

Testing report D04

INFORMACIÓN DE GRUPO

Group: C2.043

Repository: `https://github.com/alepingar/Acme-Sf-C2.043`

Realizado por:

Student #4

UVUS: alepingar

Contact: alepingar@alum.us.es

Otros miembros:

Student #2

UVUS: alvmarmun1

Contact: alvmarmun1@alum.us.es

Date: Sevilla Mayo 20 ,06, 2024

TABLA DE CONTENIDO

Portada.....	1
Resumen Ejecutivo.....	3
Tabla de Revisiones.....	4
Introducción.....	5
Contenidos.....	6,7,8,9,10,11,12
Conclusiones.....	12
Bibliografía.....	13

RESUMEN EJECUTIVO

Este informe proporcionará un examen detallado tanto del proceso de prueba como de sus resultados. Contendrá secciones dedicadas a las pruebas funcionales y a las pruebas de rendimiento. Seguiré un enfoque preciso pero sencillo con el objetivo de promover la comprensión y asegurar la calidad del producto.

TABLA DE REVISIONES

Revisión	Fecha	Descripción
1	20/05/2024	<ul style="list-style-type: none">• Se realizaron las pruebas funcionales con sus correspondientes revisiones.
2	22/05/2024	<ul style="list-style-type: none">• Se realizaron las pruebas de rendimiento.
3	27/05/2024	<ul style="list-style-type: none">• Se realizó el presente informe y se corrigieron los fallos vistos en ambos tipos de pruebas.
4	27/06/2024	Se modifico el documento para cambiar los coverage y todo después de las correcciones

INTRODUCCIÓN

Este documento proporcionará un análisis detallado del procedimiento de pruebas y los resultados para las siguientes características:

- Operaciones de patrocinadores sobre Patrocinios.
- Operaciones de patrocinadores sobre Facturas.

El contenido de un informe de pruebas está organizado en dos capítulos:

- Pruebas Funcionales: Esta sección consiste en una compilación de casos de prueba implementados, categorizados por características. Cada caso de prueba está acompañado de una breve descripción y una evaluación de su efectividad en la detección de errores.
- Pruebas de Rendimiento: Este segmento incluye gráficos informativos y un intervalo de confianza del 95% para el tiempo de respuesta del proyecto durante las pruebas funcionales. Adicionalmente, se proporciona un contraste de hipótesis con un nivel de confianza del 95%.

CONTENIDOS

Pruebas Funcionales

Operaciones de patrocinadores sobre Patrocinios:

Caso de prueba 1: Listar

Para el comando de listado, simplemente realizamos la lista de patrocinios para los dos usuarios existentes con el rol de patrocinador.

Para hacking, consideramos acceder a la URL con un rol incorrecto. No es posible probar la situación de usuario incorrecto para este comando, ya que es una operación para cada usuario, de forma individual, es decir de tipo "mío".

Proporcionó una cobertura del 93.5%, cubriendo todas las instrucciones excepto una afirmación por defecto, lo cual es lógico. No se detectaron errores.

Caso de prueba 2: Mostrar

El caso de prueba número 2 consistió en mostrar los patrocinios pertenecientes a sponsor1 y de sponsor2.

Para hacking, tanto un usuario anónimo como el sponsor1 intentaron acceder y mostrar los datos de un patrocinio de sponsor2.

Proporcionó una cobertura del 97.0%, cubriendo todas las instrucciones excepto una afirmación por defecto, lo cual es lógico. No se detectaron errores.

Caso de prueba 3: Crear

Para este comando, intentamos crear un patrocinio con todos los tipos de datos inválidos.

Comenzando desde el formulario en blanco, cada atributo se llevó al límite. Una vez que comprobamos que el sistema rechaza datos inválidos, proporcionamos entradas válidas.

Proporcionó una cobertura del 96%, cubriendo todas las instrucciones excepto dos afirmaciones por defecto, lo cual es lógico y en algún if ha faltado alguna Branch pero no supone ningún problema ya que están evaluadas todas las condiciones internas de los bucles. No se detectaron errores.

Caso de prueba 4: Actualizar

Para este comando, actualizamos un patrocinio. Para cada atributo, verificamos que el sistema rechaza todos los tipos de datos inválidos. Luego, para cada atributo, verificamos que el sistema

acepta todos los tipos de datos válidos. Proporcionó una cobertura del 92.7%. Se cubrió todas las instrucciones excepto las afirmaciones por defecto y lo mismo que anteriormente con el create alguna Branch de algún if. No se detectaron errores.

