

INFORME DE PRUEBAS

ESTUDIANTE #2
Carlos Martín de Prado Barragán
(carmarbar9@alum.us.es)

26/05/2024

Grupo : C1.030
Repositorio:
<https://github.com/yesgarfue/Acme-SF.git>

ÍNDICE

1. <i>Historial de versiones</i>	3
2. <i>Introducción</i>	4
3. <i>Contenidos</i>	5
4. <i>Testing Funcional</i>	6
5. <i>Testing de Rendimiento</i>	8
6. <i>Conclusion</i>	10
7. <i>Bibliografía</i>	11

HISTORIAL DE VERSIONES

Versión	Fecha	Descripción
1.0	25/05/2024	Creación documento para entregable DO4
1.1	26/05/2024	Testing realizado

INTRODUCCION

En el siguiente documento se detallan todos los tests realizados por el estudiante dos para completar sus requisitos, a la vez que se analiza el rendimiento de las funcionalidades implementadas durante todos los entregables en la aplicación.

Se pretende realizar un análisis de las pruebas realizadas por el estudiante dos para verificar que la aplicación Acme-SF funciona de manera eficiente y sin errores.

Esto nos permitirá demostrar que Acme-SF se encuentra libre de errores y vulnerabilidades que puedan poner en peligro la integridad del sistema.

CONTENIDOS

DO4 –

- ESTUDIANTE #2 – 09:

“Produce a test suite for Requirements #6 and #7.”

Se nos pide reproducir un conjunto de tests para los requisitos 6 y 7.

Requisito 6:

“Operations by clients on contracts:

- List the contracts that they have created.
- Show the details of their contracts.
- Create, update, or delete their contracts. Contracts can be updated or deleted as long as they have not been published. For a contract to be published, the sum of all budgets cannot exceed the total cost of the corresponding project.”

Requisito 7:

“Operations by clients on progress logs:

- List the progress logs in their contracts.
- Show the details of their progress logs.
- Create and publish a progress log.
- Update or delete a progress log as long as it is not published.”

TESTING FUNCIONAL

Se han ejecutado pruebas positivas y negativas para comprobar que los servicios de listar, mostrar, crear, actualizar, borrar y publicar, funcionan correctamente tanto en el requisito seis como en el siete. Éstas se encontraran en los archivos .safe.

También se han realizado tests de hacking. Estos intentan realizar acciones ilegales en el sistema que solo deberían poder realizar los clientes correctos. Se encuentran en los archivos .hack. Han sido implementados para los servicios de listar, mostrar y crear, pues son los únicos a los que se pueden acceder mediante una url.

Se ha tratado de proporcionar el máximo coverage posible para los servicios mencionados, siendo todos superiores al 90%, excepto el servicio de borrado, pues tenemos implementado el método 'unbind', el cual no hace nada al borrar.

Todas estas pruebas aparecerán en la rama master, aunque donde se garantiza al cien por cien su correcto funcionamiento es en la rama hotfix/carmarbar9.

CONTRACTS

- ClientContractList:
 - Comprueba que el listado muestra los datos esperados.
 - Hacking: Comprueba que usuarios sin loggear o usuarios loggeados con un rol distinto a 'cliente', no pueden acceder al listado de los contratos de un determinado cliente.

- ClientContractShow:
 - Comprueba que se muestran los datos esperados de un contrato.
 - Hacking: Comprueba que usuarios sin loggear, usuarios loggeados con un rol distinto a 'cliente', o clientes distintos, no pueden acceder a los detalles de un contrato de otro cliente.

- ClientContractCreate:
 - Comprueba que se pueda crear correctamente un contrato. Además se intenta crear uno con datos inválidos para que salten restricciones.
 - Hacking: Comprueba que usuarios sin loggear o usuarios loggeados con un rol distinto a 'cliente', no pueden acceder al formulario de creación de un contrato.

