



Escuela Técnica Superior de
Ingeniería Informática

Testing Report

Group C3.020 | Diseño y Pruebas II | 16/09/2024

Fecha	Versión	Autor
27/05/2024	1.0	Álvaro Carrera Bernal
26/06/2024	1.1	Álvaro Carrera Bernal
16/09/2024	1.2	Álvaro Carrera Bernal

Miembros:

- Álvaro Carrera Bernal (alvcarber1@alum.us.es)

Repositorio de Github: <https://github.com/DP2-C1-020/Acme-SF-Do4>

Contenido

Resumen ejecutivo	3
Introducción	4
Contenido	5
Profiling your Project	13
Conclusiones	14
Bibliografía.....	15

Resumen ejecutivo

Este informe ofrecerá un análisis detallado del procedimiento de pruebas y sus resultados, incluyendo una sección específica para las pruebas funcionales y otra para las pruebas de rendimiento. Emplearemos un enfoque claro y conciso para facilitar la comprensión y asegurar un producto final de alta calidad.

Introducción

Este documento establece las directrices para crear un informe de pruebas, dividido en dos capítulos principales: pruebas funcionales y pruebas de rendimiento. El propósito es proporcionar un análisis detallado y organizado de los casos de prueba implementados, así como evaluar el rendimiento del proyecto en diferentes entornos.

El capítulo de pruebas funcionales presentará los casos de prueba agrupados por características, con una descripción breve y una evaluación de su efectividad para identificar errores. Este enfoque riguroso garantiza una evaluación completa y sistemática de cada aspecto funcional del proyecto.

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 procesar solicitudes. Se comparará el rendimiento del proyecto en dos computadoras distintas, realizando un análisis basado en un contraste de hipótesis con un 95% de confianza para determinar cuál de las dos computadoras es más potente.

Este informe ha sido diseñado para ser claro y fácil de entender, facilitando la interpretación de los resultados y asegurando que se puedan tomar decisiones informadas para mejorar la calidad final del producto.

Contenido

Pruebas funcionales

Comenzaremos analizando las pruebas funcionales que corresponden al requisito 6: *Operations by clients on contracts*. Dentro de este requisito, tendremos en cuenta cada funcionalidad por separado.

- List
Listamos los contratos de un Cliente. Para las pruebas de hacking, se intentó acceder a esta URL utilizando diferentes roles incorrectos o directamente sin tener ningún rol, entre ellos se probó como anónimo, manager1, mismo rol pero diferente usuario (client2).
Esta prueba ha proporcionado una cobertura del 94,4%. No se encontraron bugs.
- Show
Se han seleccionado varios contratos de un cliente para ver los detalles de estos. Para las pruebas de hacking, se intentó acceder a esta URL con diferentes roles que no son Client1 y también se probó a mostrar los detalles sin tener la sesión iniciada. Se demostró que no se puede acceder a esta información a no ser que tengas el rol correspondiente, en este caso Client1.
Esta prueba ha proporcionado una cobertura del 95,5%, no encontrándose errores.
- Delete
Se seleccionaron varios contratos no publicados y se procedió a la eliminación de estos. Para las pruebas de hacking, se ha intentado eliminar contratos una vez que muestras los detalles de los contratos, acción que no es posible al no estar la sesión iniciada o no tener el rol correspondiente. También se ha probado a eliminar un contrato ya publicado, no siendo posible esta acción. Además, se ha intentado eliminar un contrato con un rol diferente a Client o anónimo, mismo rol pero diferente usuario (client2).
Esta prueba ha proporcionado una cobertura del 85,1%.
- Create
Se han implementado todas las restricciones especificadas por el cliente para la creación de contratos, realizando pruebas con datos mínimos, intermedios y máximos para asegurar el manejo adecuado de estos datos. Además, se verificó que el sistema rechaza correctamente los datos incorrectos en todos los campos del formulario antes de crear el contrato. También se comprobó que acepta los caracteres de hacking y charsets.
Para las pruebas de hacking, se ha probado a realizar una creación de contratos con otro rol que no está permitido, es decir, un rol diferente a Client1.
Esta prueba ha proporcionado una cobertura del 95,2%.
- Update
Para esta prueba, se seleccionó un contrato no publicado y se actualizó, pero antes se tuvo en cuenta todas las restricciones que contienen los campos del formulario. Se verificó que el sistema rechaza datos incorrectos en todos los campos del formulario. Posteriormente, se probaron todos los límites correspondientes a cada campo del formulario. También se comprobó que acepta los caracteres de hacking y charsets.

