Universidad de Sevilla

Escuela Técnica Superior de Ingeniería Informática

**Documentación de la entrega D04**

**Testing**



Grado en Ingeniería Informática – Ingeniería del Software  
 Diseño y Pruebas 2

Curso 2023 – 2024

|  |  |
| --- | --- |
| **Fecha** | **Versión** |
| 03/03/2024 | v1r1 |

|  |  |  |
| --- | --- | --- |
| **Grupo de prácticas: G6-64** | | |
| **Autores por orden alfabético** | **Rol** | **Correo electrónico** |
| Aguayo Orozco, Sergio - 25604244T | Desarrollador | ahydul1@gmail.com |
| García Lama, Gonzalo - 47267072W | Desarrollador, Tester | gongarlam@alum.us.es |
| Huecas Calderón, Tomás - 17476993Y | Desarrollador | tomhuecal@alum.us.es |
| Fernández Pérez, Pablo - 54370557Y | Desarrollador,  Analista | pablofp.33@gmail.com |
| Youssafi Benichikh, Karim - 28823709V | Desarrollador, operador, manager | karyouben@alum.us.es |

## Repositorio: https://github.com/karyouben/Acme-SF-D04

**Índice de contenido**

[1. Resumen ejecutivo 3](#_Toc158967746)

[2. Control de versiones 4](#_Toc158967747)

[3. Introducción 5](#_Toc158967748)

[4. Contenido 6](#_Toc158967749)

[5. Conclusiones 6](#_Toc158967750)

[6. Bibliografía 17](#_Toc158967751)

# 1. Resumen ejecutivo

El documento de pruebas está dividido en dos apartados. El primero está centrado en pruebas funcionales y el segundo está centrado en pruebas de rendimiento.

# 2. Control de versiones

|  |  |  |
| --- | --- | --- |
| **Fecha** | **Versión** | **Descripción** |
| 20/05/2024 | v1r0 | Creada documento |
|  |  |  |
|  |  |  |
|  |  |  |

# 3. Introducción

A continuación, mostramos los resultados de las pruebas. El primer apartado incluye un listado con los casos de prueba implementados, agrupados por implementación, con descripción sobre su efectividad a la hora de encontrar errores.

El segundo apartado incluye información sobre el rendimiento de la aplicación. Gráficas e intervalos de confianza 95% tomados en dos ordenadores distintos, además de un contraste de hipótesis de confianza respecto a qué ordenador es más potente.

# 4. Contenido

**Pruebas Funcionales**

**Proyect**

|  |  |
| --- | --- |
| Create.safe  Create2.safe | Se inicia sesión como manager1 y se prueba a crear proyectos de todas las formas que permite el sistema, comprobando los límites y los mensajes de error. No se detecta ningún error. |
| Delete.safe | Se inicia sesión como manager2 y se crea un proyecto (porque no estaría publicado). Se elimina el proyecto. No se detecta ningún error. |
| List.safe | Se inicia sesión como manager1 y se listan sus proyectos. No se detecta ningún error. |
| Update.safe | Se inicia sesión como manager1 y se prueba a actualizar los datos del proyecto con id 48 de todas las formas que permite el sistema, comprobando los límites y los mensajes de error. No se detecta ningún error. |
| Publish.safe | Se inicia sesión como manager2 y se crea un proyecto. Se prueba a publicar el proyecto cuando no tiene historias de usuario asociada, cuando tiene errores y cuando las historias no están publicadas. Luego se publica el proyecto. No se detecta ningún error. |
| Show.safe | Se inicia sesión como manager1, se listan sus proyectos y se muestra cada proyecto uno a uno. No se detecta ningún error. |
| Delete.hack | Primero se hace get de /manager/project/delete con id 48 (¿id=48), con id 12312312 con usuario anónimo. Luego se inicia sesión como manager2 y se hace lo mismo. Se crea un proyecto y se usa para hackear con la consola para los id 48 (del manager1) y 12312312 (no existe). Finalmente se inicia sesión como manager1 y se hace hacking para id 47 (publicado) y el id 242342. No se detecta ningún problema. |
| List.hack | Se hace /manager/project/list con usuario anónimo. No se detecta ningún problema. |
| Update.hack | Se hace get de /manager/project/update con id 48 (no está publicado) y con id 23423(no existe) con usuario anónimo. Se hace lo mismo como manager2 y se crea un proyecto para hackear con la consola para el id 48 (del manager1) y 23423. Finalmente se inicia sesión como manager1 y se intenta actualizar el proyecto con id 47 (ya publicado). No se detecta ningún problema. |
| Publish.hack | Se hace get de /manager/project/publish con id 48 (no está publicado) y con id 123123 (no existe) con usuario anónimo. Se hace lo mismo como manager2 y se crea un proyecto para hackear con la consola para el id 48 (del manager1) y 452423. Finalmente se inicia sesión como manager1 y se intenta actualizar el proyecto con id 47 (ya publicado) y con un id inexistente. No se detecta ningún problema. |
| Show.hack | Se hace get /manager/project/show con id 48 y 24234 con usuario anónimo. Se inicia sesión como manager2 y se hace lo mismo. No se detecta ningún problema |

