

Group C1.020 | Diseño y Pruebas II | 24/05/2024

Fecha	Versión	Autor
24/05/2024	1.0	Guillermo Alonso Pacheco Rodrigues

Miembros:

- Guillermo Alonso Pacheco Rodrigues (guipacrod@alum.us.es)

Repositorio de Github: https://github.com/DP2-C1-020/Acme-SF-D04

Contenido

Resumen ejecutivo	3
Introducción	4
Contenido	5
Conclusiones	11
Bibliografía	12

Resumen ejecutivo

Este documento proporcionará un análisis exhaustivo del procedimiento de pruebas y sus resultados, con una sección dedicada a las pruebas funcionales y otra a las pruebas de rendimiento. Utilizaremos un enfoque claro y preciso para facilitar la comprensión y garantizar un producto final de alta calidad.

Introducción

El presente documento establece las directrices para la elaboración de un informe de pruebas, organizado en dos capítulos principales: pruebas funcionales y pruebas de rendimiento. El objetivo de este informe es proporcionar un análisis detallado y estructurado de los casos de prueba implementados, así como evaluar el rendimiento del proyecto en distintos entornos.

En el capítulo dedicado a las pruebas funcionales, se presentarán los casos de prueba agrupados por características, acompañados de una descripción concisa y una evaluación de su efectividad para identificar errores. Este enfoque meticuloso asegura que cada aspecto funcional del proyecto sea evaluado de manera sistemática y exhaustiva.

El capítulo de pruebas de rendimiento se centrará en proporcionar gráficos informativos y un intervalo de confianza del 95% para el tiempo de respuesta del sistema al atender solicitudes. Se comparará el desempeño del proyecto en dos computadoras distintas, proporcionando un análisis basado en un contraste de hipótesis con un 95% de confianza sobre cuál es la computadora más potente.

Este informe ha sido diseñado para ser claro y accesible, facilitando la comprensión de los resultados y garantizando que se puedan tomar decisiones informadas para mejorar la calidad final del producto.

Contenido

Pruebas funcionales

Para mantener un orden, comenzaremos con el análisis de las pruebas funcionales correspondientes al requisito número 6: *Operations by Auditor on Code Audits*. Dentro de este requisito, abordaremos cada funcionalidad por separado.

List

Listamos los Code Audits de un auditor.

Para las pruebas de hacking, se intentó acceder a esta URL utilizando diferentes roles incorrectos.

Esta prueba ha proporcionado una cobertura del 94,1%. No se encontraron bugs.

• Show

Se seleccionaron varios Code Audits y se hizo clic en cada uno de ellos para ver sus detalles.

Para las pruebas de hacking, se intentó acceder a un Code Audit del Auditor1 por varios roles, incluyendo el Auditor2, que tiene el rol correcto pero el usuario no es válido. Se demostró satisfactoriamente que ninguno de ellos podía acceder.

Esta prueba ha proporcionado una cobertura del 96,9%. No se encontraron errores.

Delete

Se seleccionaron varios Code Audits no publicados y se eliminaron cada uno de ellos.

Para las pruebas de hacking, el framework a través del navegador web solo admite pruebas de operaciones de hacking GET. Por lo tanto, no se realizaron pruebas de hacking para este caso.

Esta prueba ha proporcionado una cobertura del 83,5%. No se encontraron errores.

Create

Para esta prueba se ha creado un nuevo Code Audit. Pero antes de crearlo se verificó que el sistema rechaza datos incorrectos en todos los campos del formulario. Posteriormente, se probaron varios datos válidos para cada atributo. Para las pruebas de hacking, el framework a través del navegador web solo admite pruebas de operaciones de hacking GET. Por lo tanto, no se realizaron pruebas de hacking para este caso.

Esta prueba ha proporcionado una cobertura del 94,4%.

Durante el proceso, se identificó un fallo relacionado con el límite de caracteres en los atributos "Link", el cual generaba un error al ingresar una cadena de más de 255 caracteres.