Para las pruebas de hacking, se llevó a cabo el mismo procedimiento que con Create, es decir, se ha probado a realizar una actualización de contratos con otro rol que no está permitido, como puede ser anónimo o manager, mismo rol pero diferente usuario (client2) y por último, actualizar un contrato que ya este publicado, acción que no es posible realizar.

Esta prueba ha proporcionado una cobertura del 93,2%.

















- Publish

Para esta prueba, se seleccionó un contrato no publicado y se publicó, pero antes se tuvo en cuenta todas las restricciones que contienen los campos del formulario. Se verificó que el sistema rechaza datos incorrectos en todos los campos del formulario. Posteriormente, se probaron todos los límites correspondientes a cada campo del formulario. También se comprobó que acepta los caracteres de hacking y charsets.

Para las pruebas hacking, se llevó a cabo el mismo procedimiento, se intento publicar un contrato ya publicado, y también se intentó publicar un contrato con un rol diferente que no fuese Client (anónimo o manager, por ejemplo) o con el mismo rol pero diferente usuario.

Esta prueba ha proporcionado una cobertura del 93,2%.

Durante la ejecución de estos tests, se han detectado varios errores con respecto a las validaciones correspondientes a cada campo de los formularios.

▼  <code>acme.features.client.contract</code>	 93,2 %
>  <code>ClientContractPublishService.java</code>	 93,2 %
>  <code>ClientContractUpdateService.java</code>	 93,2 %
>  <code>ClientContractDeleteService.java</code>	 85,1 %
>  <code>ClientContractCreateService.java</code>	 95,2 %
>  <code>ClientContractListService.java</code>	 94,4 %
>  <code>ClientContractShowService.java</code>	 95,5 %
>  <code>ClientContractController.java</code>	 100,0 %

Este conjunto de casos de prueba ha logrado alcanzar una cobertura del 93,2% para el paquete `acme.features.client.contract`. Como podemos comprobar, la mayoría de la funcionalidad está cubierta, aunque encontramos dos excepciones:

- La primera, en `ClientContractDeleteService`, puesto que al realizar esta acción no se está ejecutando la función `unbind`.
- La segunda, en `ClientContractUpdateService` y `ClientContractPublishService`, ya que en ambos se quedan sin comprobar el cálculo de una variable que se utiliza para la comparación del coste del proyecto con la suma del precio de los contratos.

Ahora, vamos a pasar con el análisis de las pruebas funcionales del requisito 7: *Operations by clients on progress logs*.

- List

Listamos los registros de progresos correspondientes a los contratos. Para las pruebas de hacking se intentó acceder a esta URL utilizando diferentes roles y también sin estar registrado. Esta prueba ha proporcionado una cobertura del 95,9%.

- Show

Se han seleccionado varios registros de progreso de un contrato para ver los detalles de estos. Para las pruebas de hacking, se intentó acceder a esta URL con diferentes roles que no son Client1 y también se probó a mostrar los detalles sin tener la sesión iniciada. Se demostró que no se puede acceder a esta información a no ser que tengas el rol correspondiente, en este caso Client1.

Esta prueba ha proporcionado una cobertura del 96,4%, no encontrándose errores.

- Delete

Se seleccionaron varios registros de progreso no publicados y se procedió a la eliminación de estos. Para las pruebas de hacking, se ha intentado eliminar un registro de progreso una vez que muestras los detalles de los registros de progreso, acción que no es posible al no estar la sesión iniciada o no tener el rol correspondiente. También se ha probado a eliminar un registro de progreso ya publicado, no siendo posible esta acción.

Esta prueba ha proporcionado una cobertura del 83,7%.

