

Universidad de Guadalajara

Centro Universitario de Ciencias Exactas e Ingenierías

Computación Tolerante a Fallas



Maestro: Michel Emanuel Lopez Franco

Juan Antonio Perez Juarez

Carrera: INCO

Código: 215660996

Herramientas para manejar errores

El seguimiento de errores es el proceso de registro y monitoreo de errores durante las pruebas de software. También se conoce como seguimiento de defectos o seguimiento de problemas.

Los sistemas grandes podrían tener cientos o miles de defectos. Cada uno debe evaluarse, monitorearse y priorizarse para su depuración. A veces, es posible que sea necesario realizar un seguimiento de los errores durante un largo período.

"El seguimiento de defectos es un proceso importante en la ingeniería de software, ya que los sistemas complejos y críticos para el negocio tienen cientos de defectos", afirma Tutorials Point. "Uno de los factores desafiantes es gestionar, evaluar y priorizar estos defectos. El número de defectos se multiplica a lo largo del tiempo y, para gestionarlos de manera eficaz, se utiliza un sistema de seguimiento de defectos para facilitar el trabajo".

Un error de software ocurre cuando una aplicación o programa no funciona como debería. La mayoría de los errores son fallas o errores cometidos por arquitectos, diseñadores o desarrolladores de sistemas. Los equipos de pruebas utilizan el seguimiento de errores para monitorear e informar sobre los errores que ocurren a medida que se desarrolla y prueba una aplicación.

"Un componente importante de un sistema de seguimiento de errores es una base de datos que registra datos sobre errores conocidos", dice Wikipedia. "Los hechos pueden incluir el momento en que se reportó un error, su gravedad, el comportamiento erróneo del programa y detalles sobre cómo reproducir el error; así como la identidad de la persona que lo reportó y cualquier programador que pueda estar reparándolo".

Los errores se gestionan en función de la prioridad y la gravedad. Los niveles de gravedad ayudan a identificar el impacto relativo de un problema en la versión de un producto. Estas clasificaciones pueden variar en número, pero generalmente incluyen alguna forma de lo siguiente:

Catastrófico: provoca una falla total del software o pérdida irrecuperable de datos. No hay solución y el producto no se puede lanzar.

Funcionalidad deteriorada: puede existir una solución alternativa, pero no es satisfactoria. El software no se puede liberar.

Falla de sistemas no críticos: existe una solución alternativa razonablemente satisfactoria. El producto puede lanzarse si se documenta el error.

Menor: hay una solución alternativa o se puede ignorar el problema. No afecta al lanzamiento de un producto.

Por lo general, los estados y los niveles de gravedad se monitorean en una base de datos de seguimiento de errores. Una buena plataforma de seguimiento también se ata a sistemas de desarrollo y administración de software más grandes: el mejor para evaluar el estado de error y el impacto potencial en la producción y los plazos generales.

Se estima que los desarrolladores de software cometen entre 100 y 150 errores por cada mil líneas de código.⁴ Según un informe del Consortium for IT Software Quality (CISQ): "Incluso si solo una pequeña fracción (digamos el 10 %) de estos errores son graves, una aplicación relativamente pequeña de 20 000 líneas de código tendrá aproximadamente 200 errores de codificación graves".⁵

Las pruebas de software son esenciales para aislar y mitigar errores. Un buen proceso de control de calidad puede descubrir cientos o incluso miles de defectos, y los equipos de pruebas tienen que gestionarlos todos. Integrar el seguimiento de errores en el flujo de trabajo de las pruebas mejora la eficacia al ayudar a los probadores a priorizar, supervisar e informar sobre el estado de cada error.

"El seguimiento de defectos ayuda a garantizar que los errores encontrados en el sistema realmente se solucionen", comenta Yvette Francino, consultora de Agile. "Las herramientas de seguimiento no solo proporcionan una forma de garantizar el seguimiento, sino que también proporcionan métricas valiosas. Dependiendo de la herramienta que se utilice, el equipo puede vincular los defectos a códigos modificados, pruebas u otros datos que permitirán la trazabilidad o el análisis de las tendencias de defectos. Si un determinado módulo está plagado de defectos, puede ser el momento de revisar y reescribir ese módulo".⁶

Lo ideal es que las pruebas se realicen lo antes posible, cuando es más fácil y mucho menos costoso corregir los errores. Un estudio anterior de IBM reveló que los defectos detectados después de la producción o el lanzamiento pueden costar 15 veces más que los errores resueltos en las primeras fases del desarrollo.