Update

Para esta prueba se seleccionó un Code Audit no publicado y se actualizó. Pero antes de actualizarlo se verificó que el sistema rechaza datos incorrectos en todos los campos del formulario. Posteriormente, se probaron varios datos válidos para cada atributo.

Para las pruebas de hacking, el framework a través del navegador web solo admite pruebas de operaciones de hacking GET. Por lo tanto, no se realizaron pruebas de hacking para este caso.

Esta prueba ha proporcionado una cobertura del 91,2%. No se encontraron errores.

Publish

Para esta prueba se seleccionó un Code Audit no publicado y se procedió a publicarlo. Pero antes de publicarlo se verificó que el sistema rechaza datos incorrectos en todos los campos del formulario. Posteriormente, se probaron varios datos válidos para cada atributo.

Para las pruebas de hacking, el framework a través del navegador web solo admite pruebas de operaciones de hacking GET. Por lo tanto, no se realizaron pruebas de hacking para este caso.

Esta prueba ha proporcionado una cobertura del 95,3%.

Durante la ejecución de esta prueba, se identificó un error que no había sido detectado hasta el momento. El error consistía en que, al completar el formulario con datos incorrectos y recibir un mensaje de error, el atributo "Mark" se eliminaba del formulario. Como resultado, al intentar actualizar el formulario con datos correctos, surgía un error que impedía la publicación del Code Audit.

Este conjunto de casos de prueba ha logrado alcanzar una cobertura del 93,4% para el paquete *acme.features.auditor.code_audit*.

A continuación, pasamos con el análisis de las pruebas funcionales para el requisito 7: Operations by Auditors on Audit Records.

List

Listamos los Audit Records correspondientes a varios Code Audits.

Para las pruebas de hacking, se intentó acceder a esta URL utilizando diferentes roles incorrectos. También se probó para un rol correcto pero usuario incorrecto. Esta prueba ha proporcionado una cobertura del 94,7%. No se encontraron bugs.

Show

Se seleccionaron varios Audit Records y se mostraron sus detalles.

Para las pruebas de hacking, se intentó acceder a un Audit Record de Code Audit del Auditor1 por tres roles, Anonymous, Manager y Auditor, usando Auditor2, que tiene el rol correcto pero el usuario no es válido. Se demostró satisfactoriamente que ninguno de ellos podía acceder.

En este caso, al grabar la prueba de hacking para los tres roles simultáneamente, surgieron errores al ejecutar la prueba debido a solicitudes fantasma causadas por la aleatoriedad de los banners. Para solucionar esto, grabé las pruebas de hack en tres archivos diferentes: show1.hack, show2.hack y show3.hack.

Esta prueba ha proporcionado una cobertura del 95,1%. No se encontraron bugs.

Delete

Se seleccionaron varios Audit Records no publicados y se eliminaron cada uno de ellos.

Para las pruebas de hacking, el framework a través del navegador web solo admite pruebas de operaciones de hacking GET. Por lo tanto, no se realizaron pruebas de hacking para este caso.

Esta prueba ha proporcionado una cobertura del 82,9%. No se encontraron errores.

Create

Para esta prueba se ha creado un nuevo Audit Record. Pero antes de crearlo se verificó que el sistema rechaza datos incorrectos en todos los campos del formulario. Posteriormente, se probaron varios datos válidos para cada atributo. Para las pruebas de hacking, el framework a través del navegador web solo admite pruebas de operaciones de hacking GET. Por lo tanto, no se realizaron pruebas de hacking para este caso.

Esta prueba ha proporcionado una cobertura del 94,0%. No se encontraron errores.

Update

Para esta prueba se seleccionó un Audit Record no publicado y se actualizó. Pero antes de actualizarlo se verificó que el sistema rechaza datos incorrectos en todos los campos del formulario. Posteriormente, se probaron varios datos válidos para cada atributo.