- Create

Se han implementado todas las restricciones especificadas por el cliente para la creación de registros de progreso, realizando pruebas con datos mínimos, intermedios y máximos para asegurar el manejo adecuado de estos datos. Además, se verificó que el sistema rechaza correctamente los datos incorrectos en todos los campos del formulario antes de crear el registro de progreso. También se comprobó que acepta los caracteres de hacking y charsets.

Para las pruebas de hacking, se ha probado a realizar una creación de registros de progreso con otro rol que no está permitido, es decir, un rol diferente a Client1 anónimo o manager, por ejemplo, y también mismo rol pero diferente usuario.

Esta prueba ha proporcionado una cobertura del 94,8%.

- Update

Para esta prueba, se seleccionó un registro de progreso no publicado y se actualizó, teniendo en cuenta todas las restricciones que contienen los campos del formulario. Se verificó que el sistema rechaza datos incorrectos en todos los campos del formulario. Posteriormente, se probaron todos los límites correspondientes a cada campo del formulario. También se comprobó que acepta los caracteres de hacking y charsets.

En cuanto a las pruebas de seguridad, se llevó a cabo el mismo procedimiento que con la creación, es decir, se intentó actualizar registros de progreso utilizando un rol no autorizado, anónimo o manager, por ejemplo, mismo rol pero diferente usuario y, por último, mismo rol y usuario, pero con un progress log ya publicado, acción que no es posible.

Estas pruebas lograron una cobertura del 93,3%.

- Publish

















Para esta prueba, se seleccionó un registro de progreso no publicado y se publicó, teniendo en cuenta todas las restricciones de los campos del formulario. Se verificó que el sistema rechaza datos incorrectos en todos los campos del formulario. Posteriormente, se probaron todos los límites correspondientes a cada campo del formulario. También se comprobó que acepta los caracteres de hacking y charsets.

En cuanto a las pruebas de seguridad, se llevó a cabo el mismo procedimiento: se intentó publicar un registro de progreso ya publicado y también se intentó publicar

un registro de progreso utilizando un rol diferente al de Client, y con el mismo rol pero diferente usuario.

Estas pruebas lograron una cobertura del 93,4%.

Durante la ejecución de estos test, se han detectado varios errores con respecto a las validaciones correspondientes a cada campo de los formularios.

▼  acme.features.client.progressLog	 93,4 %
>  ClientProgressLogsDeleteService.java	 83,7 %
>  ClientProgressLogsPublishService.java	 93,4 %
>  ClientProgressLogsUpdateService.java	 93,3 %
>  ClientProgressLogsCreateService.java	 94,8 %
>  ClientProgressLogsListService.java	 95,9 %
>  ClientProgressLogsShowService.java	 96,4 %
>  ClientProgressLogsController.java	 100,0 %

Este conjunto de casos de prueba ha logrado alcanzar una cobertura del 93,4% para el paquete *acme.features.client.progressLog*. Como podemos comprobar, la mayoría de la funcionalidad está cubierta, aunque encontramos una excepción:

- La primera, en *ClientProgressLogsDeleteService*, puesto que al realizar esta acción no se está ejecutando la función *unbind*.

En la mayoría de los casos, las fallas de línea de cobertura son causadas por las verificaciones *assert not null* debido a que en la aplicación no es posible verificar si una entidad es nula, ya que no tenemos una lista con una entidad nula para poder seleccionarla y probarla.

Pruebas de rendimiento

En esta sección se presenta un análisis del rendimiento basado en los datos obtenidos durante las pruebas funcionales. Además, se comparan estos resultados al ejecutar las pruebas en diferentes equipos. Los datos generados se han utilizado para crear un gráfico que muestra el tiempo de respuesta promedio para cada una de las rutas involucradas en las pruebas funcionales.

