

**CENTRO UNIVERSITARIO DE CIENCIAS EXACTAS E  
INGENIERÍAS (CUCEI)**

**DIVISIÓN DE ELECTRÓNICA Y COMPUTACIÓN**

**DEPARTAMENTO DE CIENCIAS COMPUTACIONALES**

**Carrera:** Ingeniería en Computación

**Nombre Materia:** Computación Tolerante a Fallas

**Profesor:** López Franco Michel Emanuel

**SECCIÓN:** Do6

**Nombre alumno:** López Arellano Ricardo David

**CODIGO:** 217136143



**(Part. 2) Otras herramientas para el manejar errores**

**Fecha de entrega:** 29/08/2023

## (Par. 2) Otras herramientas para el manejar errores

Una herramienta comúnmente utilizada para el manejo de errores en la programación es el uso de excepciones. Las excepciones son eventos que ocurren durante la ejecución de un programa y que interrumpen el flujo normal de ejecución debido a un error. Permiten que el programa maneje situaciones inesperadas y tome medidas adecuadas en lugar de simplemente fallar.

Por supuesto, aquí tienes un ejemplo de cómo usar bloques try y catch en Python para manejar excepciones:

```
try:
    num = int(input("Ingresa un número: "))
    result = 10 / num
    print("El resultado es:", result)
except ZeroDivisionError:
    print("Error: No puedes dividir entre cero.")
except ValueError:
    print("Error: Debes ingresar un valor numérico.")
except Exception as e:
    print("Ocurrió un error:", e)
else:
    print("No se generaron errores.")
finally:
    print("Este bloque siempre se ejecuta, sin importar qué.")

print("¡Programa terminado!")
```

En este ejemplo, el programa intentará realizar una división y manejará diferentes tipos de excepciones:

- Si el usuario ingresa cero, se capturará una excepción `ZeroDivisionError`.
- Si el usuario ingresa algo que no es un número, se capturará una excepción `ValueError`.
- Si ocurre cualquier otra excepción, se capturará en el bloque `except Exception as e`.
- El bloque `else` se ejecutará si no se generan excepciones.
- El bloque `finally` se ejecutará siempre, independientemente de si ocurre una excepción o no.

Recuerda que es importante manejar excepciones de manera adecuada en tu código para que puedas identificar y solucionar problemas potenciales sin que el programa se detenga abruptamente