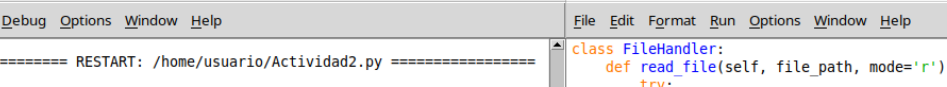


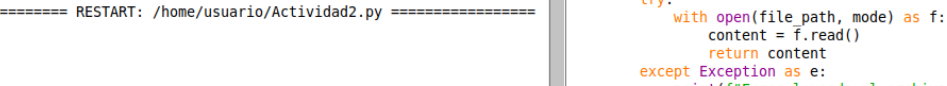
Actividades Gestión de información almacenada en ficheros.

Actividad 2 de clase:

Adjunta la captura de pantalla habitual de un programa en Python que hagan uso de esta clase. El programa debe usar un archivo que se llame "12345678.txt", siendo el nombre del fichero tu DNI. El programa debe escribir en dicho archivo una cadena de texto que sea tu fecha de nacimiento y debe leerla luego del fichero. Por último el programa ha de mostrar por pantalla lo que haya leído del fichero.



```
IDLE Shell 3.12.3
File Edit Shell Debug Options Window Help
>>> ===== RESTART: /home/usuario/Actividad2.py =====
>>> ===== RESTART: /home/usuario/Actividad2.py =====
28/12/2002
>>>
>>>
>>>
...
>>>
...
...
>>>
>>>
```



```
Actividad2.py - /home/usuario/Actividad2.py (3.12)
File Edit Format Run Options Window Help

class FileHandler:
    def read_file(self, file_path, mode='r'):
        try:
            with open(file_path, mode) as f:
                content = f.read()
            return content
        except Exception as e:
            print(f"Error leyendo el archivo: {e}")
    def write_file(self, file_path, content, mode='w'):
        try:
            with open(file_path, mode) as f:
                f.write(content)
        except Exception as e:
            print(f"Error escribiendo en el archivo: {e}")

file = FileHandler()
file.write_file('779343360.txt', "28/12/2002")

print(file.read_file('779343360.txt'))
```

```
usuario@usuario-B250M-DS3H: ~  
usuario@usuario-B250M-DS3H:~$ cat 77934336D.txt  
28/12/2002usuario@usuario-B250M-DS3H:~$
```

Actividad 3 de clase:

Amplía la clase anterior para escribir un JSON en el fichero de data.json. Dicho JSON ha de tener dos claves: una para tu DNI (por ejemplo 12345678) y otra para tu fecha de nacimiento (por ejemplo 12/06/76). Usa en Python un dato tipo diccionario para construir el JSON.

```
Python 3.12.3 (main, Sep 11 2024, 14:17:37) [GCC 13.2.0] on line
Type "help", "copyright", "credits" or "license()" for more info

===== RESTART: /home/usuario/Actividad3.py =====
JSON escrito correctamente.
{'DNI': '779343360', 'fecha_de_nacimiento': '28/12/02'}

import json

class JSONFileHandler:
    def read_json(self, file_path):
        try:
            with open(file_path, 'r') as f:
                return json.load(f)
        except Exception as e:
            print("Error leyendo JSON: ", e)

    def write_json(self, file_path, data):
        try:
            with open(file_path, 'w') as f:
                json.dump(data, f, indent=4)
            print("JSON escrito correctamente.")
        except Exception as e:
            print("Error escribiendo JSON: ", e)

json_handler = JSONFileHandler()

data_to_write = {
    "DNI": "779343360",
    "fecha_de_nacimiento": "28/12/02"
}

json_handler.write_json('data.json', data_to_write)

data = json_handler.read_json('data.json')
print(data)
```

Actividad 4 de clase:

Extiende la clase anterior para construir un método inverso, que pase de formato json a formato csv. Adjunta tres capturas de pantalla:

- (1) La clásica con división a la izquierda (traza) y a la derecha (código).
- (2) Contenido del fichero JSON
- (3) Contenido del fichero CSV