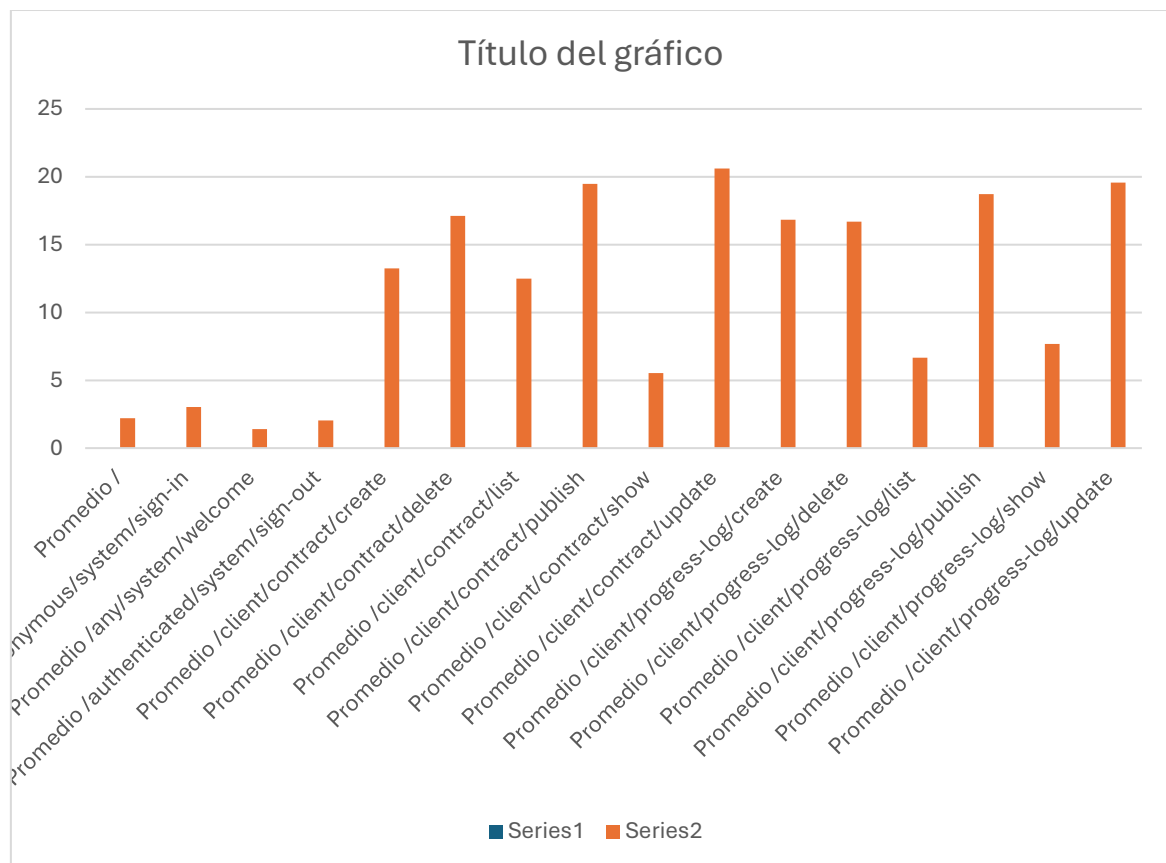


Ilustración 1 Gráfico de tiempo medio de respuesta para cada ruta en mi pc

Se evidencia que las acciones con tiempos más cortos no son realmente funciones complejas, sino más bien las operaciones básicas de navegación, como acceder a la página de inicio de sesión, la página de bienvenida y la opción para cerrar sesión. También tenemos las listas de contratos o registros de progreso y sus respectivas vistas de detalles.

En cuanto a las funcionalidades reales, destaca sobre el resto la actualización de Contract y la actualización de Progress Log.

El resto de las funcionalidades también tienen tiempos lógicos

Vamos a observar ahora el nivel de confianza (95%) para el conjunto de pruebas.

Columna1				
			Interval (ms)	8,61859102
			Interval(s)	0,00861859
Media	9,153515121			9,68843922
Error típico	0,272566028			
Mediana	5,7414			
Moda	1,5015			
Desviación estándar	8,26733696			
Varianza de la muestra	68,3488604			
Curtosis	1,809184252			
Coefficiente de asimetría	1,149030812			
Rango	64,1136			
Mínimo	0,8733			
Máximo	64,9869			
Suma	8421,233912			
Cuenta	920			
Nivel de confianza(95,0%)	0,534924101			

Ilustración 2 Análisis de datos y nivel de confianza en mi pc

El nivel de confianza del 95% en el reporte de testing indica que hay un 95% de probabilidad de que el valor medio verdadero de la muestra se encuentre dentro del intervalo dado.

