



# Universidad de Guadalajara

Centro Universitario de  
Ciencias Exactas e  
ingenierías

División de Tecnologías para la  
Integración Ciber-Humana

Computación Tolerante a fallas



## Actividad 3: Otras herramientas para el manejo de errores

Profesor: Michel Emanuel López Franco

Elaborado por: Sanchez Fregoso Jose Manuel

Código: 215476966

Carrera: ingeniería en computación

## Contenido

Introducción.....	3
Desarrollo .....	4
Conclusión.....	6
Referencias .....	7

## Introducción

Los errores, fallas, etc. es un concepto el cual está gravado en la existencia propia del ser humano, ya que prácticamente el día a día de cierta forma se resuelve algún tipo de error en cierto nivel. Por lo cual es un concepto bastante amplio y aplicable en diferentes áreas.

Sin embargo, no nosotros nos enfocaremos particularmente en el área de desarrollo de software donde es muy común el encontrar situaciones las cuales son ocasionadas por algún tipo de error.

En el desarrollo de software existen múltiples técnicas y soluciones para trabajar con los posibles errores los cuales se pueden llegar a presentar, ya sea desde problemas lógicos (errores de código, fallo humano, etc.), hasta errores físicos (algún componente tangible falta o falla). Sin embargo, la existencia de dichos errores no significa que no podamos encontrar la forma de manejarlos y trabajar con ellos para poder obtener un resultado esperando (tal vez no sea el caso ideal u óptimo, pero tampoco será algo de lo cual no sabemos que es lo que sucede).

Para poder lograr esto existen técnicas, herramientas, softwares, etc. los cuales nos permiten manejar, controlar, identificar, registrar, entre otras cuestiones estos errores, para lograr nuestro cometido, el cual es minimizar el impacto que puede llegar a tener nuestro desarrollo si dicho error sucede. A continuación, hablaremos de las técnicas y herramientas que nos puedan ayudar a manejar los errores.

# Desarrollo

## Herramientas y técnicas para el manejo de errores

Dentro de las opciones que tenemos para el manejo de los errores tenemos 2 tipos, el primero al cual nos referimos son las herramientas técnicas las cuales nos podemos auxiliar, las cuales pueden ser tanto internas como externas, a que nos referimos con esto, es decir que pueden estar integradas en el software que estamos usando, o que se encuentren por fuera del software en uso; La segunda opción que tenemos son técnicas que se han adoptado la cuales nos permiten el manejar los errores de una forma mas practica a la hora de desarrollar software.

### *Herramientas para el manejo de errores*

Dentro de las herramientas que contamos para el manejo de los errores tenemos:

- Las integradas directamente en el software: Estas son utilerías o forman parte de la estructura propia del software, las cuales están diseñadas específicamente para ese propósito en específico el cual es el manejo de los distintos errores propios del software, como lo son los manejadores de excepciones, los cuales como lo indica su nombre tiene como propósito el tomar dichas excepciones las cuales generarían un comportamiento irregular y nos permiten el tomar un curso de acción en base a eso.
  - Por lo general la mayoría de los lenguajes de programación cuentan con dicha característica, lo más común con las Excepciones y los bloques “Try, Catch y Finally”. Las excepciones nos permiten el elevar un error que nosotros deseamos si alguna condición no se cumplió, lo cual cambia el ciclo de ejecución como se tenía pensado, dichas excepciones pueden ser personalizables a algo o ser predefinidas por el software; Los bloques “Try, Catch, Finally” sirven para que si llega a suceder la excepción nosotros la podamos atrapar y consecutivamente hacer algo pertinente si se llega a presentar la situación.
  - Los debuggers, son herramientas de software el cual permite examinar y comprender el funcionamiento de un programa durante el tiempo de ejecución. Lo cual nos facilita la identificación y correcciones de errores por medio de la información proporcionada como lo es el valor en la variable, el flujo de ejecución, establecer puntos de interrupción en el código, etc. Facilitando enormemente el error en la aplicación.

### **Técnicas para el manejo de errores**

Independiente de las herramientas que nos ayudan al manejo de los errores, con los años se han ido desarrollando técnicas que nos ayudan a reducir estos errores, como lo son:

- Validación de datos: Simplemente consiste en prever un posible error que se podría ocasionar y validar que los datos cumplan con el mínimo requerido

para que la aplicación funcione, es decir revisar si el valor es tipo específico, o cumple con un criterio específico, lo cual se llega a manejar con bloques “If-Else” e “iteradores”, con lo cual o podemos lanzar una excepción o un código de error.

