

Universidad de Guadalajara

CENTRO UNIVERSITARIO DE CIENCIAS EXACTAS E INGENIERÍAS.



PROGRAMACIÓN TOLERANTE A FALLAS

I7036

TEMA: HERRAMIENTAS DE MANEJO DE ERRORES

AUTOR: SEBASTIAN A. BAEZ RAMOS

PROFESOR: MICHEL EMANUEL LOPEZ FRANCO

GUADALAJARA, JALISCO A 29 DE AGOSTO DEL 2023

HERRAMIENTAS PARA EL MANEJO DE ERRORES.

En el mundo del desarrollo de software, los errores pueden surgir debido a diversas razones, como la complejidad del código, la interacción de múltiples componentes, la variabilidad de las plataformas de ejecución y los cambios constantes en los requisitos. Sin embargo, lo que realmente marca la diferencia en la calidad y la confiabilidad del software es cómo se gestionan estos errores cuando surgen.

El manejo de errores se ha convertido en una piedra angular en la creación de aplicaciones y sistemas robustos. De esta forma, las herramientas diseñadas para detectar errores desempeñan un papel crítico en todo el ciclo de vida del desarrollo de software. Desde la etapa de diseño y codificación hasta las pruebas y la implementación, y más allá en el mantenimiento y actualización continua, estas herramientas actúan como guardianes que aseguran la integridad y la estabilidad del software.

En el presente informe, exploraremos a fondo la trascendental importancia de las herramientas de manejo de errores en el ámbito del desarrollo de software y daremos una visión alternativa al uso de los típicos *try-catch* o *try-except*. Analizaremos cómo estas herramientas permiten a los equipos de desarrollo identificar y abordar problemas de manera más eficiente, reducir costos, mejorar la experiencia del usuario y preservar la reputación de la organización.

SONARQUBE.

La calidad de software es un elemento fundamental en la industria del desarrollo. Disponer de herramientas que nos ayuden a evaluar nuestro código se convierte en una tarea imprescindible para garantizar el desarrollo de una manera correcta y la aplicación de buenas prácticas. SonarQube es una plataforma muy popular que permite a todos los desarrolladores escribir código más limpio y seguro [1]. Teniendo en cuenta las buenas prácticas de desarrollo de software, tales como las que predica [Robert C. Martin](#), es un buen punto de partida para realizar un sistema tolerante a fallas, teniendo en cuenta que en gran medida los errores son implementados por el código que escribe el programador.

Las principales funcionalidades de SonarQube incluyen:

1. **Análisis Estático de Código:** SonarQube realiza un análisis exhaustivo del código fuente en busca de vulnerabilidades de seguridad, malas prácticas de codificación, duplicación de código y otros problemas relacionados con la calidad.
2. **Medición de la Deuda Técnica:** La herramienta proporciona métricas que cuantifican la deuda técnica, lo que permite a los equipos priorizar y abordar los aspectos del código que requieren mejoras.
3. **Integración Continua:** SonarQube puede integrarse con herramientas de integración continua, lo que permite realizar análisis automáticos cada vez que se realiza una nueva confirmación de código.

4. **Detección de Vulnerabilidades y Bugs:** La plataforma identifica vulnerabilidades conocidas y posibles bugs en el código, lo que ayuda a prevenir problemas de seguridad y errores en etapas tempranas.

Importancia de SonarQube

La importancia de SonarQube en el desarrollo de software es evidente por varias razones:

1. **Mejora de la Calidad del Código:** SonarQube permite a los equipos identificar y corregir problemas en el código antes de que lleguen a producción, lo que resulta en aplicaciones más estables y confiables.
2. **Reducción de Costos y Tiempo:** Al abordar problemas en etapas tempranas, se evita la acumulación de deuda técnica, lo que ahorra tiempo y recursos que de otro modo se invertirían en la corrección de problemas más adelante.
3. **Promoción de Buenas Prácticas:** La herramienta fomenta la adopción de buenas prácticas de codificación al señalar violaciones y proporcionar recomendaciones para mejorar el código.
4. **Mejora de la Seguridad:** SonarQube identifica vulnerabilidades y problemas de seguridad en el código, lo que contribuye a la creación de aplicaciones más seguras y resistentes a ataques.
5. **Facilita la Colaboración:** Al brindar información clara sobre la calidad del código, SonarQube promueve la colaboración entre los miembros del equipo, ya que todos pueden trabajar hacia un objetivo común de mejora continua.

JMETER.

Apache JMeter es una herramienta de código abierto diseñada para realizar pruebas de carga, rendimiento y estrés en aplicaciones web y servicios. A través de su interfaz gráfica intuitiva y su capacidad para simular cargas de trabajo realistas, JMeter permite a los equipos de desarrollo evaluar el comportamiento y el rendimiento de las aplicaciones bajo diversas condiciones [2].

Uso de Apache JMeter en el Desarrollo de Software

Las principales funciones y usos de Apache JMeter incluyen:

1. **Pruebas de Carga y Rendimiento:** JMeter permite a los desarrolladores emular múltiples usuarios concurrentes interactuando con una aplicación, lo que ayuda a identificar cuellos de botella y problemas de rendimiento bajo cargas pesadas.

2. **Escenarios de Estrés:** La herramienta puede someter las aplicaciones a condiciones extremas para evaluar cómo responden en situaciones de alta demanda o picos de tráfico.
3. **Pruebas de Estabilidad:** JMeter realiza pruebas a largo plazo para verificar la estabilidad de una aplicación, identificando posibles fugas de memoria o problemas de rendimiento a lo largo del tiempo.
4. **Monitoreo de Servicios Web:** Además de realizar pruebas, JMeter puede monitorear servicios web para medir la latencia, el tiempo de respuesta y otros indicadores de rendimiento en tiempo real.

Importancia de Apache JMeter en el Desarrollo de Software

1. **Garantía de Rendimiento:** JMeter permite identificar problemas de rendimiento y cuellos de botella antes de que los usuarios finales los experimenten, garantizando una experiencia de usuario satisfactoria.
2. **Optimización de Recursos:** Al identificar los recursos que una aplicación consume bajo diferentes cargas, JMeter ayuda a optimizar el uso de recursos y a prevenir problemas de escalabilidad.
3. **Reducción de Riesgos:** Al simular diversas situaciones de carga, JMeter ayuda a prever y abordar problemas que podrían surgir en condiciones del mundo real, minimizando el riesgo de fallos inesperados.
4. **Ahorro de Costos:** Al detectar problemas de rendimiento antes del despliegue, JMeter evita costosos rechazos y retrabajos posteriores al lanzamiento.
5. **Mejora Continua:** JMeter fomenta un enfoque de mejora continua al proporcionar información valiosa sobre el rendimiento de la aplicación en diferentes etapas de su ciclo de vida.

BIBLIOGRAFÍA.

[1] <https://sentry.io/blog/que-es-sonarqube/>

[2] <https://jmeter.apache.org/>