- (1) La clásica con división a la izquierda (traza) y a la derecha (código).

```
Python 3.12.3 (main, Sep 11 2024, 14:17:37) [GCC 13.2.0] on linux64
Type "help", "copyright", "credits" or "license()" for more

===== RESTART: /home/usuario/Actividad4.py
JSON escrito correctamente.
JSON convertido a CSV correctamente en data.csv.

import json
import csv

class JSONFileHandler:
    def read_json(self, file_path):
        try:
            with open(file_path, 'r') as f:
                return json.load(f)
        except Exception as e:
            print(f"Error leyendo JSON: {e}")

    def write_json(self, file_path, data):
        try:
            with open(file_path, 'w') as f:
                json.dump(data, f, indent=4)
            print("JSON escrito correctamente.")
        except Exception as e:
            print(f"Error escribiendo JSON: {e}")

    def json_to_csv(self, json_file_path, csv_file_path):
        try:
            data = self.read_json(json_file_path)

            if data:
                with open(csv_file_path, 'w', newline='') as csv_file:
                    writer = csv.writer(csv_file)

                    writer.writerow(data.keys())

                    writer.writerow(data.values())

                print(f"JSON convertido a CSV correctamente en {csv_file_path}.")
            except Exception as e:
                print(f"Error convirtiendo JSON a CSV: {e}")

json_handler = JSONFileHandler()

data_to_write = {
    "DNI": "77934336D",
    "fecha_de_nacimiento": "28/12/02"
}

json_handler.write_json('data.json', data_to_write)
json_handler.json_to_csv('data.json', 'data.csv')
```

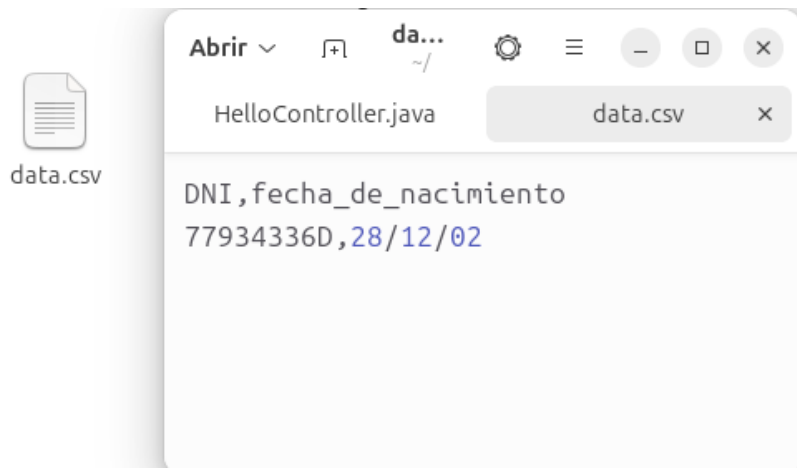
(2) Contenido del fichero JSON



The screenshot shows a file editor window with three tabs: 'elloController.j', 'data.csv', and 'data.json'. The 'data.json' tab is active, displaying the following JSON content:

```
{
  "DNI": "77934336D",
  "fecha_de_nacimiento":
"28/12/02"
}
```

