

Uso de tracebacks para buscar errores

Tracebacks

```
open("/path/to/mars.jpg")
!python3 open.py

⊗ 0.9s Python

-----
-----
FileNotFoundError                                Traceback (most recent call last)
c:\Users\ALMA\Desktop\CursoFull\Katas\Kata10\Kata 10.ipynb Cell 12' in <module>
----> 1 open("/path/to/mars.jpg")
      2 get_ipython().system('python3 open.py')

FileNotFoundError: [Errno 2] No such file or directory: '/path/to/mars.jpg'
```

Python3 open.py

```
C:\Users\ALMA\Desktop\CursoFull\Katas\Kata10>python open.py
Traceback (most recent call last):
  File "C:\Users\ALMA\Desktop\CursoFull\Katas\Kata10\open.py", line 5, in <module>
    main()
  File "C:\Users\ALMA\Desktop\CursoFull\Katas\Kata10\open.py", line 2, in main
    open("/path/to/mars.jpg")
FileNotFoundError: [Errno 2] No such file or directory: '/path/to/mars.jpg'

C:\Users\ALMA\Desktop\CursoFull\Katas\Kata10>
```

Controlando las excepciones

```
try:
    open('config.txt')
except FileNotFoundError:
    print("Couldn't find the config.txt file!")
```

[6] ✓ 0.8s

... Couldn't find the config.txt file!

```
main()
File "C:\Users\ALMA\Desktop\CursoFull\Katas\Kata10\config.py", line 4, in main
    configuration = open('config.txt')
PermissionError: [Errno 13] Permission denied: 'config.txt'

C:\Users\ALMA\Desktop\CursoFull\Katas\Kata10>python config.py
Couldn't find the config.txt file!

C:\Users\ALMA\Desktop\CursoFull\Katas\Kata10>python config.py
Found config.txt but it is a directory, couldn't read it

C:\Users\ALMA\Desktop\CursoFull\Katas\Kata10>
```

Borrar archivo config.txt y luego buscarlo

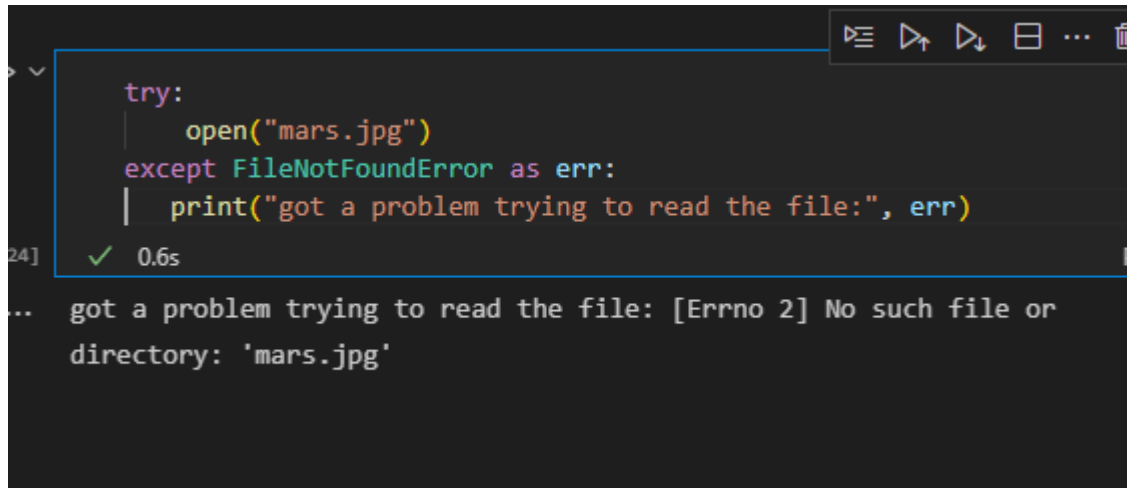
PROBLEMS 12 OUTPUT DEBUG CONSOLE TERMINAL JUPYTER

```
Couldn't find the config.txt file!

C:\Users\ALMA\Desktop\CursoFull\Katas\Kata10>python config.py
Couldn't find the config.txt file!

C:\Users\ALMA\Desktop\CursoFull\Katas\Kata10>python config.py
Couldn't find the config.txt file!
```

detectar BlockingIOError y TimeoutError juntos:

