



Escuela Técnica Superior de
Ingeniería Informática

Testing Report

Group C2.020 | Diseño y Pruebas II | 26/06/2024

Fecha	Versión	Autor
27/05/2024	1.0	Álvaro Carrera Bernal
26/06/2024	1.1	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	12
Conclusiones.....	13
Bibliografía.....	14

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 96,4%, 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 91,4%.

- 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 91,2%.

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

▼  acme.features.client.contract	 92,3 %
>  ClientContractPublishService.java	 91,2 %
>  ClientContractUpdateService.java	 91,4 %
>  ClientContractDeleteService.java	 85,1 %
>  ClientContractCreateService.java	 95,2 %
>  ClientContractListService.java	 94,4 %
>  ClientContractShowService.java	 96,4 %
>  ClientContractController.java	 100,0 %

Este conjunto de casos de prueba ha logrado alcanzar una cobertura del 92,3% 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 93,4%.

- 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 92,7%.

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		93,2 %
>	ClientProgressLogsDeleteService.java		83,7 %
>	ClientProgressLogsPublishService.java		92,7 %
>	ClientProgressLogsUpdateService.java		93,4 %
>	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 93,2% 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.

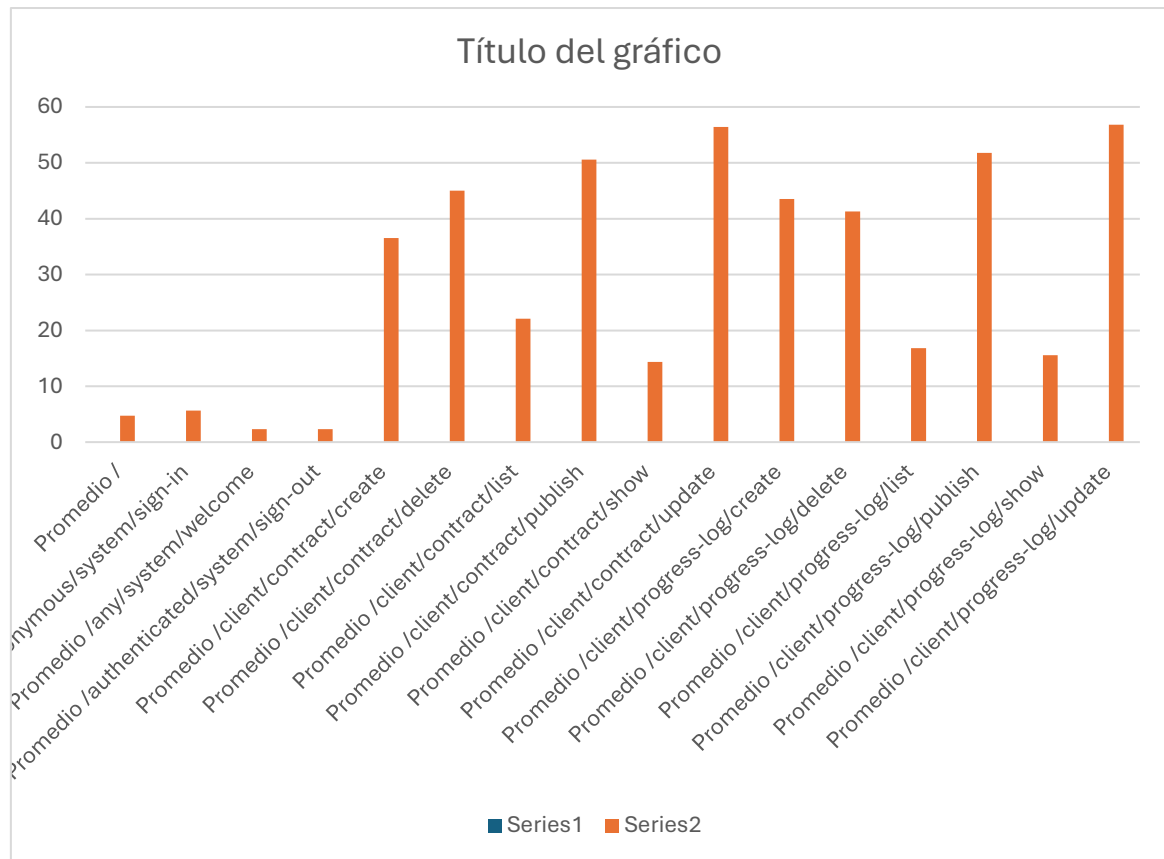


Ilustración 1 Gráfico de tiempo medio de respuesta para cada ruta en mi pc

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)	24.28283031	27.8526021
Media	26.0677162		Interval(s)	0.02428283	0.0278526
Error típico	0.9091182				
Mediana	17.46685				
Moda	#N/D				
Desviación estándar	24.2242046				
Varianza de la muestra	586.812088				
Curtosis	5.876388				
Coefficiente de asimetría	1.6869856				
Rango	205.776				
Mínimo	0.9269				
Máximo	206.7029				
Suma	18508.0785				
Cuenta	710				
Nivel de confianza(95.0%)	1.78488589				

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.

