



Universidad de Guadalajara
Centro Universitario
de
Ciencias Exactas
e
Ingenierías
(CUCEI)

Materia: COMPUTACION TOLERANTE A FALLAS

Nombre:
Jonathan Aceves López

Código:
217481363
NRC: 179961

29/08/2023

(Par. 1) Otras herramientas para el manejar errores, genera un reporte con otras herramientas para el manejo de errores en programación.

El manejo de errores es una parte fundamental en el desarrollo de software para garantizar la robustez y confiabilidad de las aplicaciones. Aquí te presento una lista de herramientas y enfoques para manejar errores tanto desde el código como utilizando herramientas externas:

Manejo de Errores en el Código:

Excepciones y Try-Catch Blocks: En lenguajes que admiten excepciones, como Java, Python o C#, puedes utilizar bloques try-catch para capturar y manejar excepciones que puedan ocurrir durante la ejecución.

Validación de Entradas: Asegúrate de validar y verificar las entradas del usuario o de fuentes externas para prevenir errores antes de que ocurran.

Logs Detallados: Implementa un sistema de registro (logging) en tu aplicación para registrar información detallada sobre los errores, incluyendo mensajes de error, trazas de pila y valores de variables relevantes.

Retornos de Funciones Significativos: En lugar de usar valores mágicos o nulos para indicar errores, utiliza códigos de retorno o tipos especiales para indicar el resultado de una función y posibles errores.

Manejo de Errores Gradual (Graceful Degradation): En aplicaciones web, asegúrate de que la aplicación pueda funcionar de manera parcial o con funciones limitadas en caso de errores, en lugar de fallar por completo.

Herramientas Externas:

Linters: Utiliza herramientas como ESLint (JavaScript/TypeScript), pylint (Python), o Checkstyle (Java) para analizar el código en busca de problemas de estilo y posibles errores.

Depuradores (Debuggers): Utiliza depuradores integrados en entornos de desarrollo para rastrear y resolver errores paso a paso, inspeccionando variables y ejecución de código.

Sistemas de Monitoreo de Aplicaciones (Application Monitoring): Herramientas como New Relic, AppDynamics o Prometheus ayudan a monitorear el rendimiento y los errores en tiempo real en aplicaciones en producción.

Manejo de Errores en Tiempo Real: Plataformas como Sentry y Rollbar permiten capturar y registrar errores en tiempo real, proporcionando información detallada para identificar y solucionar problemas.

Pruebas Unitarias y de Integración: Utiliza frameworks de pruebas como JUnit (Java), pytest (Python) o Jest (JavaScript) para escribir pruebas que verifiquen el comportamiento esperado y el manejo de errores.

Recolección de Basura (Garbage Collection) y Análisis Estático: Utiliza herramientas de recolección de basura para liberar recursos no utilizados y herramientas de análisis estático como FindBugs (Java) o Pylint (Python) para encontrar problemas potenciales antes de la ejecución.

Control de Versiones y Ramas de Desarrollo: Utiliza sistemas de control de versiones como Git y prácticas de ramificación para aislar y corregir errores sin afectar el código principal.

Gewarren. (2023, September 5). Procedimiento Uso del bloque Try/Catch para detectar excepciones. Microsoft Learn. <https://learn.microsoft.com/es-es/dotnet/standard/exceptions/how-to-use-the-try-catch-block-to-catch-exceptions>

Equipo editorial de IONOS. (2020). Depuradores: importantes herramientas para detectar errores de software. IONOS Digital Guide. <https://www.ionos.mx/digitalguide/paginas-web/desarrollo-web/depurador/>

Equipo editorial de IONOS. (2020). Depuradores: importantes herramientas para detectar errores de software. IONOS Digital Guide. <https://www.ionos.mx/digitalguide/paginas-web/desarrollo-web/depurador/>

Equipo editorial de IONOS. (2020). ¿Qué es un sistema operativo? IONOS Digital Guide. <https://www.ionos.mx/digitalguide/servidores/know-how/el-sistema-operativo/>

Philavanh, V. (2022). Depuradores: Conceptos básicos. Arrow.com. <https://www.arrow.com/es-mx/research-and-events/articles/debuggers-the-basics>