Caso de prueba 5: Eliminar

Para el comando de eliminación, eliminamos un par de páginas de patrocinios de los sponsor1 y sponsor2. Proporcionó una cobertura del 92.4%. Cubrió todas las instrucciones excepto las afirmaciones por defecto y una de las Branch dentro de la condición del validate pero nuevamente cubierta.

Caso de prueba 6: Publicar

El procedimiento para probar el comando de publicación fue el siguiente:

Proceder de la misma manera que hicimos para el comando de actualización. Probar que el sistema rechaza datos inválidos y acepta entradas válidas.

Probar la regla de negocio que establece que un patrocinio no puede ser publicado si las facturas no suman la cantidad del patrocinio.

A continuación, publicamos un patrocinio. Proporcionó una cobertura del 95.8%. Cubrió todas las instrucciones excepto la afirmación por defecto de no nulo, junto con lo mismo que el create y update que falta en un par de if una Branch posible.

La siguiente imagen constrasta lo anterior:

		94,8 %	1.744	96	1.840
acme.features.sponsor.sponsorship	SponsorSponsorshipUpdateService	92,7 %	422	33	455
	SponsorSponsorshipPublishService	95,8 %	504	22	526
	SponsorSponsorshipDeleteService	92,4 %	208	17	225
	SponsorSponsorshipCreateService	96,0 %	385	16	401
	SponsorSponsorshipListMineService	93,5 %	58	4	62
	SponsorSponsorshipShowService	97,0 %	131	4	135
	SponsorSponsorshipController	100,0 %	36	0	36

Operaciones de patrocinadores sobre Facturas:

Caso de prueba 1: Listar

Se listaron todas las facturas para los dos usuarios con rol de patrocinador. Para hacking consideramos acceder a las listas con un rol diferente. Intentar acceder a una lista de facturas con un patrocinador diferente no es posible ya que es una lista independiente de cada usuario.

Proporcionó una cobertura del 93.3%, cubriendo todas las instrucciones excepto unas afirmaciones por defecto, lo cual es lógico. No se detectaron errores.

Caso de prueba 2: mostrar

En este caso de prueba mostramos varias páginas de facturas. Para hacking, tanto un usuario anónimo como el patrocinador 1 intentaron acceder y mostrar una factura del patrocinador 2. Proporcionó una cobertura del 94.9%, cubriendo todas las instrucciones excepto algunas afirmaciones por defecto lógicas y la de añadir una sponsorship a las opciones de asociación de las facturas, esta no se pudo pero sin embargo no supone ningún problema únicamente en el caso de que la sponsorship seleccionada no estuviera entre las que no están sin publicar para asociarlas esta se añadiría. No se detectaron errores.

Caso de prueba 3: crear

Para el comando de creación, intentamos crear una factura comenzando desde un formulario en blanco y verificando que el sistema rechaza datos inválidos. Luego, verificamos todos los rangos de entradas válidas. Proporcionó una cobertura del 95.1%, cubriendo todas las instrucciones excepto las afirmaciones por defecto, lo cual es lógico y alguna Branch dentro de algún if, pero no habría ningún problema al respecto. No se detectaron errores.

Caso de prueba 4: actualizar

En este caso de prueba intentamos actualizar una factura, verificando el rechazo de entradas inválidas y llevando los rangos de datos válidos al límite. Proporcionó una cobertura del 94.5%. Cubrió todas las instrucciones excepto las afirmaciones por defecto de no nulo y lo mismo que en el show con una instrucción.

No se detectaron errores.

Caso de prueba 5: eliminar

Se eliminaron un par de páginas de facturas. Proporcionó una cobertura del 88.6%. Como se vio antes ya que no hay restricciones para la eliminación de una factura, el unbind se eliminó ya que no se ejecutaba nunca y subimos la cobertura.

Caso de prueba 6: publicar

En este caso de prueba procedimos de la siguiente manera:

Probar el comando de publicación como si fuera el comando de actualización. Verificar que las entradas inválidas son rechazadas y que los datos válidos son aceptados.