Para las pruebas de hacking, el framework a través del navegador web solo admite pruebas de operaciones de hacking GET. Por lo tanto, no se realizaron pruebas de hacking para este caso.

Esta prueba ha proporcionado una cobertura del 93,0%. No se encontraron errores.

Publish

Para esta prueba se seleccionó un Audit Record no publicado y se procedió a publicarlo. Pero antes de publicarlo se verificó que el sistema rechaza datos incorrectos en todos los campos del formulario. Posteriormente, se probaron varios datos válidos para cada atributo.

Para las pruebas de hacking, el framework a través del navegador web solo admite pruebas de operaciones de hacking GET. Por lo tanto, no se realizaron pruebas de hacking para este caso.

Esta prueba ha proporcionado una cobertura del 93,0%. No se encontraron errores.

Este conjunto de casos de prueba ha logrado alcanzar una cobertura del 92,8% para el paquete *acme.features.auditor.audit_record*.

Pruebas de rendimiento

En esta sección se va a presentar un análisis de las pruebas de rendimiento realizadas sobre los datos obtenidos en las pruebas funcionales. También se realizará una comparativa de estos datos al ejecutar las pruebas en diferentes equipos.

Con los datos generados por las pruebas se ha generado el siguiente gráfico que representa la media del tiempo de respuesta por cada una de las rutas que han sido involucradas en las pruebas funcionales.

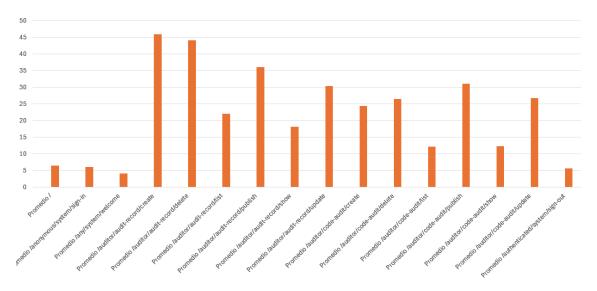


Ilustración 1 Gráfico de tiempo medio de respuesta por cada ruta

Podemos observar que las "funcionalidades" con tiempos más bajos no son realmente funcionalidades, sino las rutas de la página de inicio de sesión, la página de bienvenida y la ruta para cerrar sesión.

En cuanto a las funcionalidades reales, destacan dos sobre el resto: la creación y eliminación de Code Audits. Esto tiene sentido dado que la entidad de Code Audit es más compleja que la de Audit Record, ya que la primera engloba a la segunda. Por lo tanto, crear o eliminar una auditoría de código será una tarea más compleja.

El resto de las funcionalidades también tienen tiempos lógicos.

Veamos ahora el nivel de confianza (95%) para el conjunto de pruebas.

Columna1				
		Interval (ms)	12,4657881	15,4452284
Media	13,9555083	Interval (s)	0,01246579	0,01544523
Error típico	0,75713628			
Mediana	10,16485			
Moda	#N/D			
Desviación estándar	13,4164891			
Varianza de la muestra	180,00218			
Curtosis	21,2281005			
Coeficiente de asimetría	3,31167092			
Rango	127,4518			
Mínimo	2,006			
Máximo	129,4578			
Suma	4382,0296			
Cuenta	314			
Nivel de confianza (95,0%)	1,48972017			

Ilustración 2 Análisis de datos y nivel de confianza

En la Ilustración 2 podemos observar el análisis de los datos que se ha realizado para calcular el intervalo de nivel de confianza del 95% de nuestros datos. El cual es, en milisegundos [12.4657881, 15,4452284].

A continuación, realizaremos una comparación de los tiempos medios de respuesta y el nivel de confianza del 95% al ejecutar las pruebas en dos máquinas diferentes.