Muchos equipos utilizan ahora una metodología conocida como pruebas continuas. En este caso, se realizan pruebas de calidad y comentarios en todas las etapas del desarrollo, desde el diseño y la codificación hasta la implementación. Las tecnologías modernas como la inteligencia artificial (IA) también pueden ayudar en el proceso de prueba al detectar y analizar errores al principio del ciclo de vida.

Existen varias plataformas que te ayudan con el seguimiento de errores dentro de metodologías de desarrollo así como dentro de IDEs de programación y desarrollo, Aquí tenemos el ejemplo de algunas.

FogBugz



Una reconocida herramienta de seguimiento de errores, está diseñada para agilizar la gestión de proyectos de desarrollo de software. Ofrece funciones versátiles para el seguimiento de errores, la gestión de incidencias y la coordinación de proyectos. Con una interfaz intuitiva, FogBugz permite a los equipos identificar, registrar y priorizar errores de manera eficiente.

Específicamente para los profesionales de control de calidad, FogBugz proporciona funcionalidades para crear y gestionar casos de prueba, rastrear defectos y generar informes completos. La integración con otras herramientas de desarrollo y la compatibilidad con flujos de trabajo personalizados mejoran aún más su aplicabilidad en diversos escenarios de pruebas.

FogBugz ofrece opciones en la nube y en las instalaciones, que se adaptan a las distintas necesidades de las organizaciones. La estructura de precios está diseñada para adaptarse a equipos de distintos tamaños y a limitaciones presupuestarias, con opciones adaptadas a las empresas. Su atención al cliente y los recursos de la comunidad hacen que sea una opción popular entre los equipos de desarrollo y control de calidad por igual.

Jira



Desarrollada por Atlassian, es una herramienta líder de gestión de proyectos y seguimiento de incidencias que se ha convertido en una parte esencial de muchos ciclos de vida de desarrollo de software (SDLC). Aunque es conocida por sus sólidas funciones de planificación de proyectos, colaboración y elaboración de informes, también ofrece funciones especializadas de control de calidad y seguimiento de errores. Estas funcionalidades de seguimiento de errores se integran a la perfección en el contexto más amplio del desarrollo, por lo que es una opción versátil para los equipos que buscan un enfoque racionalizado y completo de sus proyectos.

Jira destaca en la gestión estructurada de casos de prueba, integrándose fácilmente con herramientas como Zephyr para una ejecución sin problemas.

También ofrece un sólido seguimiento de defectos con campos personalizables, lo que le permite registrar detalles y adjuntar problemas relacionados.

La colaboración en equipo es fluida, gracias a funciones como los comentarios en hilos y los paneles compartidos. Usted no está confinado a flujos de trabajo rígidos; Jira permite la personalización para adaptarse a sus procesos de pruebas.

Además, la plataforma se integra sin esfuerzo con herramientas CI/CD y sistemas de control de versiones. Además, sus herramientas de generación de informes ofrecen información vital sobre las tendencias de los defectos y la salud del proyecto.

La adaptabilidad de Jira hace que sea muy versátil para diversas metodologías de pruebas, atendiendo a las diferentes necesidades del proyecto. Su disponibilidad en ediciones de nube, servidor y centro de datos permite flexibilidad en los precios. Los equipos pequeños de 1 a 10 usuarios pueden utilizar Jira de forma gratuita, mientras que los equipos más grandes y las empresas pueden elegir entre varios niveles de pago en función de las características requeridas.