- ClientContractUpdate:
 - Comprueba que se pueda actualizar correctamente un contrato. Además se intenta actualizar uno con datos inválidos para que salten restricciones.

- ClientContractDelete:
 - Comprueba que se puede borrar correctamente un contrato.
- ClientContractPublish:
 - Comprueba que se puede publicar correctamente un contrato. Además se intenta publicar uno con datos inválidos para que salten restricciones.

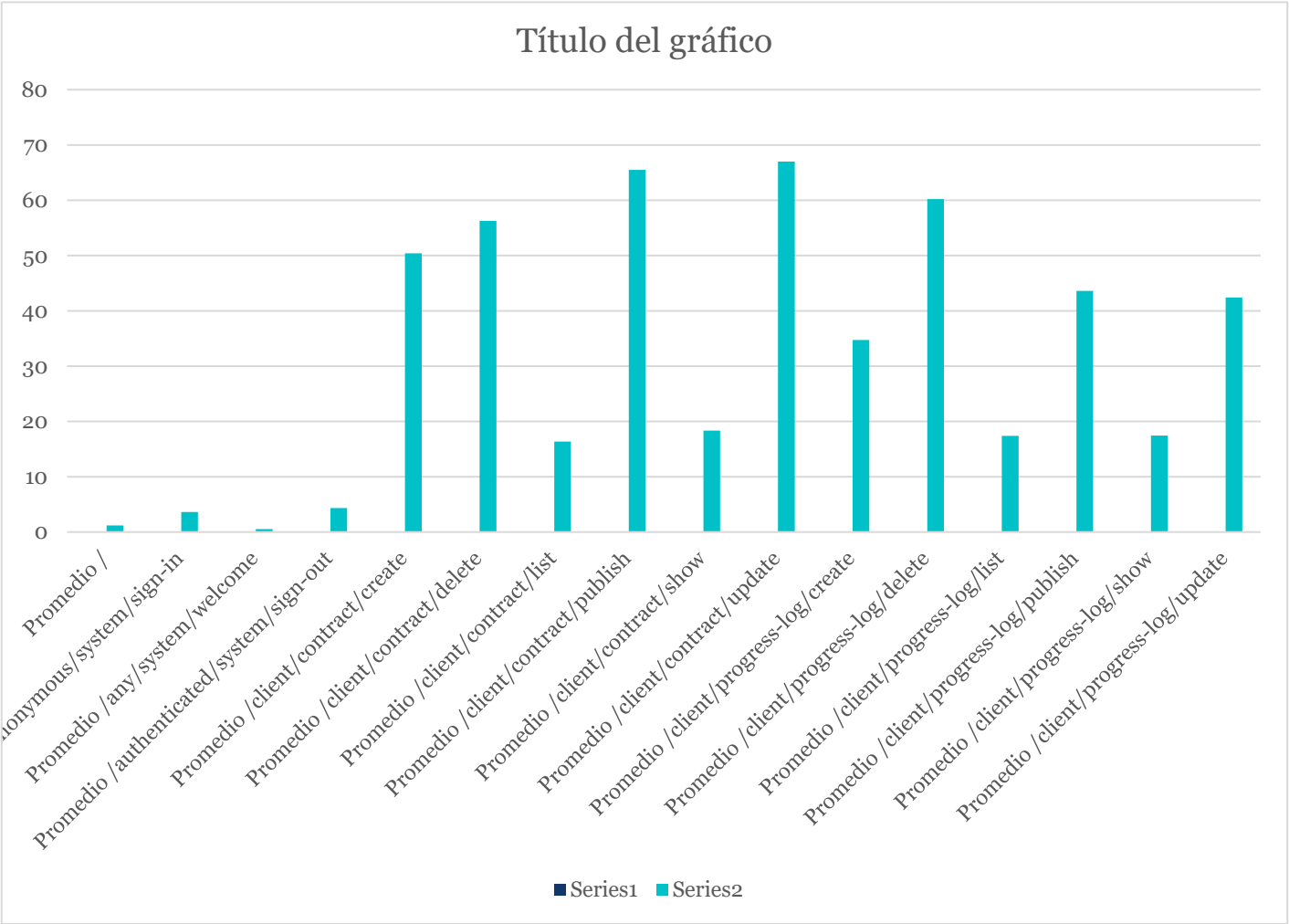
Progress Logs

- ClientProgressLogList:
 - Comprueba que el listado muestra los datos esperados.
 - Hacking: Comprueba que usuarios sin loggear o usuarios loggeados con un rol distinto a 'cliente', no pueden acceder al listado de los registros de progreso de un determinado cliente.
- ClientProgressLogShow:
 - Comprueba que se muestran los datos esperados de un registro de progreso.
 - Hacking: Comprueba que usuarios sin loggear, usuarios loggeados con un rol distinto a 'cliente', o clientes distintos, no pueden acceder a los detalles de un registro de progreso de otro cliente.
- ClientProgressLogCreate:
 - Comprueba que se pueda crear correctamente un registro de progreso. Además se intenta crear uno con datos inválidos para que salten restricciones.
 - Hacking: Comprueba que usuarios sin loggear o usuarios loggeados con un rol distinto a 'cliente', no pueden acceder al formulario de creación de un registro de progreso.
- ClientProgressLogUpdate:
 - Comprueba que se pueda actualizar correctamente un registro de progreso. Además se intenta actualizar uno con datos inválidos para que salten restricciones.
- ClientProgressLogDelete:
 - Comprueba que se puede borrar correctamente un registro de progreso.
- ClientProgressLogPublish:
 - Comprueba que se puede publicar correctamente un registro de progreso. Además se intenta publicar uno con datos inválidos para que salten restricciones.

TESTING DE RENDIMIENTO

