





Visual Studio Code interface showing a Python project named "CURSORINTROPYTHON-MYNOTES". The Explorer sidebar on the left lists the project structure, including modules 0 through 10. The main editor displays the "config.txt" file, which contains a Python script for handling file operations and errors.

```
</> def main():
    try:
        configuration = open('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")

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
$ python config.py
Couldn't find the config.txt file!

Cuando los errores son de una naturaleza similar y no es necesario controlarlos individualmente, puedes agrupar las excepciones como una usando paréntesis en la línea except. Por ejemplo, si el sistema de navegación está bajo cargas pesadas y el sistema de archivos está demasiado ocupado, tiene sentido detectar BlockingIOError y TimeoutError juntos:

</> def main():
    try:
        configuration = open('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 complete reading configuration file")
```

The bottom terminal window shows the execution of the script in a bash shell. The user runs the command `$ rm -f config.txt` to delete the file, and then runs `$ python config.py` to execute the script, which outputs `Couldn't find the config.txt file!`.

