

Kata 10 evidencia:

Tracebacks

```
Módulo 10 - Manejo de errores > kata10.py > ...  
1 def main():  
2     | open("/path/to/mars.jpg")  
3  
4 if __name__ == '__main__':  
5     | main()  
  
PROBLEMS (24) OUTPUT DEBUG CONSOLE TERMINAL JUPYTER: VARIABLES  
  
Windows PowerShell  
Copyright (C) Microsoft Corporation. Todos los derechos reservados.  
  
Prueba la nueva tecnología PowerShell multiplataforma https://aka.ms/pscore6  
  
PS C:\Users\narno\Documents\CURSOS\Microsoft\CursoIntroPython-main> & C:\Users\narno\AppData\Local\Microsoft\WindowsApps\python3.9.exe  
Python 3.9.10 (tags/v3.9.10:f2f3f53, Jan 17 2022, 15:14:21) [MSC v.1929 64 bit (AMD64)] on win32  
Type "help", "copyright", "credits" or "license()" for more information.  
>>> def main():  
...     | open("/path/to/mars.jpg")  
...  
>>> if __name__ == '__main__':  
...     | main()  
... 
```

Controlando las excepciones

```
Módulo 10 - Manejo de errores > kata10.py  
1 try:  
2     | open('config.txt')  
3 except FileNotFoundError:  
4  
5     | print("Couldn't find the config.txt file!")  
  
PROBLEMS (26) OUTPUT DEBUG CONSOLE TERMINAL JUPYTER: VARIABLES  
  
SyntaxError: invalid syntax  
>>> ...     | print("Couldn't find the config.txt file!")  
file "<stdin>", line 1  
...     | print("Couldn't find the config.txt file!")  
...     ^  
SyntaxError: invalid syntax  
>>>  
>>> print("Couldn't find the config.txt file!")  
Couldn't find the config.txt file!  
>>> print("Couldn't find the config.txt file!")  
Couldn't find the config.txt file!  
>>> []
```

Módulo 10 - Manejo de errores > config.py > ...

```
1 def main():
2     try:
3         configuration = open('config.txt')
4     except FileNotFoundError:
5         print("Couldn't find the config.txt file!")
6
7
8 if __name__ == '__main__':
9     main()
```

PROBLEMS (26) OUTPUT DEBUG CONSOLE TERMINAL JUPYTER VARIABLES

```
...         print("Couldn't find the config.txt file!")
...
>>>
>>> if __name__ == '__main__':
...     main()
... main()
File "<stdin>", line 3
    main()
    ^
SyntaxError: invalid syntax
>>> main()
Couldn't find the config.txt file!
>>>
```

Módulo 10 - Manejo de errores > config.py > main

```
1 def main():
2     try:
3         configuration = open('config.txt')
4     except FileNotFoundError:
5         print("Couldn't find the config.txt file!")
6     except IsADirectoryError:
7         print("Found config.txt but it is a directory, couldn't read it")
```

PROBLEMS (26) OUTPUT DEBUG CONSOLE TERMINAL JUPYTER VARIABLES

```
>>>
>>>
>>>
>>>
>>>
>>>
>>>
>>>
>>>
>>> print("Found config.txt but it is a directory, couldn't read it")
Found config.txt but it is a directory, couldn't read it
```

Módulo 10 - Manejo de errores > config.py > main

```
1 def main():
2     try:
3         configuration = open('config.txt')
4     except FileNotFoundError:
5         print("couldn't find the config.txt file!")
6     except IsADirectoryError:
7         print("Found config.txt but it is a directory, couldn't read it")
8     except (BlockingIOError, TimeoutError):
9         print("Filesystem under heavy load, can't complete reading configuration file")
```

PROBLEMS (26) OUTPUT DEBUG CONSOLE TERMINAL JUPYTER: VARIABLES

```
>>>
>>>
>>>
>>>
>>>
>>>
>>>
>>>
>>>
>>> print("Filesystem under heavy load, can't complete reading configuration file")
Filesystem under heavy load, can't complete reading configuration file
```

Módulo 10 - Manejo de errores > config.py > ...

```
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
5     return f"Total water left after {days_left} days is: {total_water_left} liters"
6
7 print(water_left(5, 100, 2))
```

PROBLEMS (26) OUTPUT DEBUG CONSOLE TERMINAL JUPYTER: VARIABLES

```
File "<stdin>", line 1, in <module>
NameError: name 'water_left' is not defined
>>> def water_left(astronauts, water_left, days_left):
...
File "<stdin>", line 2
^
IndentationError: expected an indented block
>>> print(water_left(5, 100, 2))
Traceback (most recent call last):
File "<stdin>", line 1, in <module>
NameError: name 'water_left' is not defined
```

```
Módulo 10 - Manejo de errores > config.py > water_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
5     if total_water_left < 0:
6         raise RuntimeError(f"There is not enough water for {astronauts} astronauts after {days_left} days!")
7     return f"Total water left after {days_left} days is: {total_water_left} liters"
8
9 print(water_left(5, 100, 2))
```

PROBLEMS (26) OUTPUT DEBUG CONSOLE **TERMINAL** JUPYTER-VARIABLES

```
File "<stdin>", line 1
SyntaxError: 'return' outside function
>>> print(water_left(5, 100, 2))
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
NameError: name 'water_left' is not defined
>>> def water_left(astronauts, water_left, days_left):
...
File "<stdin>", line 2
```

```
Módulo 10 - Manejo de errores > config.py > water_left
1 def water_left(astronauts, water_left, days_left):
2     for argument in [astronauts, water_left, days_left]:
3         try:
4             # If argument is an int, the following operation will work
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 received: '{argument}'")
10    daily_usage = astronauts * 11
11    total_usage = daily_usage * days_left
12    total_water_left = water_left - total_usage
13    if total_water_left < 0:
14        raise RuntimeError(f"There is not enough water for {astronauts} astronauts after {days_left} days!")
15    return f"Total water left after {days_left} days is: {total_water_left} liters"
16
17 print(water_left("3", "200", None))
```

PROBLEMS (26) OUTPUT DEBUG CONSOLE **TERMINAL** JUPYTER-VARIABLES

```
>>>
>>>
>>>
>>>
>>>
>>>
>>> print(water_left("3", "200", None))
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
NameError: name 'water_left' is not defined
```