



**UNIVERSIDAD DE GUADALAJARA**  
**CENTRO UNIVERSITARIO**  
**DE CIENCIAS EXACTAS E**  
**INGENIERÍAS**



**Computación Tolerante a Fallas**

(Par. 01) Otras herramientas para el manejo de  
errores

Ingeniería en Computación

29 Agosto 2023

Por: Carlos Uriel Salcido Aviña 217560751

Maestro: Michel Emanuel López Franco

Mientras estamos en la labor de programar, es posible que nuestro código tenga la posibilidad de fallar, por ejemplo, si queremos insertar ocho elementos una tabla que puede contener solo cinco, o si durante la ejecución del programa, una variable toma el valor de una letra, y luego intentamos hacer operaciones aritméticas con ella, o si intentamos hacer pop en una pila con cero elementos. Cuando errores de este tipo ocurren, lo más normal es que simplemente nuestro programa truene y debamos iniciarlo desde cero. Esto puede tener poca importancia con programas sencillos, pero a la hora de trabajar con códigos más grandes o incluso desarrollar un software con el que se pretende ganar dinero, un programa que truena demuestra poca profesionalidad. Es en estas situaciones cuando podemos hacer uso de diversas herramientas, según el lenguaje con el que se trabaje.

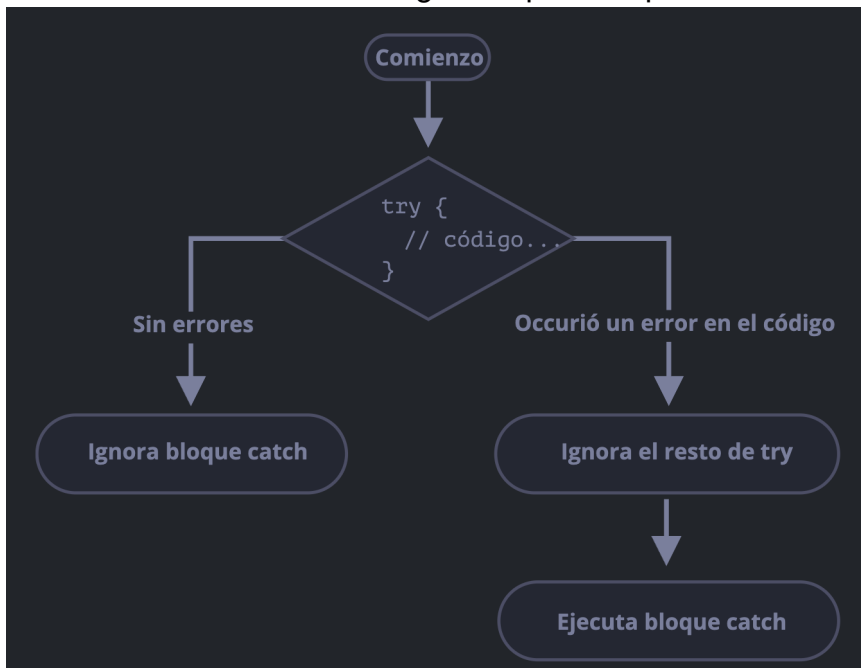
### Try Catch

En lo personal, Java es uno de los lenguajes con el que más familiaridad tengo, gracias al editor de Apache NetBeans. Mientras lo aprendía me encontré varias veces con la instrucción *try – catch*, cuyo uso no comprendí hasta que sucedió un error.

Esta sentencia funciona así:

- Primero se ejecuta el código escrito en la sección *try*.
- Si no se encuentran errores, se ignora el código escrito en *catch (err)*.
- Si se produce un error, la ejecución de *try* se detiene y el control fluye a la sección de *catch (err)*. La variable *err* contendrá un objeto de error con detalles sobre lo que sucedió, el cuál de hecho se puede imprimir para poder leer qué sucedió.

Encontré en internet este diagrama que lo explica.



## Try Exception

Por otro lado, en Python se pueden manejar las excepciones, gracias a la instrucción *try – exception*. Esta funciona de forma idéntica al *try catch* que existe en Java, con la diferencia de que no tengo mucha familiaridad con esta, aunque para eso es esta investigación.

Su funcionalidad es la siguiente:

- 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 del código dentro 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; si no se encuentra un gestor, se genera una *unhandled exception* y la ejecución se interrumpe con un mensaje.

De esta forma, es importante remarcar que la cláusula de la excepción puede o no especificar el tipo de excepción, de modo que podemos enfocarnos en un solo error en específico, y por lo general será la terminal la que nos da más detalles. En la parte dos de esta actividad hay ejemplos.

## Referencias

S/A (2023) “Manejo de errores, “try...catch””, en JavaScript.info. Disponible en: <https://es.javascript.info/try-catch>

S/A (S/F) “8.Errores y excepciones”, en Python. Disponible en: <https://docs.python.org/es/3/tutorial/errors.html>