



Nombre: Miguel Alejandro Lomeli Haro

Materia: Computación tolerante a fallas

Actividad: Otras herramientas para el manejar errores

Introducción:

Con la realización de esta práctica buscaremos otras herramientas para el manejo de errores, esto con el fin de conocer más alternativas aparte de la herramienta try-catch, para tener un programa el cual si hay un error no necesariamente tiene que fallar, si no que podemos tener condiciones para que en caso de errores el programa continúe ejecutándose.

Contenido:

El seguimiento de errores, también conocido como seguimiento de defectos o seguimiento de problemas, es el proceso de registro y seguimiento de errores o errores durante pruebas de software. Los grandes sistemas o aplicaciones web pueden tener decenas o cientos de errores. Cada defecto/error debe ser monitoreado, evaluado y priorizado para su depuración.

Un error típico puede pasar por las siguientes etapas:

- Error activo. Una investigación está en curso.
- Error probado. El error ya está solucionado y está listo para la prueba.
- Error verificado. El departamento de control de calidad volvió a probar y verificar el error.
- error cerrado. El departamento de control de calidad volvió a probar el error después de corregirlo o descubrir que no era un error.
- Reabierto. Desafortunadamente, un error puede pasar por las etapas anteriores y aún así no se soluciona. Tal error se puede reabrir.

Herramientas:

- monday.com: La plataforma es adaptable y fue creada pensando en la comodidad del usuario. Permite el registro de errores, la priorización y el seguimiento del progreso sin esfuerzo hasta la resolución.
Los desarrolladores pueden diseñar hojas de ruta personalizadas y diagramas de Gantt para visualizar la planificación de sprints y el seguimiento de errores, brindándoles una descripción completa de toda la información esencial.
- Marker.io: Es una herramienta visual de notificación de errores para agencias y equipos de desarrollo de software. La herramienta tiene un widget de sitio web que permite a los desarrolladores recopilar comentarios a través de metadatos técnicos, capturas de pantalla y anotaciones.

Excepciones y errores:

- try: Primero, se ejecuta la cláusula try, Si no ocurre ninguna excepción, la cláusula except se omite y la ejecución de la cláusula try finaliza, Si ocurre una excepción durante la ejecución de la cláusula try, se omite el resto de la cláusula. Luego, si su tipo coincide con la excepción nombrada después de la palabra clave except, se ejecuta la cláusula except, y luego la ejecución continúa después del bloque try/except. Si ocurre una excepción que no coincide con la indicada en la cláusula except se pasa a los try más externos.
- except: Cuando ocurre una excepción, puede tener un valor asociado, también conocido como el argumento de la excepción. La presencia y el tipo de argumento depende del tipo de excepción.
- BaseException: es la clase base común de todas las excepciones. Una de sus subclases, Exception, es la clase base de todas las excepciones no fatales. Las excepciones que no

son subclases de Exception no se suelen manejar, porque se utilizan para indicar que el programa debe terminar. Entre ellas se incluyen SystemExit, que es lanzada por sys.exit() y KeyboardInterrupt, que se lanza cuando un usuario desea interrumpir el programa.

- Exception se puede utilizar como un comodín que atrapa (casi) todo. Sin embargo, es una buena práctica ser lo más específico posible con los tipos de excepciones que pretendemos manejar, y permitir que cualquier excepción inesperada se propague.
- raise: permite al programador forzar a que ocurra una excepción específica.

Conclusiones:

Con la realización de esta actividad, aprendí otras herramientas para que tu programa al momento de ser ejecutado y tener un error no finalice, si no que siga funcionando mediante condiciones.

De esta manera podemos tener una mejor optimización a la hora de programar, ya que si surge algún error y lo podemos preveer, el código no terminaría a la hora de detectar un error.

Bibliografía:

- Kamunya, T. (2023). Las 11 mejores herramientas de seguimiento de errores para equipos de desarrollo modernos. *Geekflare*. <https://geekflare.com/es/bug-tracking-tools/>
- 8. Errores y excepciones. (s. f.). Python documentation. <https://docs.python.org/es/3/tutorial/errors.html>