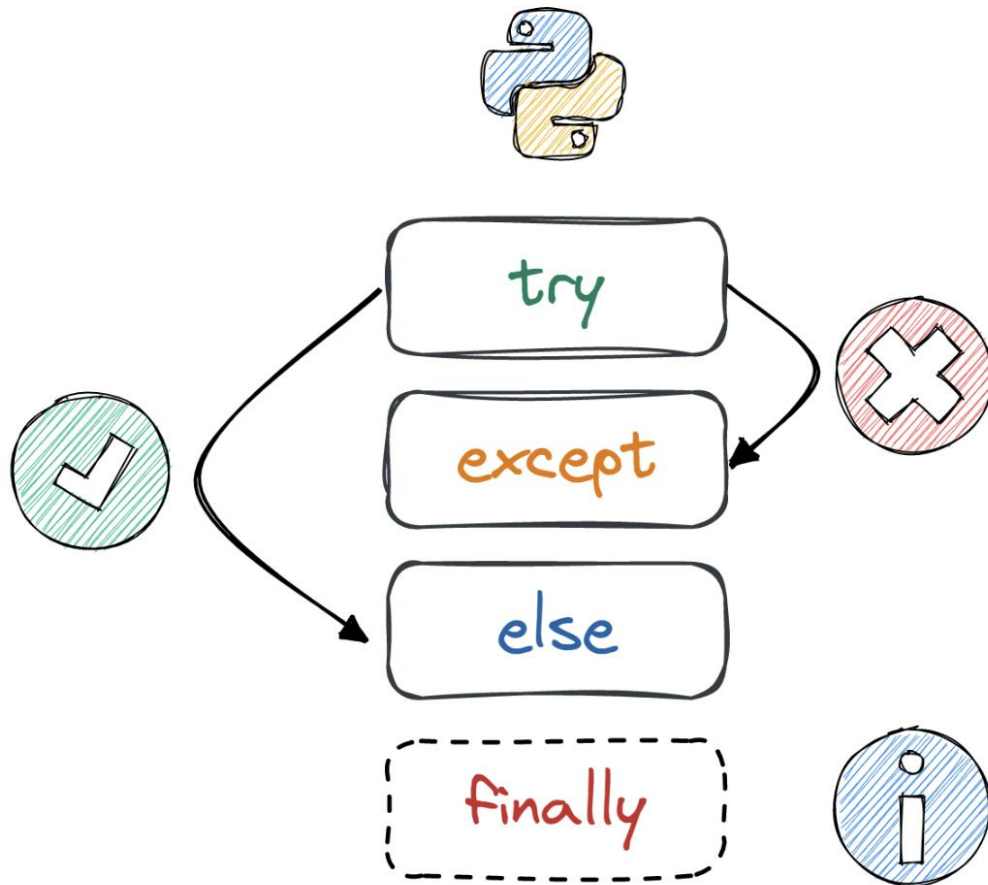
Los casos de uso de Jira abarcan desde pruebas funcionales hasta pruebas de aplicaciones web y comercio electrónico, lo que proporciona una solución completa y sólida para los equipos de desarrollo modernos.

Pero si nos enfocamos en el manejo de errores por parte de un lenguaje de programación, el lenguaje python es el que tiene una manera de manejar los errores con un comando.

Los errores y las excepciones pueden provocar un comportamiento inesperado o incluso detener la ejecución de un programa. Python proporciona varias funciones y mecanismos para manejar estos problemas y mejorar la robustez del código.

Un error es un problema en un programa que impide que éste complete su tarea. En comparación, una excepción es una condición que interrumpe el flujo normal del programa. Tanto los errores como las excepciones son un tipo de error en tiempo de ejecución, lo que significa que se producen durante la ejecución de un programa.

En palabras sencillas, el error es un problema crítico que una aplicación normal no debería detectar, mientras que una excepción es una condición que un programa debería detectar.



Manejo de excepciones con try, except, else y finally

Después de aprender sobre errores y excepciones, aprenderemos a manejarlos utilizando los bloques try, except, else y finally.

Entonces, ¿qué entendemos por manejarlos? En circunstancias normales, estos errores detendrán la ejecución del código y mostrarán el mensaje de error. Para crear sistemas estables, debemos anticiparnos a estos errores e idear soluciones alternativas o mensajes de advertencia.

En esta sección, aprenderemos qué hace cada bloque y cómo podemos utilizarlos para escribir código robusto.