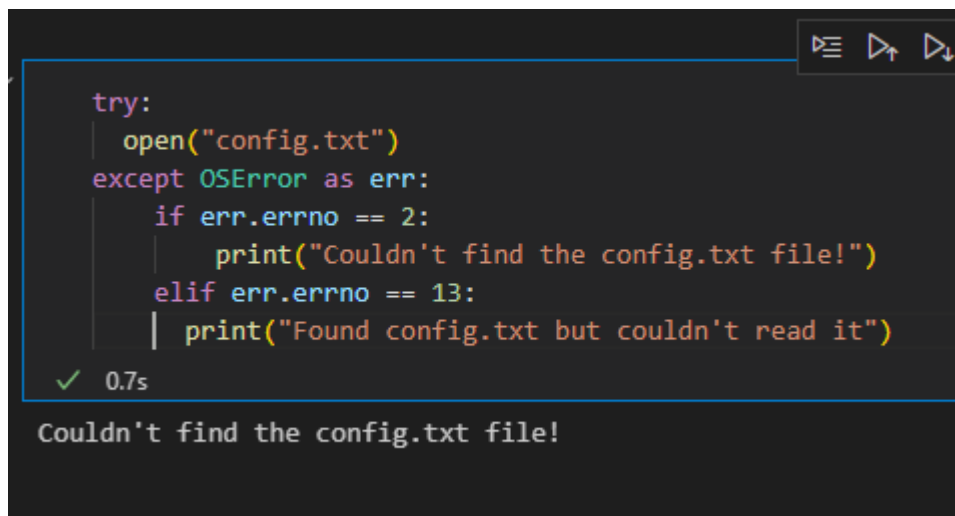


```
try:
    open("mars.jpg")
except FileNotFoundError as err:
    print("got a problem trying to read the file:", err)
```

24] ✓ 0.6s

... got a problem trying to read the file: [Errno 2] No such file or directory: 'mars.jpg'

detecta una excepción OSError más genérica



```
try:
    open("config.txt")
except OSError as err:
    if err.errno == 2:
        print("Couldn't find the config.txt file!")
    elif err.errno == 13:
        print("Found config.txt but couldn't read it")
```

✓ 0.7s

Couldn't find the config.txt file!

Generación de excepciones

```
def water_left(astronauts, water_left, days_left):
    daily_usage = astronauts * 11
    total_usage = daily_usage * days_left
    total_water_left = water_left - total_usage
    return f"Total water left after {days_left} days is: {total_water_left} liters"

water_left(5, 100, 2)
```

✓ 0.7s Python

'Total water left after 2 days is: -10 liters'

```
> def water_left(astronauts, water_left, days_left): ...
    water_left(5, 100, 2)
```

[30] ✗ 0.9s Python

...

```
-----
RuntimeError                                Traceback (most recent call last)
c:\Users\ALMA\Desktop\CursoFull\Katas\Kata10\Kata 10.ipynb Cell 10' in <module>
      6         raise RuntimeError(f"There is not enough water for {astronauts} a
astronauts after {days_left} days!")
      7         return f"Total water left after {days_left} days is: {total_water_lef
t} liters"
----> 8 water_left(5, 100, 2)

c:\Users\ALMA\Desktop\CursoFull\Katas\Kata10\Kata 10.ipynb Cell 10' in water_left
(astronauts, water_left, days_left)
      4 total_water_left = water_left - total_usage
      5 if total_water_left < 0:
----> 6         raise RuntimeError(f"There is not enough water for {astronauts} astro
nauts after {days_left} days!")
      7 return f"Total water left after {days_left} days is: {total_water_left} l
iters"

RuntimeError: There is not enough water for 5 astronauts after 2 days!
```

```
c:\Users\ALMA\Desktop\CursoFull\Katas\Kata10\Kata 10.ipynb Cell 11' in water_left
(astronauts, water_left, days_left)
     5         argument / 10
     6     except TypeError:
     7         # TypeError will be raised only if it isn't the right type
     8         # Raise the same exception but with a better error message
----> 9         raise TypeError(f"All arguments must be of type int, but receive
d: '{argument}'")
    10 daily_usage = astronauts * 11
    11 total_usage = daily_usage * days_left

TypeError: All arguments must be of type int, but received: '3'
```