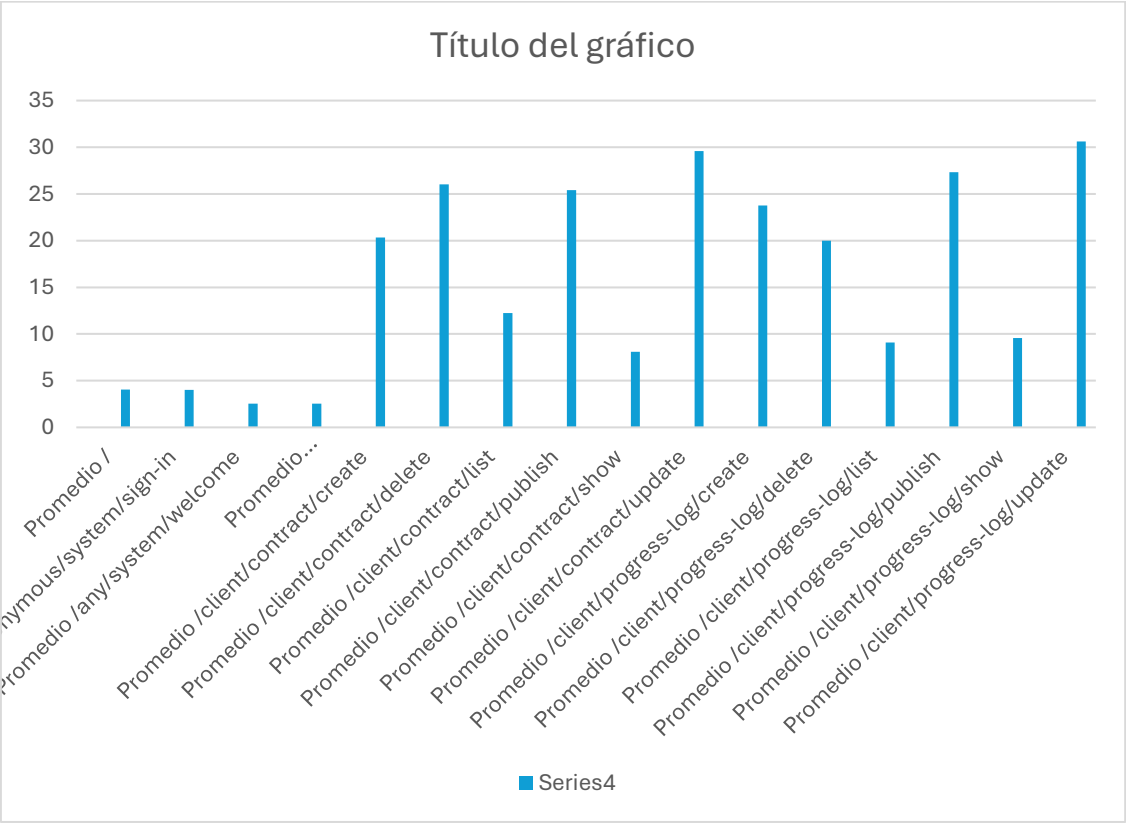


Ilustración 3 Gráfico de tiempo medio de respuesta para cada ruta en el pc de mi compañero Guillermo

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 menores 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					
Media	14.4403091		Interval (ms)	13.51781535	15.3628029
Error típico	0.46986412		Interval (s)	0.013517815	0.0153628
Mediana	9.4348				
Moda	24.7205				
Desviación estándar	12.5110972				
Varianza de la muestra	156.527553				
Curtosis	16.6847853				
Coefficiente de asimetría	2.42974745				
Rango	142.0465				
Mínimo	1.7648				
Máximo	143.8113				
Suma	10238.1792				
Cuenta	709				
Nivel de confianza(95.0%)	0.92249376				

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 (13.51, 15.36) y PC1 (24.28, 27.85) 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 PC1 teniendo un valor mayor que PC2.

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.progressLog.ClientProgressLogsListService.unbind()` consume tiempo de CPU debido a las operaciones internas complejas que realiza, como cálculos, acceso a datos y limpieza de recursos, lo que incrementa su carga de trabajo. En contraste, otros métodos muestran 0 ms de tiempo de CPU porque realizan operaciones triviales, delegan tareas a otros componentes, o no se ejecutaron con suficiente frecuencia durante el monitoreo.

