

Ejercicio Módulo 13

A un ejemplo en donde se busca un archivo "Test.txt" que no existe en el directorio, generar las excepciones y mensajes correspondientes:

- Función CATCH (raise?)
- Excepciones y las funciones TRY/EXCEPT
- Ejemplos con TRY / EXCEPT / ELSE / FINALLY

In []: *# Error original: tratar de abrir el archivo que no existe en el directorio*

```
with open('Test.txt') as file:
    pass
```

```
-----
FileNotFoundError                                Traceback (most recent call last)
Cell In [12], line 1
----> 1 with open('Test.txt') as file:
      2     pass
```

FileNotFoundError: [Errno 2] No such file or directory: 'Test.txt'

In []: *# Raise Exception: Verificar si el archivo existe, si no existe, Levantar un error*
Obtener la lista de nombres de los archivos que se encuentran en la carpeta actual
import os
curr_dir = os.getcwd()
dir = os.listdir(curr_dir)

```
# Verificar que el archivo Test.txt se encuentre en el directorio actual, si no es
if 'Test.txt' not in dir:
    raise Exception('El archivo Test.txt no se encuentra en la carpeta actual')

# En caso de que encontrara el archivo, con este bloque lo abriría
else:
    with open('Test.txt') as file:
        pass
```

```
-----
Exception                                Traceback (most recent call last)
Cell In [13], line 9
      7 # Verificar que el archivo Test.txt se encuentre en el directorio actual, s
i no es así, mandar un error con un mensaje personalizado
      8 if 'Test.txt' not in dir:
----> 9     raise Exception('El archivo Test.txt no se encuentra en la carpeta actu
al')
     11 # En caso de que encontrara el archivo, con este bloque lo abriría
     12 else:
     13     with open('Test.txt') as file:
```

Exception: El archivo Test.txt no se encuentra en la carpeta actual

```
In [ ]: # Try - Except: Intenta abrir el abrir el archivo, si no lo encuentra, corre el exc
try:
    with open('Test.txt') as file:
        pass
except FileNotFoundError as error:
    print('El archivo Test.txt no se encuentra en la carpeta actual')
    raise error
```

El archivo Test.txt no se encuentra en la carpeta actual

FileNotFoundError Traceback (most recent call last)

Cell In [18], line 7

```
5 except FileNotFoundError as error:
6     print('El archivo Test.txt no se encuentra en la carpeta actual')
----> 7     raise error
```

Cell In [18], line 3

```
1 # Try - Except: Intenta abrir el abrir el archivo, si no lo encuentra, corr
e el except FileNotFoundError y arroja el error
2 try:
----> 3     with open('Test.txt') as file:
4         pass
5 except FileNotFoundError as error:
```

FileNotFoundError: [Errno 2] No such file or directory: 'Test.txt'

```
In [ ]: # Try / Except / Else / Finally
```

Intenta abrir y cerrar el archivo para ver que exista

```
try:
    f = open('Test.txt')
    f.close()
```

#Si no existe (FileNotFoundError) arroja un print y el error

```
except FileNotFoundError as error:
    print('El archivo Test.txt no existe en el directorio')
    raise error
```

Si encontrara el archivo, correría el siguiente bloque, en este caso no hace nada

```
else:
    with open('Test.txt') as file:
        pass
```

#Finalmente hace un print de que terminó de correr el bloque

```
finally:
    print('Proceso terminado')
```

El archivo Test.txt no existe en el directorio

Proceso terminado

```
-----
FileNotFoundError                                Traceback (most recent call last)
Cell In [20], line 11
      9 except FileNotFoundError as error:
     10     print('El archivo Test.txt no existe en el directorio')
----> 11     raise error
     13 # Si encontrara el archivo, correría el siguiente bloque, en este caso no h
ace nada
     14 else:
     15     with open('Test.txt') as file:
```

```
Cell In [20], line 5
      1 # Try / Except / Else / Finally
      2
      3 # Intenta abrir y cerrar el archivo para ver que exista
      4 try:
----> 5     f = open('Test.txt')
      6     f.close()
      8 #Si no existe (FileNotFoundError) arroja un print y el error
```

```
FileNotFoundError: [Errno 2] No such file or directory: 'Test.txt'
```

Debugging:

Generar en una celda una operación matemática simple y mostrar los diferentes pasos que se siguieron para su debugging

```
In [ ]: # Simular un error de lógica por falta de paréntesis
a = 10
b = 20
c = 10
d = 20

# Operación correcta: (a+b) / (c+d), Debería dar 30 / 30 = 1
resultado = a + b / c + d

if resultado == 1:
    print('Correcto')
else:
    print('Algo anda mal')
```

```
32.0
```

```
In [ ]: a = 10
        b = 20
        c = 10
        d = 20

        # Operación correcta: (a+b) / (c+d), Debería dar 30 / 30 = 1
        resultado = a + b / c + d

        # Poner el stop justo en donde se evalúa la condición y poner la variable que evalúa
        # Reviso la operación donde se define resultado y me doy cuenta que, por jerarquía
        if resultado == 1:
            print('Correcto')
        else:
            print('Algo anda mal')
```

Algo anda mal

```
In [ ]: # Bloque de código corregido
        a = 10
        b = 20
        c = 10
        d = 20

        # Operación correcta: (a+b) / (c+d), Debería dar 30 / 30 = 1
        resultado = (a + b) / (c + d)

        if resultado == 1:
            print('Correcto')
        else:
            print('Algo anda mal')
```

Correcto