Proporcionó una cobertura del 94.0%. Las condiciones que no se revisaron han sido exactamente las mismas que en el update, anteriormente explicadas y sin errores.

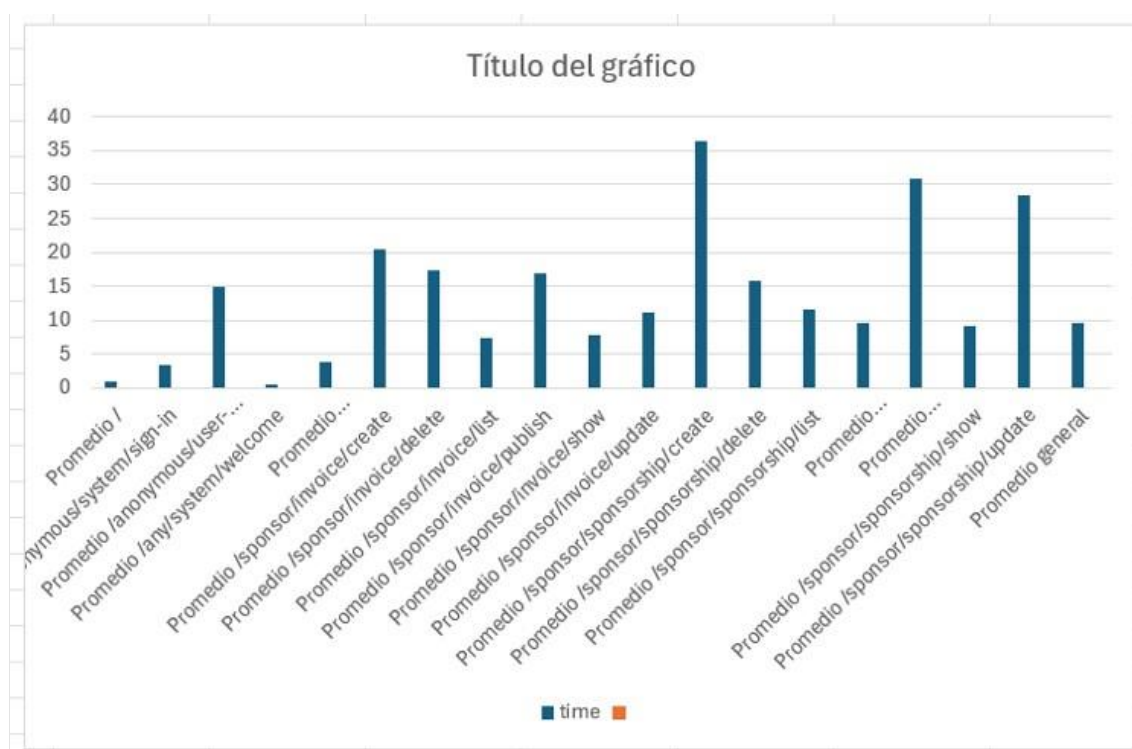
▼ acme.features.sponsor.invoice	94,1 %	1.695	106	1.801
> SponsorInvoicePublishService	94,0 %	502	32	534
> SponsorInvoiceUpdateService	94,5 %	409	24	433
> SponsorInvoiceCreateService	95,1 %	349	18	367
> SponsorInvoiceDeleteService	88,6 %	101	13	114
> SponsorInvoiceShowService	94,9 %	148	8	156
> SponsorInvoiceListInSponsors	92,6 %	88	7	95
> SponsorInvoiceListMineService	93,3 %	56	4	60
> SponsorInvoiceController.java	100,0 %	42	0	42

Pruebas de rendimiento

Comencemos revisando los datos de rendimiento recopilados durante las pruebas funcionales. Inicialmente, describiremos los resultados de las pruebas, seguidos de un examen de un contraste de hipótesis.

Datos de rendimiento

Estas pruebas se han realizado en un ordenador con las siguientes características: CPU AMD Ryzen 5 4500U y 8 GB de RAM. Tras completar las pruebas, se ha producido un gráfico que ilustra el tiempo de respuesta promedio para cada ruta de solicitud.



Lo que observamos es que las solicitudes que toman más tiempo son las de publicación, actualización y creación de los patrocinios. Esto tiene sentido ya que contienen validaciones más complicadas y extensas. Por ejemplo, la solicitud de publicación de patrocinio verifica si la suma de todas las facturas publicadas equivale al precio del patrocinio.