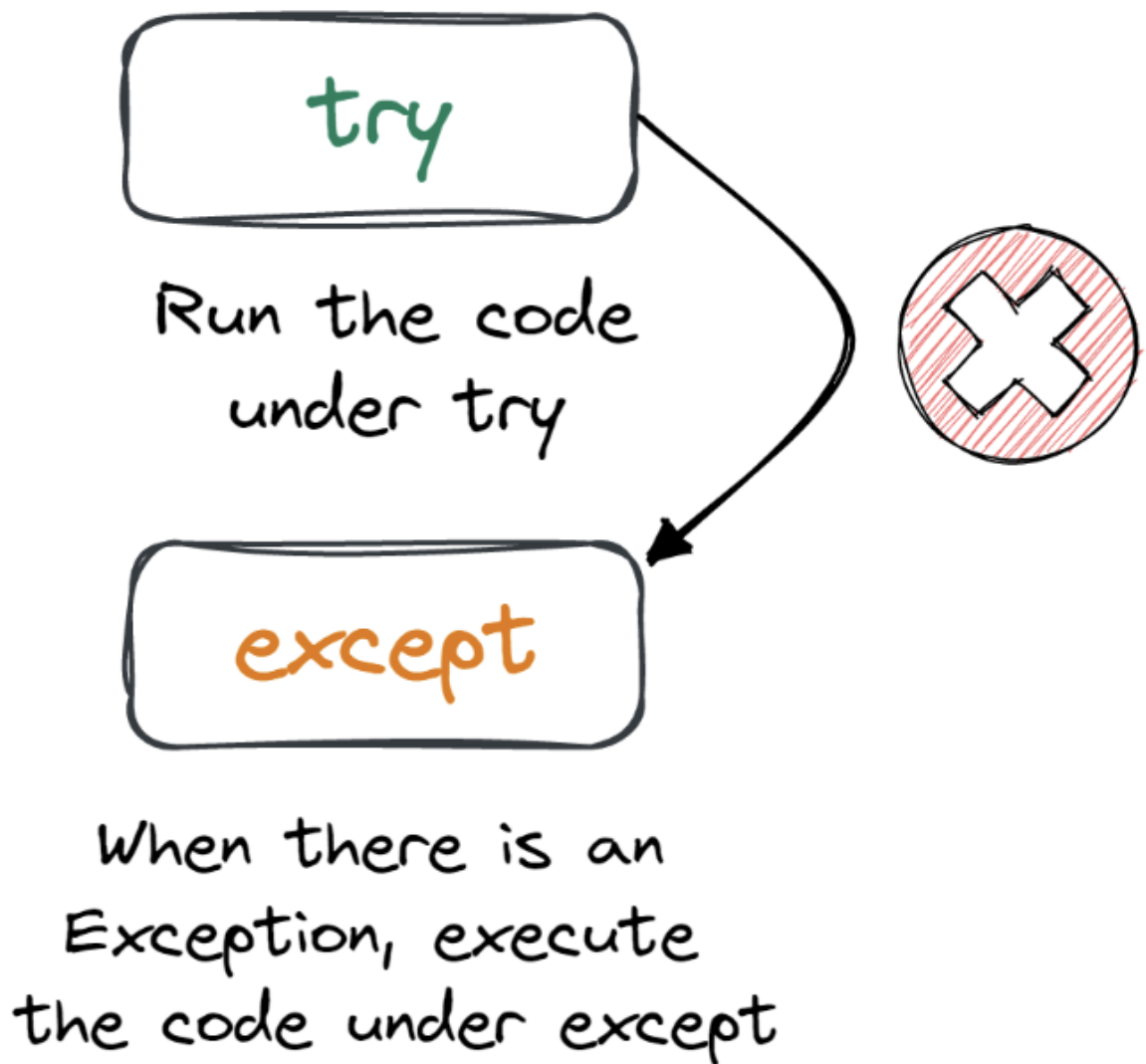
Declaración try y except

La forma más sencilla de manejar excepciones en Python es usando los bloques ``try`` y ``except``.

Ejecute el código bajo la sentencia ``try``.

Cuando se lanza una excepción, ejecuta el código bajo la sentencia ``except``.

En lugar de detenerse ante un error o una excepción, nuestro código avanzará hacia soluciones alternativas.



Ejemplo sencillo

En el primer ejemplo, intentaremos imprimir la variable indefinida `x`. En circunstancias normales, debería lanzar el error y detener la ejecución, pero con el bloque `try` y `except`, podemos cambiar el comportamiento del flujo.

El programa ejecutará el código bajo la sentencia `try`.

Como sabemos que `x` no está definida, ejecutará la sentencia `except` e imprimirá la advertencia.

```
try:
    print(x)
```

```
except:
    print("An exception has occurred!")

```

Referencias:

¿Qué es el seguimiento de errores? | IBM. (s. f.). <https://www.ibm.com/mx-es/topics/bug-tracking>

QAwerk ES. (2024, 17 abril). 11 mejores herramientas y software de seguimiento de errores en 2024 | QAwerk. QAwerk. <https://qawerk.es/blog/mejores-herramientas-de-seguimiento-de-errores/>

Manejo de excepciones en Python. (s. f.). Data & AI. Recuperado 23 de agosto de 2024, de <https://www.datacamp.com/es/tutorial/exception-handling-python>