A continuación, se muestran las pruebas de rendimiento realizadas en mi PC. Para ello, he seguido las diapositivas proporcionadas en la asignatura, midiendo el tiempo de ejecución de todos los test realizados mediante herramientas de análisis de datos en ficheros csv desde Excel. Se ha utilizado el fichero tester-trace generado en la carpeta logs tras realizar un tester replayer de mis test.

Análisis de datos de rendimiento extraído de los logs:



Interval (ms)	7,86293393	12,6953057
Interval (s)	0,00786293	0,01269531

<i>Análisis de Tiempo</i>	
Media	10,2791198
Error típico	1,22552812
Mediana	0,7558
Moda	#N/D
Desviación estándar	17,6322791
Varianza de la muestra	310,897267
Curtosis	9,92516838
Coeficiente de asimetría	2,86625452
Rango	111,5949
Mínimo	0,2889
Máximo	111,8838
Suma	2127,7778
Cuenta	207
Nivel de confianza(95,0%)	2,41618588

Promedio /		1,19011951
Promedio /anonymous/system/sign-in		3,62061875
Promedio /any/system/welcome		0,52583958
Promedio /authenticated/system/sign-out		4,3497
Promedio /client/contract/create		50,3998333
Promedio /client/contract/delete		56,2611
Promedio /client/contract/list		16,3592385
Promedio /client/contract/publish		65,4944
Promedio /client/contract/show		18,3421857
Promedio /client/contract/update		67,0073
Promedio /client/progress-log/create		34,7339
Promedio /client/progress-log/delete		60,2085
Promedio /client/progress-log/list		17,3911385
Promedio /client/progress-log/publish		43,61265
Promedio /client/progress-log/show		17,4500143
Promedio /client/progress-log/update		42,4127
Promedio general		10,2791198

CONCLUSIONES

Tras realizar un análisis exhaustivo de los servicios implementados en Acme-SF, podemos confirmar que la aplicación se muestra libre de vulnerabilidades y errores que puedan afectar a la experiencia del usuario.

También se ha comprobado que las peticiones se responden de manera rápida y eficiente, cumpliendo así las restricciones impuestas por el cliente.

BIBLIOGRAFÍA

No procede.