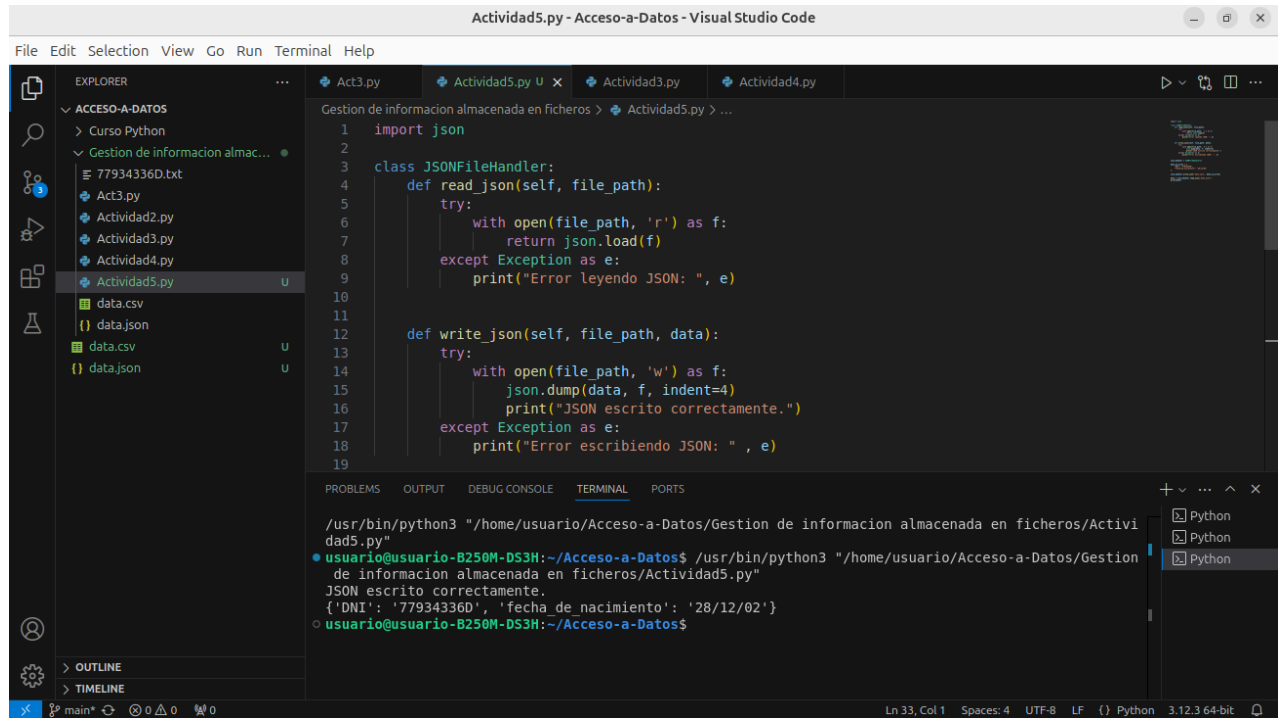
(3) Contenido del fichero CSV



Actividad 5 de clase:

con estas instrucciones, ejecuta la actividad 3 de clase con Visual Studio

Code y proporciona una captura de pantalla, en la que se vea el código de la actividad 3 de clase y la traza de ejecución. Se adjunta captura de pantalla a modo de ejemplo.



The screenshot shows the Visual Studio Code interface with the file 'Actividad5.py' open. The code defines a class 'JSONFileHandler' with two methods: 'read_json' and 'write_json'. The 'read_json' method attempts to load a JSON file, and the 'write_json' method attempts to save a dictionary to a JSON file. Both methods use try-except blocks to handle exceptions.

```
1 import json
2
3 class JSONFileHandler:
4     def read_json(self, file_path):
5         try:
6             with open(file_path, 'r') as f:
7                 return json.load(f)
8         except Exception as e:
9             print("Error leyendo JSON: ", e)
10
11
12     def write_json(self, file_path, data):
13         try:
14             with open(file_path, 'w') as f:
15                 json.dump(data, f, indent=4)
16                 print("JSON escrito correctamente.")
17         except Exception as e:
18             print("Error escribiendo JSON: ", e)
19
```

The terminal output shows the execution of the script, which successfully writes a JSON file. The output is as follows:

```
/usr/bin/python3 "/home/usuario/Acceso-a-Datos/Gestion de informacion almacenada en ficheros/Actividad5.py"
• usuario@usuario-B250M-DS3H:~/Acceso-a-Datos$ /usr/bin/python3 "/home/usuario/Acceso-a-Datos/Gestion de informacion almacenada en ficheros/Actividad5.py"
JSON escrito correctamente.
{'DNI': '77934336D', 'fecha de nacimiento': '28/12/02'}
o usuario@usuario-B250M-DS3H:~/Acceso-a-Datos$
```