Name	Self Time (CPU)	Total Time (CPU)
acme.features.client.progressLog.ClientProgressLogsListService.unbind ()	11,8 ms (100%)	14,8 ms (0,5%)
acme.features.client.progressLog.ClientProgressLogsShowService.authorise ()	0,0 ms (0%)	15,5 ms (0,5%)
acme.features.client.progressLog.ClientProgressLogsPublishService.authorise ()	0,0 ms (0%)	15,4 ms (0,5%)
acme.features.client.progressLog.ClientProgressLogsDeleteService.authorise ()	0,0 ms (0%)	11,3 ms (0,4%)
acme.features.client.contract.ClientContractShowService.load ()	0,0 ms (0%)	0,0 ms (0%)
acme.features.client.contract.ClientContractUpdateService.authorise ()	0,0 ms (0%)	13,7 ms (0,4%)
acme.features.client.progressLog.ClientProgressLogsUpdateService.authorise ()	0,0 ms (0%)	5,98 ms (0,2%)
acme.features.client.progressLog.ClientProgressLogsUpdateService.bind ()	0,0 ms (0%)	528 ms (16,7%)
acme.features.client.progressLog.ClientProgressLogsListService.load ()	0,0 ms (0%)	15,6 ms (0,5%)
acme.features.client.contract.ClientContractUpdateService.checkBudgetLessThanProjectCost ()	0,0 ms (0%)	0,0 ms (0%)
acme.features.client.contract.ClientContractUpdateService.validate ()	0,0 ms (0%)	92,1 ms (2,9%)
acme.features.client.contract.ClientContractUpdateService.bind ()	0,0 ms (0%)	634 ms (2,0%)
acme.features.client.contract.ClientContractShowService.unbind ()	0,0 ms (0%)	12,7 ms (0,4%)
acme.features.client.contract.ClientContractShowService.authorise ()	0,0 ms (0%)	71,0 ms (2,2%)
acme.features.client.progressLog.ClientProgressLogsPublishService.validate ()	0,0 ms (0%)	3,7 ms (0,1%)
acme.features.client.progressLog.ClientProgressLogsPublishService.load ()	0,0 ms (0%)	0,0 ms (0%)
acme.features.client.progressLog.ClientProgressLogsPublishService.bind ()	0,0 ms (0%)	204 ms (6,4%)
acme.features.client.contract.ClientContractPublishService.validate ()	0,0 ms (0%)	30,7 ms (1%)
acme.features.client.contract.ClientContractPublishService.bind ()	0,0 ms (0%)	210 ms (6,6%)
acme.features.client.progressLog.ClientProgressLogsDeleteService.bind ()	0,0 ms (0%)	16,0 ms (0,5%)
acme.features.client.contract.ClientContractDeleteService.bind ()	0,0 ms (0%)	32,6 ms (1%)
acme.features.client.progressLog.ClientProgressLogsListService.authorise ()	0,0 ms (0%)	24,9 ms (0,8%)
acme.features.client.progressLog.ClientProgressLogsCreateService.perform ()	0,0 ms (0%)	0,0 ms (0%)
acme.features.client.progressLog.ClientProgressLogsCreateService.unbind ()	0,0 ms (0%)	12,0 ms (0,4%)
acme.features.client.contract.ClientContractListService.unbind ()	0,0 ms (0%)	15,7 ms (0,5%)
acme.features.client.contract.ClientContractCreateService.perform ()	0,0 ms (0%)	32,8 ms (1%)
acme.features.client.progressLog.ClientProgressLogsCreateService.authorise ()	0,0 ms (0%)	87,0 ms (2,7%)
acme.features.client.progressLog.ClientProgressLogsCreateService.validate ()	0,0 ms (0%)	47,7 ms (1,5%)
acme.features.client.progressLog.ClientProgressLogsCreateService.bind ()	0,0 ms (0%)	452 ms (14,3%)
acme.features.client.progressLog.ClientProgressLogsCreateService.load ()	0,0 ms (0%)	3,11 ms (0,1%)
acme.features.client.contract.ClientContractCreateService.bind ()	0,0 ms (0%)	15,3 ms (0,5%)
acme.features.client.contract.ClientContractCreateService.validate ()	0,0 ms (0%)	88,9 ms (2,8%)
acme.features.client.contract.ClientContractCreateService.unbind ()	0,0 ms (0%)	148 ms (4,7%)

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 91.65% 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. PC1 exhibió tiempos de respuesta más rápidos y consistentes en comparación con PC2. Esta observación se respalda con un análisis de intervalos de confianza del 95%, donde se encontró un intervalo más estrecho para PC1 (24.28, 27.85), indicando una mayor precisión y menos variabilidad en los tiempos de respuesta de esta máquina en comparación con PC2 (13.51, 15.36).

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.