

✎ Import important packages

```
import pandas as pd
import glob
import os
```

✎ Upload files

```
ruta = 'C:/Users/ingfe/Downloads/Temps'
#C:/Users/ingfe/OneDrive/Escritorio/Data analysis/Proyecto Data
```

✎ Get all sorted Tmin and Tmax files

```
archivos_tmin = sorted(glob.glob(os.path.join(ruta, '*Tmin.csv')))
archivos_tmax = sorted(glob.glob(os.path.join(ruta, '*Tmax.csv')))
```

✎ Validate that there are complete pairs

```
assert len(archivos_tmin) == len(archivos_tmax), "Faltan archivos Tmin o Tmax"
```

✎ Expected columns in each file

```
columnas_clave = ['Lon', 'Lat', 'Clave', 'Edo', 'Est']
col_tmin = 'Tmin'
col_tmax = 'Tmax'
```

✎ List for storing monthly DataFrames

```
df_mensuales = []

for archivo_min, archivo_max in zip(archivos_tmin, archivos_tmax):
    #Read and clean each file
    df_min = pd.read_csv(archivo_min, encoding='latin1')
    df_max = pd.read_csv(archivo_max, encoding='latin1')

    #Remove empty columns
    df_min = df_min.loc[:, ~df_min.columns.str.contains('^Unnamed')]
    df_max = df_max.loc[:, ~df_max.columns.str.contains('^Unnamed')]

    # Validate expected columns
    if not all(col in df_min.columns for col in columnas_clave + [col_tmin]):
        raise ValueError(f"Columnas faltantes en {archivo_min}")
    if not all(col in df_max.columns for col in columnas_clave + [col_tmax]):
        raise ValueError(f"Columnas faltantes en {archivo_max}")

    # Extract year and month from filename
    nombre = os.path.basename(archivo_min)
    fecha = nombre[:6] # '202401' for example
    año, mes = int(fecha[:4]), int(fecha[4:6])

    # Merge by key columns
    df_mes = pd.merge(df_min, df_max, on=columnas_clave, how='inner')

    # