**24/05/2024**

**Reporte de Testing**

Logotipo

Descripción generada automáticamente

**Grupo:** C1.39

**Repositorio:** <https://github.com/pabalcber/C1.039-Acme-SF>

**Integrantes:**

|  |  |  |
| --- | --- | --- |
| Nombre | Apellidos | Correo Corporativo |
| 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 |
| 26/05/2024 | 2.0 | Terminar el documento | 4 |
|  |  |  |  |
|  |  |  |  |

# **1.Índice**

[**1.Índice** 3](#_Toc167734250)

[**2. Resumen Ejecutivo** 4](#_Toc167734251)

[**3.Tabla de revisiones** 5](#_Toc167734252)

[**4. Introducción** 6](#_Toc167734253)

[**5.Pruebas Funcionales** 7](#_Toc167734254)

[5.1. Sponsorship 7](#_Toc167734255)

[5.4 Invoice 9](#_Toc167734256)

[**6.Pruebas de Rendimiento** 12](#_Toc167734257)

[6.1. Gráficos e Intervalos de Confianza 12](#_Toc167734258)

[6.2. Contraste de Hipótesis 15](#_Toc167734259)

[**7.Conclusión** 18](#_Toc167734260)

[**8.Bibliografía** 19](#_Toc167734261)

# **2. Resumen Ejecutivo**

Este informe de pruebas resume los resultados de las pruebas realizadas en el proyecto del estudiante 4. 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 | 27/05/2024 | 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 estudiante 4 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 student 4.Los tests implementados son los siguientes:

## 5.1. Sponsorship

**Caso de Prueba 1: List.safe**

* **Descripción:** para hacer esta prueba, me he metido en las listas de sponsorships de los 3 usuarios de mi rol(sponsor1,sponsor2,sponsor3)

**Caso de Prueba 2: List.hack**

* **Descripción:** Para realizar esta prueba, inicié la aplicación y, sin registrarme con ningún usuario, intenté acceder a la siguiente URL: http://localhost:8081/acme-sf-d04/sponsor/sponsorship/list, lo que resultó en un error 500 con el mensaje "el acceso no está autorizado".

**Caso de Prueba 3: Show.safe**

* **Descripción:** para hacer esta prueba, he metido en listas de sponsorships de los usuarios de mi rol (sponsor1y2)y me he metido en todas las sponsorships.

**Caso de Prueba 4: Show.hack**

* **Descripción:** Para realizar esta prueba, inicié la aplicación y, sin registrarme en ningún usuario, intenté acceder a la siguiente URL que pertenece a un sponsorship del sponsor1: http://localhost:8081acme-sf-d04/sponsor/sponsorship/show?id=274, lo que me dio un error 500 con el mensaje "access is not authorised". Luego, me registré como sponsor2 e intenté acceder a ese mismo sponsorship con la misma URL, obteniendo el mismo error.

**Caso de Prueba 5: Create.safe**

* **Descripción:** Para realizar esta prueba, intenté crear un sponsorship con todos los posibles valores negativos, comenzando con todos los valores en nulo y luego evaluando atributo por atributo con sus respectivos casos negativos. Después de probar estos escenarios negativos, creé varias entidades sponsorships con casos positivos, evaluando atributo por atributo. Estos casos positivos incluyeron los valores extremos superiores e inferiores, así como valores intermedios, respetando los rangos de cada atributo.

**Caso de Prueba 6:Create.hack**

* **Descripción:** Para realizar esta prueba, inicié la aplicación y, sin registrarme en ningún usuario, intenté acceder a la URL: http://localhost:8081acme-sf-d04/sponsor/sponsorship/create, lo que me dio un error 500 con el mensaje "access is not authorised"

**Caso de Prueba 7:Update.safe**

* **Descripción:** Para realizar esta prueba, intenté crear un sponsorship utilizando todos los posibles valores negativos, comenzando con todos los valores nulos y luego evaluando atributo por atributo con sus respectivos casos negativos. Después de probar estos escenarios negativos, creé diversas entidades sponsorship con casos positivos, evaluando atributo por atributo. Estos casos positivos incluyeron valores extremos, tanto superiores como inferiores, así como valores intermedios, respetando los rangos de cada atributo.

**Caso de Prueba 8:Update.hack**

* **Descripción:** Para realizar esta prueba, inicié la aplicación y me registré con el usuario sponsor1. Después de iniciar sesión, accedí a un sponsorship y, utilizando la herramienta de inspección, cambié el valor del ID a 307 (un sponsorship perteneciente al sponsor2) y pulsé el botón de actualización. Luego, repetí el proceso con otro sponsorship, cambiando el ID a 275 (un sponsorship publicado por sponsor1) y volviendo a pulsar el botón de actualización. En ambos casos, apareció un error 500 con el mensaje "access is not authorised".

