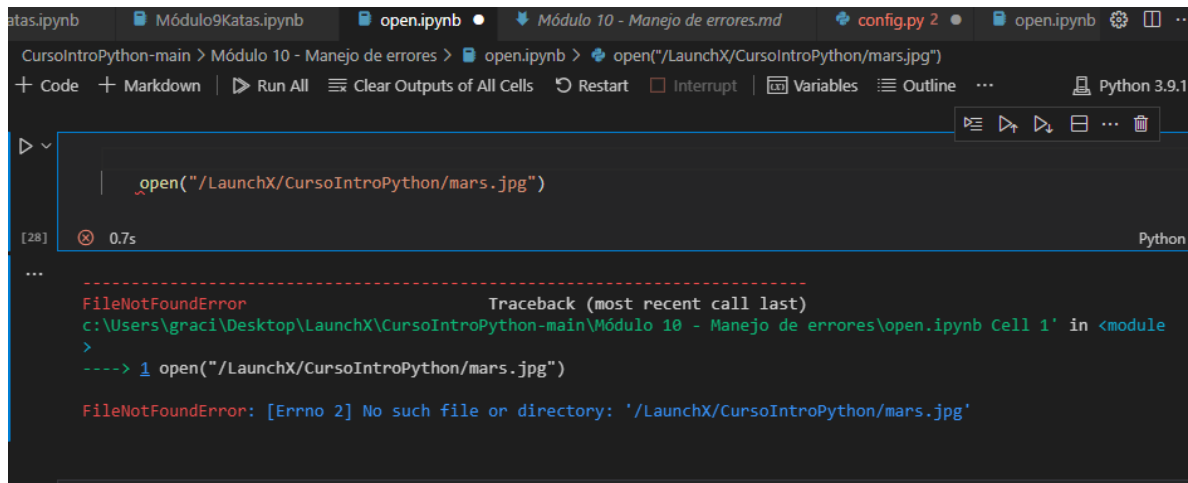


Kata10_Graciela Sánchez López



The screenshot shows a Jupyter Notebook interface with several tabs at the top: 'atas.ipynb', 'Módulo9Katas.ipynb', 'open.ipynb', 'Módulo 10 - Manejo de errores.md', 'config.py 2', and another 'open.ipynb'. The active cell in 'open.ipynb' contains the code `open("/LaunchX/CursoIntroPython/mars.jpg")`. Below the code, the execution output shows a `FileNotFoundError` with a traceback. The traceback indicates the error occurred in 'Cell 1' of the 'open.ipynb' file. The error message is: `FileNotFoundError: [Errno 2] No such file or directory: '/LaunchX/CursoIntroPython/mars.jpg'`.

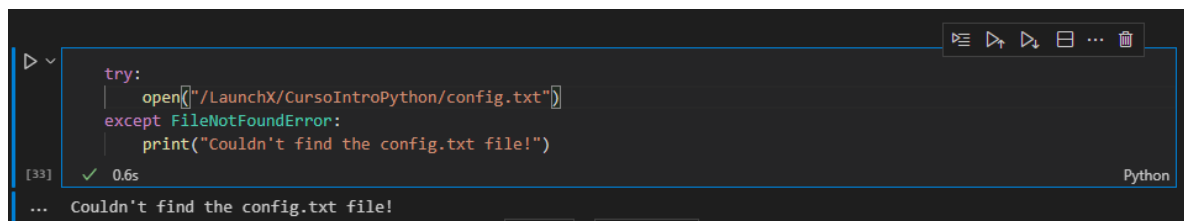
```
open("/LaunchX/CursoIntroPython/mars.jpg")
```

[28] 0.7s Python

```
-----
FileNotFoundError                                Traceback (most recent call last)
c:\Users\graci\Desktop\LaunchX\CursoIntroPython-main\Módulo 10 - Manejo de errores\open.ipynb Cell 1' in <module>
>
----> 1 open("/LaunchX/CursoIntroPython/mars.jpg")

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

```
C:\Users\graci\Desktop\LaunchX\CursoIntroPython-main\Módulo 0 - Preparación del entorno de trabajo\Evirtual_python
>python3 open.ipynb
python3: can't open file 'C:\Users\graci\Desktop\LaunchX\CursoIntroPython-main\Módulo 0 - Preparación del entorno de trabajo\Evirtual_python\open.ipynb': [Errno 2] No such file or directory
```

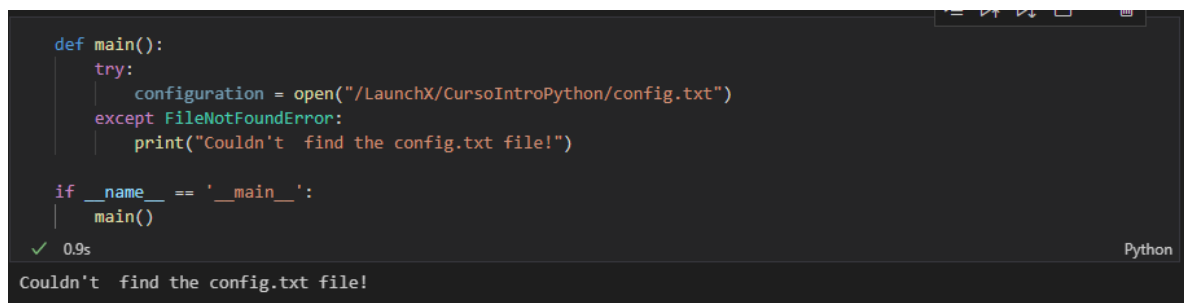


The screenshot shows a Jupyter Notebook cell with a try-except block. The code attempts to open a file named 'config.txt' in the directory '/LaunchX/CursoIntroPython/'. If the file is not found, it prints a message: "Couldn't find the config.txt file!". The execution output shows the message being printed.

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

[33] 0.6s Python

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



The screenshot shows a Jupyter Notebook cell with a `main` function. The function attempts to open a file named 'config.txt' in the directory '/LaunchX/CursoIntroPython/'. If the file is not found, it prints a message: "Couldn't find the config.txt file!". The execution output shows the message being printed.

```
def main():
    try:
        configuration = open("/LaunchX/CursoIntroPython/config.txt")
    except FileNotFoundError:
        print("Couldn't find the config.txt file!")

if __name__ == '__main__':
    main()
```

0.9s Python

Couldn't find the config.txt file!

```
def main():
    try:
        configuration = open("/LaunchX/CursoIntroPython/config.txt")
    except Exception:
        print("Couldn't find the config.txt")
```

✓ 0.7s Python

```
def main():
    try:
        configuration = open("/LaunchX/CursoIntroPython/config.txt")
    except FileNotFoundError:
        print("Couldn't find the config.txt file!")
    except IsADirectoryError:
        print("Found config.txt but it is a directory, couldn't read it")
```

✓ 0.6s Python

```
C:\Users\graci\Desktop\LaunchX\CursoIntroPython-main\Módulo 10 - Manejo de errores>python3 config.ipynb
Traceback (most recent call last):
  File "C:\Users\graci\Desktop\LaunchX\CursoIntroPython-main\Módulo 10 - Manejo de errores\config.ipynb", line 5,
in <module>
    "execution_count": null,
NameError: name 'null' is not defined
```

```
def main():
    try:
        configuration = open("/LaunchX/CursoIntroPython-main/Módulo 10 - Manejo de errores/config.txt")
    except FileNotFoundError:
        print("Couldn't find the config.txt file!")
    except IsADirectoryError:
        print("Found config.txt but it is a directory, couldn't read it")
    except (BlockingIOError, TimeoutError):
        print("Filesystem under heavy load, can't complet reading configuration file")
```

✓ 0.8s Python

```
def main():
    try:
        configuration = open("/LaunchX/CursoIntroPython-main/Módulo 10 - Manejo de errores/mars.jpg")
    except FileNotFoundError as err:
        print("Couldn't find the config.txt file!",err)
```

19] ✓ 0.4s Python

```
try:
    open("/LaunchX/CursoIntroPython-main/Módulo 10 - Manejo de errores/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!")
```

[21] ✓ 0.7s Python

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

```
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 ledt after {days_left} days is: {total_water_left} liters"
```

✓ 0.9s Python

```
water_left(5, 100, 2)
```

✓ 0.1s Python

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

```
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

    if total_water_left < 0:
        raise RuntimeError(f"There is not enough water for {astronauts} astronauts after {days_left} days!")
    return f"Total water ledt after {days_left} days is: {total_water_left} liters"
```

[25] ✓ 0.6s Python

```
water_left(5, 100, 2)
```

[26] ✗ 0.5s Python

...

RuntimeError Traceback (most recent call last)
c:\Users\graci\Desktop\LaunchX\CursoIntroPython-main\Módulo 10 - Manejo de errores\config.ipynb Cell 10' in <mod
ule>
----> 1 water_left(5, 100, 2)

c:\Users\graci\Desktop\LaunchX\CursoIntroPython-main\Módulo 10 - Manejo de errores\config.ipynb Cell 9' in water
_left(astronauts, water_left, days_left)
 4 total_water_left = water_left - total_usage
 6 if total_water_left < 0:
----> 7 raise RuntimeError(f"There is not enough water for {astronauts} astronauts after {days_left} days!")
 8 return f"Total water ledt after {days_left} days is: {total_water_left} liters"

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

```
try:
    water_left(5, 100, 2)
except RuntimeError as err:
    alert_navigation_system(err)
```

0.9s Python

RuntimeError Traceback (most recent call last)
c:\Users\graci\Desktop\LaunchX\CursoIntroPython-main\Módulo 10 - Manejo de errores\config.ipynb Cell 11' in <module>
1 try:
----> 2 water_left(5, 100, 2)
3 except RuntimeError as err:

c:\Users\graci\Desktop\LaunchX\CursoIntroPython-main\Módulo 10 - Manejo de errores\config.ipynb Cell 9' in water_left(astronauts, water_left, days_left)
6 if total_water_left < 0:
----> 7 raise RuntimeError(f"There is not enough water for {astronauts} astronauts after {days_left} days!")
8 return f"Total water left after {days_left} days is: {total_water_left} liters"

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

During handling of the above exception, another exception occurred:

NameError Traceback (most recent call last)
c:\Users\graci\Desktop\LaunchX\CursoIntroPython-main\Módulo 10 - Manejo de errores\config.ipynb Cell 11' in <module>
2 water_left(5, 100, 2)
3 except RuntimeError as err:
----> 4 alert_navigation_system(err)

NameError: name 'alert_navigation_system' is not defined

```
water_left(["3", "200", None])
```

0.1s Python

TypeError Traceback (most recent call last)
c:\Users\graci\Desktop\LaunchX\CursoIntroPython-main\Módulo 10 - Manejo de errores\config.ipynb Cell 12' in <module>
----> 1 water_left("3", "200", None)

c:\Users\graci\Desktop\LaunchX\CursoIntroPython-main\Módulo 10 - Manejo de errores\config.ipynb Cell 9' in water_left(astronauts, water_left, days_left)
1 def water_left(astronauts, water_left, days_left):
2 daily_usage = astronauts * 11
----> 3 total_usage = daily_usage * days_left
4 total_water_left = water_left - total_usage
6 if total_water_left < 0:

TypeError: can't multiply sequence by non-int of type 'NoneType'

```
def water_left(astronauts, water_left, days_left):  
    for arguments in [astronauts, water_left, days_left]:  
        try:  
            # if argument is an int, the following operation will work argument / 10  
        except TypeError:  
            # TypeError will be raised only if it isn't the right type  
            # Raise the same exception but with a better error message  
            raise TypeError(f"All argumets must be of type int, but received: '{arguments}' ")  
    daily_usage = astronauts * 11  
    total_usage = daily_usage * days_left  
    total_water_left = water_left - total_usage  
  
    if total_water_left < 0:  
        raise RuntimeError(f"There is not enough water for {astronauts} astronauts after {days_left} days!")  
    return f"Total water ledt after {days_left} days is: {total_water_left} liters"
```

0.7s

Python

Input In [30]

```
except TypeError:
```

IndentationError: expected an indented block

```
water_left("3", "200", None)
```

0.1s

Python