisión	Fecha	Descripción
1	07/07/2024	Revisión final antes de la entrega

4. Introducción

En el ámbito del desarrollo de software, las pruebas son cruciales para garantizar la calidad y el rendimiento del producto final. Este informe presenta los resultados de las pruebas funcionales y de rendimiento, llevadas a cabo en el contexto del grupo de nuestro proyecto. En el apartado de pruebas funcionales, se detallan los casos de prueba implementados y su capacidad para detectar errores en diversas funcionalidades del software. Por otro lado, en el apartado de pruebas de rendimiento, se analiza el tiempo de ejecución en diferentes entornos, proporcionando insights sobre la eficiencia y la capacidad del sistema. Este informe busca proporcionar una evaluación objetiva y completa de la calidad del software desarrollado.

5.Pruebas Funcionales

En este documento se va a hablar sobre los casos de prueba implementados y el rendimiento de los mismos del banners .Los tests implementados son los siguientes:

5.1. Banners

Caso de Prueba 1: List.safe

- **Descripción:** Se prueba la capacidad del sistema para listar los registros de progreso asociados a un banner específico.

Caso de Prueba 2: List.hack

- **Descripción:** Se prueba si el sistema permite publicar contratos de otros usuarios. Este caso de prueba es crucial para garantizar que el sistema no es vulnerable a usuarios con intenciones maliciosas.

Caso de Prueba 3: Show.safe

- **Descripción:** Se prueba la capacidad del sistema para mostrar los detalles de un contrato específico.

Caso de Prueba 4: Show.hack

- **Descripción:** Se prueba si el sistema permite publicar contratos de otros usuarios. Este caso de prueba es crucial para garantizar que el sistema no es vulnerable a usuarios con intenciones maliciosas.

Caso de Prueba 5: Create.safe

- **Descripción:** Se prueba la capacidad del sistema para crear un registro de progreso válido para un contrato específico. El caso de prueba es efectivo en la detección de errores, ya que verifica si el sistema puede crear correctamente un registro de banner con datos válidos proporcionados por el usuario.

Caso de Prueba 6:Create.hack

- **Descripción:** Se prueba si el sistema permite publicar contratos de otros usuarios. Este caso de prueba es crucial para garantizar que el sistema no es vulnerable a usuarios con intenciones maliciosas.

Caso de Prueba 7:Update.safe

- **Descripción:** Se prueba el escenario en el que el usuario selecciona un contrato existente, realiza cambios válidos en los campos y confirma la actualización. Este caso de prueba es efectivo para detectar posibles errores en la lógica de actualización de banners y asegurar que los cambios se reflejen correctamente en el sistema.

Caso de Prueba 8:Update.hack

- **Descripción:** Se prueba si el sistema permite publicar contratos de otros usuarios. Este caso de prueba es crucial para garantizar que el sistema no es vulnerable a usuarios con intenciones maliciosas.

Caso de Prueba 9:Delete.safe

- **Descripción:** Se prueba la capacidad del sistema para eliminar un contrato existente. El caso de prueba es efectivo en la detección de errores, ya que verifica si el sistema puede eliminar correctamente un banner existente sin generar errores o inconsistencias en la base de datos..

Caso de Prueba 10:Delete.hack

- **Descripción:** Se prueba si el sistema permite publicar contratos de otros usuarios. Este caso de prueba es crucial para garantizar que el sistema no es vulnerable a usuarios con intenciones maliciosas.

7.Conclusión

Se han realizado todas las tareas, por lo que consideramos una entrega satisfactoria.

8.Bibliografía

En blanco intencionalmente.