**Caso de Prueba 9 :Publish.safe**

* **Descripción:** Para realizar esta prueba, intenté crear un sponsorship con todos los posibles valores negativos, comenzando con todos los valores en nulo y luego evaluando atributo por atributo con sus respectivos casos negativos. Después de probar estos escenarios negativos, creé diversas entidades sponsorship con casos positivos, evaluando atributo por atributo. Estos casos positivos incluyeron valores extremos, tanto superiores como inferiores, así como valores intermedios, respetando los rangos de cada atributo.

**Caso de Prueba 10 Publish.hack**

* **Descripción:** Para realizar esta prueba, inicié la aplicación y me registré con el usuario sponsor1. Tras iniciar sesión, accedí a un sponsorship y, utilizando la herramienta de inspección, cambié el valor del ID a 307 (un sponsorship perteneciente al sponsor2) y pulsé el botón de publicación. Luego, repetí el proceso con otro sponsorship, cambiando el ID a 275 (un sponsorship publicado por sponsor1) y volví a pulsar el botón de publicación. En ambos casos, apareció un error 500 con el mensaje "access is not authorised".

**Caso de Prueba 11:Delete.safe**

* **Descripción:** Para realizar esta prueba, intenté borrar un sponsorship con entidades secundarias publicadas (caso negativo) y también intenté borrar un sponsorship sin entidades secundarias (caso positivo).

**Caso de Prueba 12:Delete.hack**

* **Descripción:** Para realizar esta prueba, inicié la aplicación y me registré con el usuario sponsor1. Tras iniciar sesión, accedí a un sponsorship y, utilizando la herramienta de inspección, cambié el valor del ID a 307 (un sponsorship perteneciente al sponsor2) y pulsé el botón de eliminar. Luego, repetí el proceso con otro sponsorship, cambiando el ID a 275 (un sponsorship publicado por sponsor1) y volví a pulsar el botón de eliminar. En ambos casos, apareció un error 500 con el mensaje "access is not authorised".

## 5.2 Invoice

**Caso de Prueba 1: List.safe**

* **Descripción:** Para llevar a cabo esta prueba, accedí a la lista de sponsorships de los usuarios de mis roles (sponsor1 y 2) y enumeré los invoices de cada uno.

**Caso de Prueba 2: List.hack**

* **Descripción:** Para realizar esta prueba, inicié la aplicación sin registrarme en ningún usuario. Intenté acceder a la siguiente URL, que corresponde a los invoices de un sponsorship del sponsor1: http://localhost:8081acme-sf-d04/sponsor/invoice/list?sponsorshipId=274, lo que resultó en un error 500 con el mensaje "access is not authorised". Luego, me registré como sponsor2 e intenté acceder al mismo invoice utilizando la misma URL, obteniendo el mismo error.

**Caso de Prueba 4 : Show.safe**

* **Descripción:** Para llevar a cabo esta prueba, accedí a las listas de sponsorships de los usuarios de mis roles (sponsor 1 y 2), donde listé las invoices de cada sponsorship. Luego, he metido una de las invoices existentes.

**Caso de Prueba 5:Show.hack**

* **Descripción:** S Para realizar esta prueba, inicié la aplicación sin registrarme en ningún usuario. Intenté acceder a la siguiente URL, que pertenece a un invoice del sponsor1: http://localhost:8081/acme-sfd04/sponsor/invoice/show?id=317, lo que resultó en un error 500 con el mensaje "access is not authorised". Luego, me registré como sponsor2 e intenté acceder al mismo invoice utilizando la misma URL, obteniendo el mismo error.

**Caso de Prueba 6:Create.safe**

* **Descripción:** para hacer esta prueba, he probado a crear una invoice con todos los posibles valores negativos empezando por todos los valores a nulos y a continuación yendo atributo por atributo con sus respectivos casos. Tras probar los escenarios negativos creé diversas entidades invoice con los casos positivos yendo atributo por atributo, estos casos positivos son los extremos tanto superiores como inferiores y también valores intermedios; respetando los rangos de cada atributo.

**Caso de Prueba 7: Create.hack**

* **Descripción:** Para realizar esta prueba, inicié la aplicación sin registrarme en ningún usuario. Intenté acceder a la siguiente URL, que pertenece a una invoice del sponsor1: http://localhost:8081/acme-sf-d04/sponsor/invoice/create?sponsorshipId=274, lo que resultó en un error 500 con el mensaje "access is not authorised". Luego, me registré como sponsor2 e intenté acceder a esa invoice utilizando la misma URL, obteniendo el mismo error.

**Caso de Prueba 8:update.safe**

* **Descripción:** Para realizar esta prueba, intenté actualizar una invoice con todos los posibles valores negativos, comenzando con todos los valores nulos y luego evaluando atributo por atributo con sus respectivos casos negativos. Después de probar estos escenarios negativos, actualicé diversas entidades invoice con casos positivos, evaluando atributo por atributo. Estos casos positivos incluyeron valores extremos, tanto superiores como inferiores, así como valores intermedios, respetando los rangos de cada atributo..