Esto significa que, basándonos en los datos, podemos estar bastante seguros de que el tiempo promedio medido cae dentro de ese rango.

Ahora, vamos a realizar una comparación de los tiempos medios de respuesta y el nivel de confianza 95% al ejecutar las pruebas en dos ordenadores diferentes.



Ilustración 3 Gráfico de tiempo medio de respuesta para cada ruta en el pc de mi compañero Guillermo

En la gráfica anterior, podemos ver los tiempos medios de respuesta de las pruebas ejecutadas en el pc de mi compañero Guillermo. Se observa que en general, PC2 tienen tiempos de respuesta significativamente mayores en comparación con mi PC. 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.

Columna1					
			Interval (ms)	13,34041741	15,0880709
Media	14,2142442		Interval (s)	0,013340417	0,01508807
Error típico	0,44532278				
Mediana	8,583				
Moda	2,0406				
Desviación estándar	14,416358				
Varianza de la muestra	207,831377				
Curtosis	6,10970749				
Coefficiente de asimetría	1,86649825				
Rango	114,229				
Mínimo	1,8237				
Máximo	116,0527				
Suma	14896,5279				
Cuenta	1048				
Nivel de confianza(95,0%)	0,87382677				

Ilustración 4 Análisis de datos y nivel de confianza en pc de Guillermo

En las tablas *Ilustración 2 e Ilustración 4* 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 Guillermo (PC2).

Los intervalos de confianza del 95% para PC2 (13.34, 15.09) y PC1 (8.62, 9.69) indican que podemos estar un 95% seguros de que los valores verdaderos de estos parámetros se encuentran dentro de esos rangos. La estrechez de ambos intervalos sugiere que las estimaciones de PC1 y PC2 son precisas y confiables, con PC2 teniendo un valor mayor que PC1.

A	B	C	D	E	F	G	H	I	J
BEFORE	AFTER								
75,5404	100,8756								
19,5796	14,7475								
7,2546	10,9946		BEFORE				Columna1		
11,3847	34,6418								
3,6186	5,1681		Media	8,2627529			Media	14,144115	
3,2663	5,1154		Error típico	0,2616625			Error típico	0,4502706	
35,7304	76,6877		Mediana	4,4252			Mediana	8,3964	
18,0208	31,7985		Moda	18,1872			Moda	15,9856	
32,2354	82,1599		Desviación estándar	8,4017708			Desviación estándar	14,457825	
22,8676	44,6521		Varianza de la muestra	70,589752			Varianza de la muestra	209,0287	
14,4872	20,5691		Curtosis	5,5406591			Curtosis	6,1800813	
18,5216	42,2173		Coefficiente de asimetría	1,6793957			Coefficiente de asimetría	1,8881144	
22,2325	52,4567		Rango	74,6671			Rango	114,229	
11,599	21,4133		Mínimo	0,8733			Mínimo	1,8237	
22,2222	43,8653		Máximo	75,5404			Máximo	116,0527	
24,7721	56,3279		Suma	8518,8982			Suma	14582,582	
11,5091	22,9491		Cuenta	1031			Cuenta	1031	
17,0591	25,6547		Nivel de confianza(95,0%)	0,5134524			Nivel de confianza(95,0%)	0,8835525	
21,9429	34,3141								
12,143	21,0523		Interval (ms)	7,7493004	8,7762053		Interval (ms)	13,260562	15,027667
19,1431	36,9808		Interval (s)	0,0077493	0,0087762		Interval (s)	0,0132606	0,0150277

Ilustración 5 Comparación de tiempo entre mi pc y pc de Guillermo

Prueba z para medias de dos muestras		
	BEFORE	AFTER
Media	8,26275286	14,1441145
Varianza (conocida)	70,5897521	209,028704
Observaciones	1031	1031
Diferencia hipotética de las medias	0	
z	-11,2933894	
P(Z<=z) una cola	0	
Valor crítico de z (una cola)	1,64485363	
Valor crítico de z (dos colas)	0	
Valor crítico de z (dos colas)	1,95996398	

Ilustración 6. Prueba Z para medias de dos muestras

Dado que el p-valor de dos colas es prácticamente 0, podemos rechazar la hipótesis nula, lo que significa que hay una clara diferencia entre las dos medias ("antes" y "después"). Dado que la media "después" (14.14) es mayor que la media "antes" (8.26), esto nos muestra que el rendimiento no mejoró, sino que en realidad empeoró después de los cambios.

