

Universidad de Guadalajara
CENTRO UNIVERSITARIO DE CIENCIAS EXACTAS E
INGENIERÍAS

COMPUTACIÓN TOLERANTE A FALLAS



Otras herramientas para el manejar errores 2

Murillo Cortes, Jeanette. Código: 216455164



Dr. Michel Emanuel López Franco
Computación Tolerante a Fallas – Sección D06

Guadalajara, Jalisco. A Lunes 31 de Enero de 2022
Calendario 2022A, LM 10:00 – 12:55

Reporte

Otras herramientas para manejar errores

Para manejar excepciones, en Python se pueden utilizar excepciones ya predefinidas. Para este programa, se utilizaron las siguientes:

- try * except * else * finally * ValueError * TypeError

Try se utiliza para probar bloques de código, como se muestra en la imagen, que prueba lo que el usuario ingresa en la variable 'entero'.

Except funciona cuando existe un error en lo que probó try, como se muestra en la imagen, que si existe un error de tipo de dato o de valor, va a imprimir el correspondiente mensaje de acuerdo a si la excepción es de tipo 'ValueError' o 'TypeError'.

Por último, 'finally' nos muestra lo que va a ejecutar el programa, independientemente del resultado de 'try', que en este caso marca que el programa ya se ha ejecutado.

Imagen del código:

```
main.py ×
1  # Python
2  # Otras herramientas para el manejo de errores
3
4  while True:
5
6      try:
7          entero = int(input("Ingresa un numero entero: "))
8
9      except ValueError:
10         print ("Error")
11
12     except TypeError:
13         print ("Tipo de dato invalido")
14
15     else:
16         print(entero)
17
18     finally:
19         print("Se ha ejecutado el programa\n")
20
```

Imagen del programa en ejecución:

Console

Shell

```

Ingresa un numero entero: 4.5
Error
Se ha ejecutado el programa

Ingresa un numero entero: a
Error
Se ha ejecutado el programa

Ingresa un numero entero: 's'
Error
Se ha ejecutado el programa

Ingresa un numero entero: 4
4
Se ha ejecutado el programa

Ingresa un numero entero: 

```

Como muestra el programa, se ingresan los valores `'4.5'`, `'a'`, `"'s'"`, y el programa nos comenta que es un error debido al `'except'`.

Sin embargo, cuando el programa se ejecuta correctamente, `'else'` imprime el número que ingresó el usuario.

`'finally'` por su parte, comenta que el programa se ha ejecutado.

Conclusiones

La implementación del programa con manejo de excepciones nos permite implementar un mejor análisis de cómo manejar un error. Como Python tiene predefinidas algunas de estas excepciones, hace que sea más sencillo e intuitivo su implementación.