

# Explicación de solución punto 2

Por: Sebastian Carmona Estrada

Fecha: 06/02/2024

## Prueba: MANIPULACIÓN DE DATOS

“Cargar un data set, del archivo Excel Master Data, únicamente las siguientes columnas: ➤ Nombre visible Agente ➤ AGENTE (OFEI) ➤ CENTRAL (dDEC, dSEGDES, dPRU...) ➤ Tipo de central (Hidro, Termo, Filo, Menor) ➤ Seleccionar los registros que pertenecen al agente EMGESA ó EMGESA S.A. y adicionalmente que el Tipo de Central sea ‘H’ o ‘T’. ➤ Cargar el archivo dDEC1204.TXT que viene por Central. ➤ Realizar el merge de los dos data sets por Central. ➤ Calcular la suma horizontal de todas las horas para cada planta. ➤ Seleccionar solamente los registros de las plantas cuya suma horizontal sea mayor que cero. ➤ Los resultados deben ser entregados en un dataset. ➤ Enviar junto con la tabla resultante el código utilizado. ➤ Explicar el paso a paso en un archivo de texto (.doc o .pdf).”

## Solución:

```
6  import pdb
7  import numpy as np
8  import pandas as pd
9
10 def excel_parser(file_path):
11     columns = ['Nombre visible Agente', 'AGENTE (OFEI)', 'CENTRAL (OFEI)', 'Tipo de central (Hidro, Termo, Filo, Menor)']
12
13     ds = pd.read_excel(file_path, sheet_name="Master Data Oficial", usecols=columns)
14     ds = ds[(ds['Nombre visible Agente'] == 'EMGESA') & ((ds['Tipo de central (Hidro, Termo, Filo, Menor)'] == 'H') | (ds['Tipo de central (Hidro, Termo, Filo, Menor)'] == 'T')) ]
15
16     return ds
17
18
19 def txt_parser(file_path):
20     columns = ['Hora_{h}'.format(h=h) for h in np.arange(1, 25)]
21     columns.insert(0, 'CENTRAL (OFEI)')
22
23     ds = pd.read_csv(file_path, encoding='ISO-8859-1', header=None)
24     ds.columns = columns
25
26     return ds
27
28 def merge_datasets(ds1, ds2):
29     ds = pd.merge(ds1, ds2, on='CENTRAL (OFEI)')
30
31     return ds
32
33 def sum_h(ds):
34     ds['Total'] = ds.iloc[:, -24:].sum(axis=1)
35     ds = ds[ds['Total'] > 0]
36     return ds
37
38
39 def main():
40     ds1 = excel_parser(r"punto_2\Datos Maestros VF.xlsx")
41     ds2 = txt_parser(r"punto_2\dDEC1204.TXT")
42     ds_merged = merge_datasets(ds1=ds1, ds2=ds2)
43     ds_sumado = sum_h(ds_merged)
44
45     ds_sumado.to_excel(r"punto_2\Resultado.xlsx")
46     # pdb.set_trace()
47
48
49
50 if __name__ == "__main__":
51     main()
```

## Explicación:

### 1. Función `excel_parser(file_path)`:

- Esta función se encarga de leer el archivo de Excel especificado en la ruta `file_path`.
- Utiliza Pandas para cargar el archivo Excel en un DataFrame (`ds`).
- La selección de columnas se realiza utilizando una lista predefinida de nombres de columnas (`'columns'`), que incluye el nombre del agente, la planta y otros detalles relevantes.
- Luego, filtra las filas para seleccionar solo aquellas donde el nombre del agente es "EMGESA" y el tipo de central es hidroeléctrica o térmica.
- Devuelve el DataFrame resultante.

### 2. Función `txt_parser(file_path)`:

- Esta función procesa un archivo de texto para crear un DataFrame.
- Genera dinámicamente una lista de nombres de columnas basada en las horas del día.
- Utiliza Pandas para leer el archivo de texto, especificando la codificación y sin considerar la primera fila como encabezado.
- Devuelve el DataFrame resultante.

### 3. Función `merge_datasets(ds1, ds2)`:

- Esta función combina dos DataFrames (`"ds"` y `"ds2"`) basándose en la columna común `'CENTRAL (OFEI)'`.
- Utiliza la función `pd.merge()` de Pandas para realizar la combinación.
- Devuelve el DataFrame resultante después de la combinación.

### 4. Función `sum_h(ds)`:

- Esta función calcula la suma de las horas en el DataFrame recibido como entrada (`'ds'`).
- Crea una nueva columna llamada `'Total'` que almacena la suma de las horas para cada fila.
- Filtra las filas donde la suma de las horas es mayor que cero, es decir, donde se registraron actividades.
- Devuelve el DataFrame resultante.

### 5. Función `main()`:

- En esta función principal, se realiza el flujo principal del programa.

- b. Se llama a las funciones ``excel_parser()`` y ``txt_parser()`` para cargar los datos de los archivos Excel y de texto, respectivamente.
- c. Luego, se llama a ``merge_datasets()`` para combinar los conjuntos de datos.
- d. Después, se llama a ``sum_h()`` para sumar las horas y filtrar los resultados.
- e. Finalmente, se guarda el DataFrame resultante en un nuevo archivo de Excel.

Este código proporciona una solución eficiente para procesar, combinar y resumir datos de diferentes fuentes, siguiendo un flujo claro y organizado.