El resto de las funciones de visualización muestran un rendimiento consistente y lógico. Ahora calculemos un intervalo de confianza para toda la suite de pruebas.

Columna1					
			Interval(ms)	8,5398302	10,6443737
Media	9,59210193		Interval(s)	0,00853983	0,01064437
Error típico	0,53562876				
Mediana	7,13				
Moda	0,6013				
Desviación estándar	12,2024649				
Varianza de la muestra	148,90015				
Curtosis	12,6263926				
Coeficiente de asimetría	2,82908874				
Rango	109,4405				
Mínimo	0,3083				
Máximo	109,7488				
Suma	4978,3009				
Cuenta	519				
Nivel de confianza(95,0%)	1,05227173				

Basándonos en nuestro resumen del análisis de datos, calculamos el intervalo de confianza del 95.0% para nuestros datos, que varía de 8.53 a 10.64 milisegundos. Aunque este proyecto carece de un punto de referencia específico de rendimiento para comparar este intervalo de confianza, este tiempo de respuesta se considera muy bueno.

Contraste de hipótesis

Dado que no existen requisitos específicos de rendimiento para esta entrega, simularemos una prueba de hipótesis. Generaremos nuevos datos aumentando los datos de prueba reales en un 10%. Primero, crearemos un resumen del análisis de datos para ambas muestras de rendimiento y luego compararemos los resultados.

BEFORE						AFTER					
			Interval(ms)	8,5398302	10,6443737						
Media	9,59210193		Interval(s)	0,00853983	0,01064437	Media	11,1248677		Interval(ms)	9,81505038	12,434685
Error típico	0,53562876					Error típico	0,66651654		Interval(s)	0,00981505	0,01243468
Mediana	7,13					Mediana	7,63035				
Moda	0,6013					Moda	0,5797				
Desviación es	12,2024649					Desviación es	14,2640769				
Varianza de la	148,90015					Varianza de la	203,463889				
Curtosis	12,6263926					Curtosis	15,7217386				
Coeficiente de	2,82908874					Coeficiente de	2,86013117				
Rango	109,4405					Rango	140,822				
Mínimo	0,3083					Mínimo	0,3089				
Máximo	109,7488					Máximo	141,1309				
Suma	4978,3009					Suma	5095,1894				
Cuenta	519					Cuenta	458				
Nivel de confi	1,05227173					Nivel de confi	1,3098173				

Los resultados han aumentado naturalmente, lo que podría considerarse una disminución en el rendimiento. Sin embargo, comparar intuitivamente los intervalos de confianza es complicado. Por lo tanto, utilizaremos una prueba Z para facilitar esta comparación. Aquí están los resultados:

Prueba z para medias de dos muestras			
	<i>before</i>	<i>after</i>	
Media	9,729016622	11,2874759	
Varianza (conocida)	96,76005903	98,7049362	
Observaciones	538	477	
Diferencia hipotética de las medias	0		
z	-2,505898323		
P(Z<=z) una cola	0,006107037		
Valor crítico de z (una cola)	1,644853627		
Valor crítico de z (dos colas)	0,012214074		
Valor crítico de z (dos colas)	1,959963985		

Una prueba Z se basa en un valor llamado alfa, que se calcula como 1 menos el porcentaje del nivel de confianza. Para este caso, alfa será 0.05. Para interpretar el resultado de la prueba Z, comparamos el valor p de dos colas con alfa. Si el valor p es inferior a alfa, entonces comparamos los promedios para determinar si el nuevo promedio ha disminuido.

En nuestro caso, sin embargo, el valor p es 0.012214074, que es mayor que alfa. Esto indica que los cambios que implementamos no llevaron a ninguna mejora significativa. Aunque los tiempos de las muestras difieren, generalmente se consideran equivalentes en general.

CONCLUSIONES

Este riguroso proceso de pruebas formales nos permitió examinar nuestro código para descubrir cualquier error y poder solucionarlo con toda la información que se nos aporta testeando. En general, los resultados de las pruebas fueron muy favorables, logrando una cobertura de instrucciones casi completa, excepto por algunas ramas lógicas, y demostrando un rendimiento excelente, incluyendo un tiempo de respuesta promedio notable.

BIBLIOGRAFÍA

Intencionalmente en blanco: