

The screenshot shows the Visual Studio Code interface with the Explorer sidebar on the left displaying a project named 'KATASFEROCHOA'. The file 'archivo.py' is selected. The main editor displays the following Python code:

```
1 def main():
2
3     open("/path/to/mars.jpg")
4
5
6 if __name__ == '__main__':
7
8     main()
9
10
```

The TERMINAL pane at the bottom shows the output of running the script in a Windows PowerShell environment. The error message is:

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

The status bar at the bottom indicates 'Python 3.10.2 64-bit' and 'Ln 11, Col 1'.

The screenshot shows the Visual Studio Code interface with the Explorer sidebar on the left displaying the same project. The file 'archivo.py' is selected. The main editor displays the following Python code:

```
8 # main()
9
10 try:
11
12     open('config.txt')
13 except FileNotFoundError:
14     print("couldn't find the config.txt file!")
15
16
```

The TERMINAL pane at the bottom shows the output of running the script. The error message is:

```
IndentationError: unindent does not match any outer indentation level
```

The status bar at the bottom indicates 'Python 3.10.2 64-bit' and 'Ln 12, Col 23'.

The screenshot shows the Visual Studio Code interface with a Python file named `archivo.py` open. The Explorer sidebar on the left shows a project structure with folders like `KATASFEROCHOA` and `Módulo 10`. The main editor displays the following code:

```
14
15
16 # $ python config.py
17
18 Traceback (most recent call last):
19
20 #File "/tmp/config.py", line 9, in <module>
21
22     _main()
23
24 #File "/tmp/config.py", line 3, in main
25
26     configuration = open('config.txt')
```

The terminal window at the bottom shows the following error messages:

```
except FileNotFoundError:
    ^
IndentationError: unindent does not match any other indentation level
PS C:\Users\Cristina\Desktop\katas\KatasFerOchoa> & C:/Users/Cristina/AppData/Local/Programs/Python/Python310/python.exe c:/User
s/Cristina/Desktop/katas/KatasFerOchoa/CursoIntroPython-main/Módulo10/archivo.py
File "c:\Users\Cristina\Desktop\katas\KatasFerOchoa\CursoIntroPython-main\Módulo10\archivo.py", line 6
    if __name__ == '__main__':
    ^
IndentationError: expected an indented block after function definition on line 1
PS C:\Users\Cristina\Desktop\katas\KatasFerOchoa>
```

The screenshot shows the Visual Studio Code interface with a document named `kata10.md` open. The Explorer sidebar on the left shows a project structure with folders like `KATASFEROCHOA` and `Módulo 10`. The main editor displays the following text:

大綱

- ▼ Introducción
- Escenario: Creación de un progr...
- ¿Qué descubrirás?
- ¿Cuál es el objetivo principal?
- Uso de tracebacks para buscar ...
- Tracebacks
- ▼ Controlando las excepciones
- Try y Except de los bloques
- Generación de excepciones
- Resumen

Ahora volvemos a ejecutarlo, en el mismo lugar donde `config.txt` está con el problema de permisos

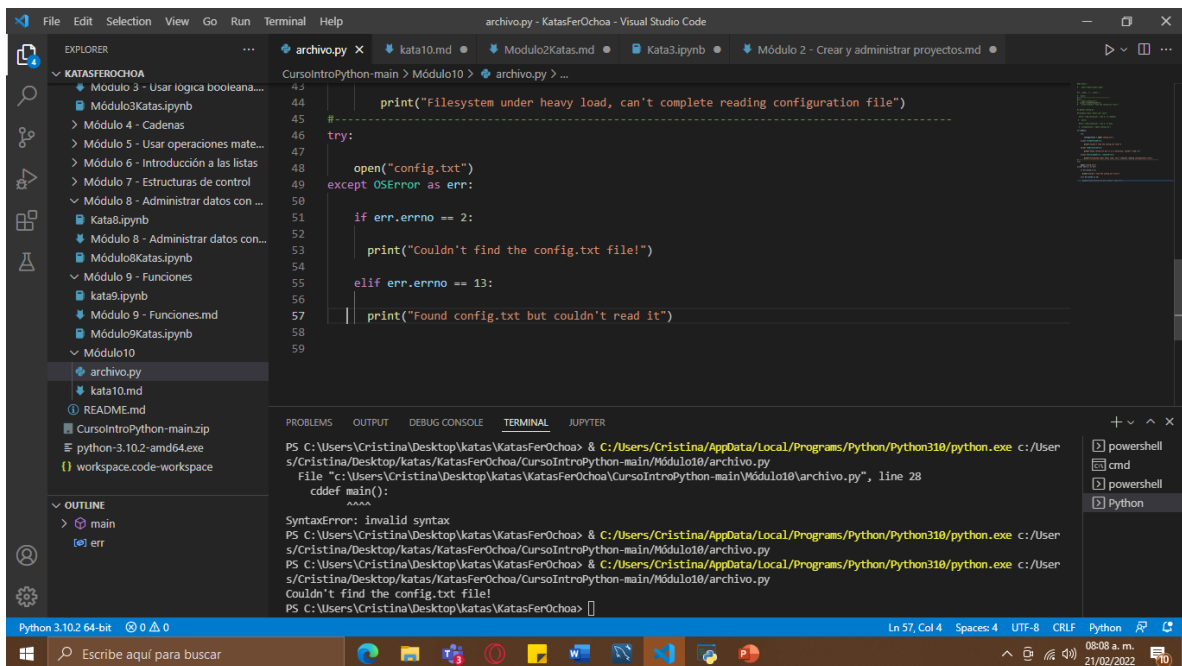
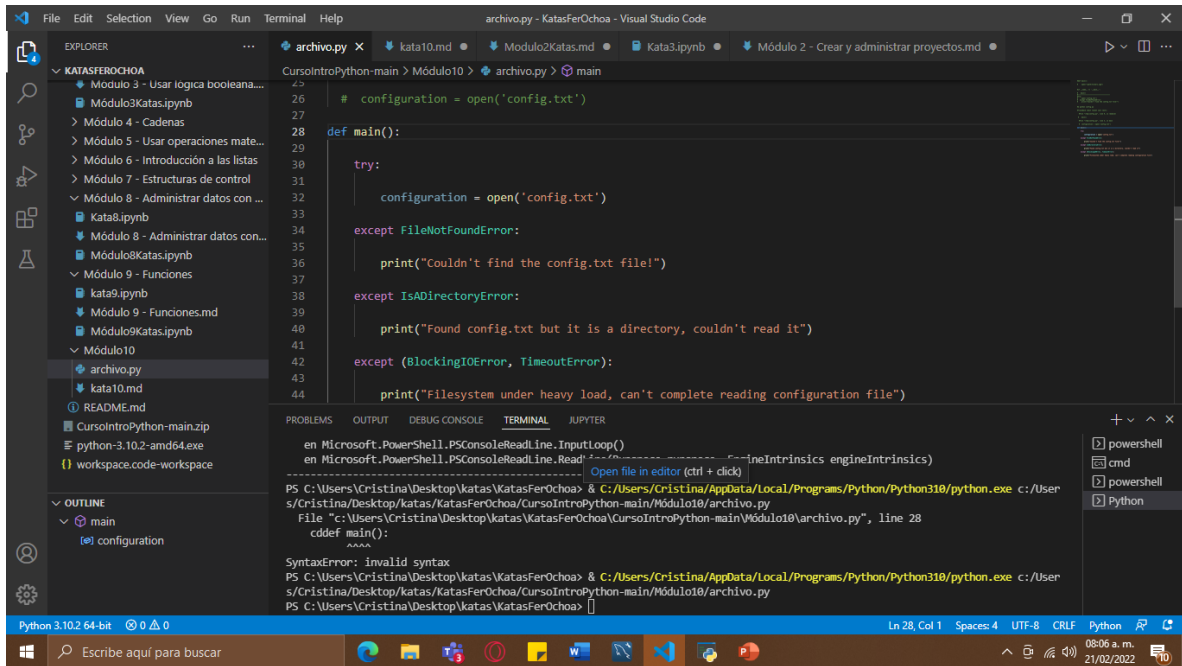
```
</>
$ python config.py
Found config.txt but couldn't read it
```

Eliminamos el archivo `config.txt` para asegurarnos de que se alcanza el primer bloque `except` en su lugar:

```
</>
$ rm -f config.txt
```

The terminal window at the bottom shows the following error messages:

```
IndentationError: expected an indented block after function definition on line 1
PS C:\Users\Cristina\Desktop\katas\KatasFerOchoa> & C:/Users/Cristina/AppData/Local/Programs/Python/Python310/python.exe c:/User
s/Cristina/Desktop/katas/KatasFerOchoa/CursoIntroPython-main/Módulo10/archivo.py
File "c:\Users\Cristina\Desktop\katas\KatasFerOchoa\CursoIntroPython-main\Módulo10\archivo.py", line 18
    Traceback (most recent call last):
    ^
> $ python config.py
> entationError: expected an indented block after function definition on line 1
> > Couldn't find the config.txt file!>KasFerOchoa>
```



Visual Studio Code interface showing a Python script named `archivo.py` in the `archivo.py` file. The script defines a function `water_left` that calculates the remaining water for astronauts based on the number of days left and the daily usage. The function raises a `RuntimeError` if there is not enough water.

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

water_left(5, 100, 2)
```

The terminal output shows the execution of the script, resulting in a `RuntimeError` because there is not enough water for 5 astronauts after 2 days.

```
PS C:\Users\Cristina\Desktop\katas\KatasFerOchoa\CursoIntroPython-main\Módulo10> python archivo.py
Traceback (most recent call last):
  File "c:\Users\Cristina\Desktop\katas\KatasFerOchoa\CursoIntroPython-main\Módulo10\archivo.py", line 87, in <module>
    water_left(5, 100, 2)
  File "c:\Users\Cristina\Desktop\katas\KatasFerOchoa\CursoIntroPython-main\Módulo10\archivo.py", line 83, in water_left
    raise RuntimeError(f"There is not enough water for {astronauts} astronauts after {days_left} days!")
RuntimeError: There is not enough water for 5 astronauts after 2 days!
PS C:\Users\Cristina\Desktop\katas\KatasFerOchoa>
```

Visual Studio Code interface showing the same Python script `archivo.py` in the `archivo.py` file. The script is modified to include a `try` block and a `finally` block to handle the `RuntimeError` and log the error.

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

water_left(5, 100, 2)

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

The terminal output shows the execution of the script, resulting in a `RuntimeError` because there is not enough water for 5 astronauts after 2 days. The error is caught and logged.

```
PS C:\Users\Cristina\Desktop\katas\KatasFerOchoa\CursoIntroPython-main\Módulo10> python archivo.py
Traceback (most recent call last):
  File "c:\Users\Cristina\Desktop\katas\KatasFerOchoa\CursoIntroPython-main\Módulo10\archivo.py", line 87, in <module>
    water_left(5, 100, 2)
  File "c:\Users\Cristina\Desktop\katas\KatasFerOchoa\CursoIntroPython-main\Módulo10\archivo.py", line 83, in water_left
    raise RuntimeError(f"There is not enough water for {astronauts} astronauts after {days_left} days!")
RuntimeError: There is not enough water for 5 astronauts after 2 days!
PS C:\Users\Cristina\Desktop\katas\KatasFerOchoa>
```