

Informe de Análisis

INFORMACIÓN DE GRUPO

Group: C1.043

Repository: <https://github.com/DaniFdezCab/DP2-2324-C1-043.git>

Student #1

UVUS: danfercab

Contact: danfercab@alum.us.es

Student #2

UVUS: alvmarmun1

Contact: alvmarmun1@alum.us.es

Student #3

UVUS: fracapgar1

Contact: fracapgar1@alum.us.es

Student #4

UVUS: alepingar

Contact: alepingar@alum.us.es

Student #5

UVUS: pabberima

Contact: pabberima@alum.us.es

Date: Sevilla Junio 27, 2024

TABLA DE CONTENIDOS

Informe de Análisis	1
INFORMACIÓN DE GRUPO	1
TABLA DE CONTENIDOS	2
RESUMEN EJECUTIVO	3
TABLA DE REVISIONES.....	3
INTRODUCCIÓN.....	3
Pruebas Realizadas.....	3
Pruebas Funcionales	3
Pruebas de Seguridad	4
Cobertura de Código	4
Pruebas de Rendimiento	4
Análisis con VisualVM	4
Monitoreo de Rendimiento	4
CONCLUSION.....	4
BIBLIOGRAFIA.....	4

RESUMEN EJECUTIVO

El presente análisis aborda el proceso de pruebas realizado en el proyecto Acme-Sf-D04. Se han llevado a cabo pruebas exhaustivas para evaluar tanto la funcionalidad como la seguridad del sistema, centradas principalmente en las entidades Contract y ProgressLog.

TABLA DE REVISIONES

Número de Revisión	Fecha	Descripción
1	22/06/2024	Realización de pruebas .hack y .safe
2	24/06/2024	Realización del Coverage
3	25/06/2024	Pruebas de rendimiento
4	26/06/2024	Elaboración del documento final

INTRODUCCIÓN

El presente informe detalla el análisis exhaustivo realizado sobre el proyecto Acme-SF-D04, enfocado en las pruebas de funcionalidad, seguridad y rendimiento del sistema. Este proyecto ha sido sometido a una serie de evaluaciones rigurosas para garantizar tanto la calidad del código implementado como la robustez frente a posibles vulnerabilidades.

Durante el proceso de análisis, se han ejecutado pruebas funcionales detalladas (.safe y .hack) en las entidades clave del sistema, como Contract y ProgressLog. Estas pruebas no solo han evaluado la correcta implementación de las funcionalidades requeridas, sino que también han validado la seguridad del sistema mediante simulaciones de ataques y técnicas de penetración.

Además, se han realizado pruebas de rendimiento para medir la eficiencia del sistema bajo cargas de trabajo simuladas, seguido de la implementación de mejoras basadas en los resultados obtenidos. Este informe proporciona un análisis detallado de cada aspecto evaluado, destacando los logros alcanzados y las áreas de mejora identificadas durante el proceso.

El siguiente documento presenta los resultados obtenidos, las conclusiones derivadas de las pruebas realizadas y las recomendaciones pertinentes para asegurar la optimización continua del sistema Acme-SF-D04.

Pruebas Realizadas

Pruebas Funcionales

Se realizaron pruebas funcionales denominadas .safe y .hack para las entidades Contract y ProgressLog. Estas pruebas incluyeron escenarios como la creación, eliminación, listado, publicación, visualización y actualización de contratos, así como registros de progreso, validando la correcta funcionalidad y la seguridad de estas operaciones.

Pruebas de Seguridad

Se llevaron a cabo intentos de hackeo específicos (Client*.hack) para evaluar la resistencia del sistema a posibles vulnerabilidades. Estos incluyeron técnicas de penetración como inyecciones SQL y manipulación de autorizaciones, asegurando la robustez de las entidades mencionadas.

Cobertura de Código

Las pruebas realizadas demostraron una cobertura del 90% de los servicios implementados, con excepción de operaciones de eliminación en las que no se utilizó adecuadamente el método unbind, según lo observado en las trazas grabadas.

Pruebas de Rendimiento

Se realizaron pruebas de rendimiento utilizando herramientas de análisis estadístico, revelando mejoras significativas después de implementar índices en las entidades evaluadas. Un Z-Test arrojó un valor de 0,0299, indicando una mejora notable en el rendimiento del sistema tras estas optimizaciones.

Análisis con VisualVM

El análisis con VisualVM identificó que los métodos bind() son los más exigentes en términos de tiempo de ejecución, destacando el create() con un tiempo de 512ms. El tiempo de ejecución prolongado se debe principalmente a los métodos invocados dentro de bind(), con un Self Time de cero milisegundos.

Monitoreo de Rendimiento

El monitoreo de rendimiento mostró un uso moderado de la red, indicando que no es un cuello de botella significativo. La CPU y la memoria del sistema mostraron subutilización, lo que sugiere capacidad adicional para manejar cargas de trabajo más intensas.

CONCLUSION

En conclusión, el proceso de pruebas del proyecto Acme-Sf-D04 ha sido exitoso, cumpliendo con los estándares de calidad y seguridad establecidos. Las pruebas exhaustivas realizadas garantizan un sistema robusto y seguro, preparado para enfrentar desafíos de funcionalidad y seguridad.

BIBLIOGRAFIA

Intencionalmente en blanco.