**UserStory**

|  |  |
| --- | --- |
| Create.safe | Se inicia sesión como manager1 y se prueba a crear historias de usuario de todas las formas que permite el sistema, comprobando los límites y los mensajes de error. No se detecta ningún error. |
| Delete.safe | Se inicia sesión como manager1 y se eliminan las historias de usuario con ids 127, 135,136,137,138,139. Luego se crea una historia de usuario, se asigna a un proyecto y se intenta eliminar, saltando mensaje de error. Se descubrió que no estaba implementada esta última parte: evitar que una historia de usuario asignada sea eliminada, por lo que saltaba un error. Esto fue solucionado. |
| List.safe | Se inicia sesión como manager1 y se listan las historias de usuario. No se detecta ningún error. |
| List-by-proyect.safe  List-by-proyect2.safe | Se inicia sesión como manager1 y se listan las historias de usuario asociadas a cada proyecto. En el otro archivo se añade una lista de usuario con draftMode = true a un proeycto y se listan las historias de usuario de ese proyecto. No se detecta ningún error. |
| Publish.safe | Se inicia sesión como manager1 y se publica su historia de usuario con id 127. No se detectó ningún problema. |
| Show.safe | Se inicia sesión como manager1, se listan sus historias de usuario y se muestra cada una. No se detecta ningún error. |
| Update.safe | Se inicia sesión como manager1 y se prueba a actualizar los datos de la historia de usuario con id 127 de todas las formas que permite el sistema, comprobando los límites y los mensajes de error. No se detecta ningún error. |
| List.hack | Se hace /manager/user-story/list con usuario anónimo. No se detecta ningún problema. |
| List-by-project.hack | /manager/user-story/list-by-proyect con projectId=30 y 24234 con usuario anónimo. Luego se hace lo mismo iniciado sesión como manager2. No se detecta ningún problema. |
| Publish.hack | /manager/user-story/publish con ids 135 y 135135 tanto con usuario anónimo como siendo manager2. Además, se crea una historia de usuario siendo manager2 para hackear usando la consola pretendiendo publicar los ids 135 (del manager1), 135135 (no existe) y 126 (no publicado del manager2). No se detecta ningún problema. |
| Delete.hack | /manager/user-story/delete con ids 135 y 135135 tanto con usuario anónimo como siendo manager2. Además, se crea una historia de usuario siendo manager2 para hackear usando la consola pretendiendo eliminar los ids 135 (del manager1), 135135 (no existe) y 126 (no publicado del manager2). No se detecta ningún problema. |
| Show.hack | Se hace get /manager/user-story/show con id 90 y 34534 siendo usuario anónimo. Se inicia sesión como manager2 y se hace lo mismo. No se detecta ningún problema |
| Update.hack | Se hace get /manager/user-story/update con ids 120 y 24234 tanto con usuario anónimo como siendo manager2. Se crea una historia de usuario para hackear usando la consola e intentar actualizar los ids 120, 126 y 13123. No se detectó ningún problema. |

**Assignment**

|  |  |
| --- | --- |
| Create.safe | Se inicia sesión como manager1 y se crea un assignment con el proyecto con id 48 y la historia de usuario con id 85. No se detectó ningún problema. |
| Delete.safe | Se inicia sesión como manager1 y se elimina el assignment con id 162. No se detectó ningún problema. |
| List.safe | Se inicia sesión como manager1, se muestra el proyecto con id 48 y se hace click en eliminar historias de usuario para listarlas. Este es el único assignment que se puede visualizar puesto que no se puede eliminar assignments de proyectos publicados, el resto la aplicación no lo permite. No se detectó ningún problema en especial, solamente se decidió hacer un cambio para hacerlo más fácil añadiendo isDraftMode al listado. |
| Show.safe | Se inicia sesión como manager1, se muestra el proyecto con id 48, se hace click en eliminar historias de usuario y se hace click en la primera. No se detectó ningún problema. |
| Delete.hack | Se inicia sesión como manager2, se crea un proyecto, al que se le añade una historia de usuario, creándose un assignment. Se usa ese assignment para hackear en la consola usando id 140, el cual es del manager1, y el id 123123, el cual no existe. Luego se inicia sesión con manager1 y se hace lo mismo con el id 155, el cual es de un proyecto publicado. Se detectó que el proyecto del assignment debe no estar publicado, lo cual no estaba implementado. Esto es tanto en delete como list y show. |
| List.hack | Se hace /manager/assignment/list-by-proyect?proyectId= con projectId 48 y 123123 tanto con usuario anónimo como habiendo iniciado como manager2. Luego como manager1 se hace lo mismo con el id 41, un proyecto publicado. Detectado el error anterior. |
| Show.hack | Se have /manager/assignment/show con id 162 y 123123 tanto con usuario anónimo como habiendo iniciado como manager2. Luego como manager1 se hace lo mismo con el id 155, assignment con proyecto publicado. Detectado el error anterior. |

**Pruebas de Rendimiento**

**PC A**

**A graph on a white sheet

Description automatically generated**

**PC B**

**A graph with orange lines

Description automatically generated**

El PC A por lo general tiene tiempos más bajos salvo por manager/assignment/create, el cual ha sido mayor de lo normal. Si se repitiera seguramente sería más bajo.

# **A screenshot of a spreadsheet Description automatically generated**

# El PCA tiene intervalo de confianza 95% (5.5, 6.7) y el PCB tiene intervalo (6.5, 7.8). Son intervalos más que aceptables.

A continuación calculamos la hipótesis de contraste con 95% confianza para intentar averiguar que ordenador es más potente:

# **A screenshot of a computer Description automatically generated**

Como podemos observar, el valor P es menor que 0.05 (1 menos 0.95, nuestro porcentaje de confianza). Por esta razón sabemos que comparar las medias de los tiempos es una buena manera de averiguar que ordenador es más potente. En este caso el PCA es mejor al tener una media de tiempos menor. Ese ordenador tiene una CPU i7-12700H, la cual es bastante moderna, mientras que el pc B tiene una CPU algo más antigua y bastante menos potente un i5-11600, aunque no sea de portátil, es inferior.

# 5. Conclusiones

Este documento recoge las pruebas realizadas los cuales han servido para encontrar algún error en el código y calcular el rendimiento de nuestro código compara con dos ordenadores.

# 6. Bibliografía

Intencionadamente en blanco