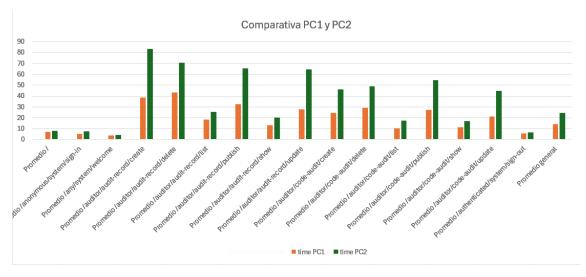


Ilustración 3 Comparativa tiempos PC1 y PC2

En el gráfico se comparan los tiempos medios de respuesta de las pruebas ejecutadas en dos máquinas diferentes: mi portátil (PC1) y el portátil de mi compañero Álvaro (PC2). Se observa que, en general, PC2 tiene tiempos de respuesta significativamente mayores en comparación con PC1. Esta diferencia en rendimiento puede deberse a variaciones en las especificaciones del hardware o en la configuración del sistema de cada máquina. En resumen, PC1 ofrece un mejor rendimiento en términos de tiempos de respuesta en todas las funcionalidades evaluadas.

PC1			PC2		
Media	13,9555083		Media	27,1986947	
Error típico	0,75713628		Error típico	1,50933524	
Mediana	10,16485		Mediana	16,8149153	
Moda	#N/D		Moda	#N/D	
Desviación estándar	13,4164891		Desviación estándar	26,1424532	
Varianza de la muestra	180,00218		Varianza de la muestra	683,427861	
Curtosis	21,2281005		Curtosis	4,91122522	
Coeficiente de asimetría	3,31167092		Coeficiente de asimetría	1,89142293	
Rango	127,4518		Rango	162,929	
Mínimo	2,006		Mínimo	2,3901	
Máximo	129,4578		Máximo	165,3191	
Suma	4382,0296		Suma	8159,6084	
Cuenta	314		Cuenta	300	
Nivel de confianza(95,0%)	1,48972017		Nivel de confianza(95,0%)	2,9702656	
Interval (ms)	12,4657881	15,4452284	Interval (ms)	24,2284291	30,1689603
Interval (s)	0,01246579	0,01544523	Interval (s)	0,02422843	0,03016896

Ilustración 4 Comparativa Nivel de confianza 95%

En la tabla se comparan los niveles de confianza al 95% de los tiempos de respuesta de las pruebas realizadas en dos máquinas: mi portátil (PC1) y el portátil de mi compañero Álvaro (PC2).

Los resultados muestran que el intervalo de confianza de PC1 es más estrecho ([12,4657881 ms, 15,4452284 ms]) en comparación con el de PC2 ([24,2284291 ms, 30,1689603 ms]). Esto indica que los tiempos de respuesta de PC1 son más consistentes y menos variables que los de PC2.

Conclusiones

En este informe, se han llevado a cabo diversas pruebas funcionales y de rendimiento para evaluar la eficacia y eficiencia del sistema. Las pruebas funcionales han mostrado una alta cobertura y han identificado un número mínimo de errores, demostrando que el sistema funciona correctamente en la mayoría de los casos. Se ha logrado una cobertura promedio del 93% en los diferentes paquetes, lo que indica una robusta implementación de las funcionalidades requeridas.

Las pruebas de rendimiento han revelado diferencias significativas en los tiempos de respuesta entre dos máquinas diferentes (PC1 y PC2). PC1 mostró tiempos de respuesta más rápidos y consistentes. El análisis del nivel de confianza al 95% respalda estos hallazgos, mostrando un intervalo de confianza más estrecho para PC1, lo que sugiere mayor estabilidad y menor variabilidad en los tiempos de respuesta.

En resumen, el sistema presenta un buen rendimiento funcional y una razonable eficiencia operativa en la mayoría de las condiciones evaluadas. Estos resultados proporcionan una base sólida para futuras optimizaciones y mejoras del sistema.

Bibliografía

Intencionalmente en blanco.