- Códigos de error: Esta técnica consiste en ir de la par con la validación de datos, si alguna condición no se cumplió, simplemente retornar un tipo de codificación la cual nos indique que sucedió dicha situación, por convención adoptada por la comunidad se suele usar el “-1” como indicador de un error al retorno de una función.
- Loggeo: Esta técnica consiste en implementar un proceso el cual este registrando la actividad generada en la aplicación, por lo cual, si se llega a registrar un error, no solamente esconderlo de forma pasiva, si no registrar el error, se puede acompañar con los datos los cuales se registro dicho error para poder investigar más a fondo que sucedió. Obviamente para contar con un sistema de loggeo, es necesario manejar excepciones de forma adecuada para evitar la interrupción de la aplicación (“Preferentemente”).
- Pruebas unitarias y de integración: Esta es una técnica la cual mezcla ambos conceptos ya que se puede dar el caso que para probar un segmento del software o instalemos alguna extensión o utilidad, o simplemente realizarlo de forma que llamamos al método con variedad de opciones y observar que el resultado sea el esperado. Recomendaría más la primera opción (usar algo externo), ya que suelen ser “Herramientas” las cuales están diseñadas completamente con ese propósito y nos otorgan la utilidad de verificar los datos, para evitar tener que ver que en realidad está funcionando algo como se espera. En si la función de las pruebas unitarias el probar un segmento de código/función y verificar si los resultados obtenidos son los esperados, dichas pruebas están diseñada para corroborar el funcionamiento esperado.
- Code review: Claro está que no está de más que un compañero o colega, te ayude a verificar el funcionamiento adecuado de tu aplicación o implementación, o introducir errores por accidente o que la implementación genere regresiones.
- Versionado de código: Aunque no sea una técnica a priori la cual podemos implementar y podrá tener impacto en el código, es una practica la cual nos ayuda a tener múltiples versiones de código de tal forma que podamos implementar funcionalidades sin riesgos de romper el código en su totalidad o sin contar con un respaldo, al igual que nos permite que si se introdujo un error poder regresar a una versión donde dicho error no existía o saber quién introdujo el error.

## Conclusión

Como se pudo observar existen muchas técnicas y muchas mas herramientas las cuales nos podrán ayudar a resolver los problemas/errores que podamos introducir en el desarrollo de software, las cuales podrán hacer la vida más sencilla para aquellos que tengan dichas complejidades.

Por lo que no podemos dejar de lado la posibilidad de decir que una aplicación va a fallar, ya que es imposible que una aplicación funcione al 100% con todas las tangentes posibles, porque se pueden presentar incluso errores los cuales son completamente ajenos a la aplicación. Por esto es fundamental el tener a la mano las diferentes estrategias o métodos que se pueden usar para minimizar el impacto que podrían generar dichos errores.

## Referencias

- 3 formas de gestionar errores en los lenguajes de programación. (2020, octubre 4). Blog Bitix. <https://picodotdev.github.io/blog-bitix/2020/10/3-formas-de-gestionar-errores-en-los-lenguajes-de-programacion/>
- 8. Errores y excepciones. (s/f). Python documentation. Recuperado el 26 de agosto de 2023, de <https://docs.python.org/es/3/tutorial/errors.html>
- Argalias, S. (2021, julio 26). Best practices for error catching and handling. Programming Duck. <https://programmingduck.com/articles/error-catching-handling>
- BillWagner. (s/f). Excepciones y control de excepciones. Microsoft.com. Recuperado el 26 de agosto de 2023, de <https://learn.microsoft.com/es-es/dotnet/csharp/fundamentals/exceptions/>
- Error, error handling and error handling technique. (2023, mayo 4). DEV Community. <https://dev.to/chiedoxie/error-error-handling-and-error-handling-technique-4ge8>
- González, J. D. M. (2018, marzo 24). Errores de programación y su solución. Programarya.com; ProgramarYa. <https://www.programarya.com/Cursos/Fundamentacion/Errores>
- Writing helpful error messages. (s/f). Google for Developers. Recuperado el 26 de agosto de 2023, de <https://developers.google.com/tech-writing/error-messages>
- Alumni Network. (2020, enero 26). JavaScript try catch: Exception handling explained. Freecodecamp.org. <https://www.freecodecamp.org/news/error-handling-and-try-catch-throw/>