La prueba indica una diferencia estadística de forma significativa, pero en la dirección contraria a la esperada si el objetivo era reducir los tiempos. Por lo tanto, podría ser necesario realizar más investigaciones o modificaciones para mejorar el rendimiento.

Profiling your Project

Se llevó a cabo un análisis con la herramienta VisualVM para identificar los cuellos de botella. Podemos ver las áreas donde se encontraron los problemas de rendimiento más importantes.

El método `acme.features.client.contract.ClientContractPublishService.validate()` consume tiempo de CPU debido a las operaciones internas que realiza, como la validación de datos, integridad de los contratos y la consistencia de los estados, lo que incrementa su carga de procesamiento. En contraste, otros métodos como `acme.features.client.contract.ClientContractCreateService.bind()` muestran 0 ms de tiempo de CPU porque posiblemente realizan operaciones triviales, delegan la mayor parte del trabajo a otros componentes, o no fueron invocados con suficiente frecuencia durante el monitoreo.

Name	Self Time (CPU)	Total Time (CPU)
<code>acme.features.client.progressLog.ClientProgressLogsCreateService.bind ()</code>	0,0 ms (0 %)	603 ms (15,4 %)
<code>acme.features.client.contract.ClientContractUpdateService.bind ()</code>	0,0 ms (0 %)	554 ms (14,1 %)
<code>acme.features.client.contract.ClientContractListService.load ()</code>	0,0 ms (0 %)	485 ms (12,4 %)
<code>acme.features.client.progressLog.ClientProgressLogsUpdateService.bind ()</code>	0,0 ms (0 %)	481 ms (12,3 %)
<code>acme.features.client.progressLog.ClientProgressLogsPublishService.bind ()</code>	0,0 ms (0 %)	317 ms (8,1 %)
<code>acme.features.client.contract.ClientContractPublishService.bind ()</code>	0,0 ms (0 %)	225 ms (5,7 %)
<code>acme.features.client.contract.ClientContractUpdateService.validate ()</code>	0,0 ms (0 %)	162 ms (4,1 %)
<code>acme.features.client.progressLog.ClientProgressLogsUpdateService.validate ()</code>	16,1 ms (56,4 %)	108 ms (2,7 %)
<code>acme.features.client.contract.ClientContractCreateService.unbind ()</code>	0,0 ms (0 %)	99,5 ms (2,5 %)
<code>acme.features.client.contract.ClientContractShowService.authorise ()</code>	0,0 ms (0 %)	94,2 ms (2,4 %)
<code>acme.features.client.contract.ClientContractCreateService.validate ()</code>	0,0 ms (0 %)	84,0 ms (2,1 %)
<code>acme.features.client.contract.ClientContractPublishService.validate ()</code>	0,0 ms (0 %)	81,6 ms (2,1 %)
<code>acme.features.client.progressLog.ClientProgressLogsPublishService.validate ()</code>	0,0 ms (0 %)	60,9 ms (1,6 %)
<code>acme.features.client.progressLog.ClientProgressLogsCreateService.authorise ()</code>	0,0 ms (0 %)	47,1 ms (1,2 %)
<code>acme.features.client.contract.ClientContractUpdateService.checkBudgetLessThanProjectCost ()</code>	0,0 ms (0 %)	43,3 ms (1,1 %)
<code>acme.features.client.contract.ClientContractCreateService.load ()</code>	0,0 ms (0 %)	42,8 ms (1,1 %)
<code>acme.features.client.contract.ClientContractCreateService.perform ()</code>	0,0 ms (0 %)	41,4 ms (1,1 %)
<code>acme.features.client.contract.ClientContractPublishService.load ()</code>	0,0 ms (0 %)	40,4 ms (1 %)
<code>acme.features.client.contract.ClientContractDeleteService.bind ()</code>	0,0 ms (0 %)	29,7 ms (0,8 %)
<code>acme.features.client.contract.ClientContractCreateService.bind ()</code>	0,0 ms (0 %)	29,4 ms (0,8 %)
<code>acme.features.client.contract.ClientContractPublishService.checkBudgetLessThanProjectCost ()</code>	0,0 ms (0 %)	27,5 ms (0,7 %)
<code>acme.features.client.progressLog.ClientProgressLogsListService.authorise ()</code>	0,0 ms (0 %)	25,4 ms (0,6 %)
<code>acme.features.client.progressLog.ClientProgressLogsUpdateService.authorise ()</code>	0,0 ms (0 %)	25,3 ms (0,6 %)
<code>acme.features.client.contract.ClientContractUpdateService.load ()</code>	0,0 ms (0 %)	21,9 ms (0,6 %)
<code>acme.features.client.progressLog.ClientProgressLogsListService.load ()</code>	0,0 ms (0 %)	21,7 ms (0,6 %)
<code>acme.features.client.contract.ClientContractDeleteService.perform ()</code>	0,0 ms (0 %)	17,7 ms (0,5 %)
<code>acme.features.client.progressLog.ClientProgressLogsPublishService.perform ()</code>	0,0 ms (0 %)	16,1 ms (0,4 %)
<code>acme.features.client.contract.ClientContractShowService.load ()</code>	0,0 ms (0 %)	16,1 ms (0,4 %)
<code>acme.features.client.contract.ClientContractCreateService.authorise ()</code>	0,0 ms (0 %)	16,0 ms (0,4 %)
<code>acme.features.client.progressLog.ClientProgressLogsListService.unbind ()</code>	0,0 ms (0 %)	14,9 ms (0,4 %)
<code>acme.features.client.progressLog.ClientProgressLogsDeleteService.bind ()</code>	0,0 ms (0 %)	14,8 ms (0,4 %)
<code>acme.features.client.progressLog.ClientProgressLogsShowService.authorise ()</code>	0,0 ms (0 %)	14,2 ms (0,4 %)
<code>acme.features.client.contract.ClientContractListService.authorise ()</code>	12,4 ms (43,6 %)	12,4 ms (0,3 %)