**Caso de Prueba 9: update.hack**

* **Descripción:** Para llevar a cabo esta prueba, inicié la aplicación y me registré con el usuario sponsor1. Después de iniciar sesión, accedí a una invoice y, utilizando la herramienta de inspección del navegador, cambié el valor del ID a 343 (una invoice perteneciente al sponsor2) y pulsé el botón de actualización. Luego, repetí el proceso con el mismo invoice y cambié el ID a 342 (un audit record publicado por sponsor1) y volví a pulsar en el botón de actualización. En ambos casos, apareció un error 500 con el mensaje "access is not authorised".

**Caso de Prueba 10: Publish.safe**

* **Descripción:** Para realizar esta prueba, intenté publicar una invoice con todos los posibles valores negativos, comenzando con todos los valores nulos y luego evaluando atributo por atributo con sus respectivos casos negativos. Después de probar estos escenarios negativos, publiqué diversas entidades invoice con casos positivos, evaluando atributo por atributo. Estos casos positivos incluyeron valores extremos, tanto superiores como inferiores, así como valores intermedios, respetando los rangos de cada atributo.

**Caso de Prueba 11: Publish.hack**

* **Descripción:** Para realizar esta prueba, inicié la aplicación y me registré con el usuario sponsor1. Después de iniciar sesión, accedí a una invoice y, utilizando la herramienta de inspección en el navegador, cambié el valor del ID a 343 (una invoice perteneciente al sponsor2) y pulsé el botón de publicación. Luego, repetí el proceso con el mismo invoice y cambié el ID a 342 (un audit record publicado por sponsor1) y volví a pulsar en el botón de publicación. En ambos casos, apareció un error 500 con el mensaje "access is not authorised".

**Caso de Prueba 12: Delete.safe**

* **Descripción:** Para realizar esta prueba, intenté borrar una invoice en dos sponsorships distintas, ya que no existe un caso negativo legal.

**Caso de Prueba 13: Delete.hack**

* **Descripción:** Para llevar a cabo esta prueba, inicié la aplicación y me registré con el usuario sponsor1. Después de iniciar sesión, accedí a una invoice y, utilizando la herramienta de inspección en el navegador, cambié el valor del ID a 343 (una invoice perteneciente al sponsor2) y pulsé el botón de eliminar. Luego, repetí el proceso con el mismo invoice y cambié el ID a 342 (un audit record publicado por sponsor1) y volví a pulsar en el botón de eliminar. En ambos casos, apareció un error 500 con el mensaje "access is not authorised"..

# **6.Pruebas de Rendimiento**

## 6.1. Gráficos e Intervalos de Confianza

**Ordenador 1**

Imagen que contiene texto, juego, periódico, gente

Descripción generada automáticamente

Gráfico, Gráfico de barras

Descripción generada automáticamente

**Ordenador 2**

Aplicación, Tabla

Descripción generada automáticamente

**Gráfico

Descripción generada automáticamente**

Se puede ver que el PC\_B tiene peor rendimiento que el PC\_A.

## 6.2. Contraste de Hipótesis

**Ordenador 1:**

Interfaz de usuario gráfica, Aplicación, Tabla, Excel

Descripción generada automáticamente

**Ordenador 2**

Interfaz de usuario gráfica, Tabla, Excel

Descripción generada automáticamente

Interfaz de usuario gráfica, Aplicación, Tabla, Excel

Descripción generada automáticamente

Como podemos observar, el valor P (valor crítico de z (dos colas)) es menor que 0.05, que es nuestro porcentaje de confianza. Por esta razón sabemos que comparar las medias de los tiempos es una buena manera de averiguar qué ordenador es más potente. En este caso PC\_A es mejor al tener una media de tiempos menor

Gráfico, Gráfico de barras, Gráfico en cascada

Descripción generada automáticamente

Gráfico, Gráfico en cascada

Descripción generada automáticamente

Se puede ver que sin índices tiene mejor rendimiento que con índices, pero prácticamente, se compensan unos con otros.

Interfaz de usuario gráfica, Aplicación, Tabla, Excel

Descripción generada automáticamente

Como podemos observar, el valor P (valor crítico de z (dos colas)) es menor que 0.05, que es nuestro porcentaje de confianza. Por esta razón sabemos que comparar las medias con y sin índices es una buena manera de averiguar qué ordenador es más potente. En este caso, antes de los índices es mejor al tener una media de tiempos menor.

# **7.Conclusión**

Las pruebas funcionales realizadas en la rama feature/S4-Man09 han demostrado que las funcionalidades clave, como la creación, actualización, publicación, eliminación y visualización de contratos y registros de progreso, operan correctamente. Los casos de prueba abordan varios escenarios, incluyendo datos inválidos, campos obligatorios vacíos, límites de entrada y posibles intentos de hacking. La efectividad en la detección de errores ha sido alta, asegurando la robustez del sistema ante diferentes tipos de entradas y validaciones. Dado que algunas modificaciones recientes en el archivo CSV pueden afectar la ejecución de estos tests, se recomienda realizar una revisión y ejecución en la rama feature/S4-Man09 para garantizar su precisión y confiabilidad.

# **8.Bibliografía**

En blanco intencionalmente.