Conclusiones

Al revisar los contratos y los registros de progreso en la aplicación, se han detectado una serie de errores que señalan discrepancias entre las respuestas esperadas y las obtenidas. Estas discrepancias sugieren problemas en la lógica de la aplicación o en el procesamiento de datos.

En resumen, los errores indican que las respuestas de la aplicación no coinciden con las expectativas en cuanto a la estructura de los datos recibidos. Esto podría deberse a fallos en la lógica de la aplicación o en la forma en que se procesan y devuelven los datos, debido a que se generan de forma aleatoria.

Estos problemas se encontraron durante las pruebas de las operaciones de Actualización, Eliminación, Publicación o Creación, lo que sugiere que están vinculados con la manipulación de datos durante estas operaciones específicas en la aplicación.

El informe destaca que se realizaron pruebas exhaustivas para evaluar tanto la funcionalidad como el rendimiento del sistema. Las pruebas funcionales indicaron una amplia cobertura y una cantidad mínima de errores, lo que sugiere que el sistema opera correctamente en la mayoría de los casos. Además, se logró una cobertura promedio del 93.30% en los diferentes aspectos del sistema, lo que demuestra una implementación robusta de las funciones requeridas.

Por otro lado, las pruebas de rendimiento revelaron diferencias notables en los tiempos de respuesta entre dos máquinas distintas, PC1 y PC2. PC2 exhibió tiempos de respuesta más rápidos y consistentes en comparación con PC1. Esta observación se respalda con un análisis de intervalos de confianza del 95%, donde se encontró un intervalo más estrecho para PC1 (8.62, 9.69), lo que indica una mayor precisión y menos variabilidad en los tiempos de respuesta de esta máquina en comparación con PC2 (13.34, 15.09).

En conclusión, el sistema demostró un buen rendimiento en cuanto a su funcionalidad y una eficiencia operativa razonable en la mayoría de las condiciones evaluadas. Estos resultados proporcionan una base sólida para futuras optimizaciones y mejoras del sistema.

Bibliografía

Intencionadamente en blanco.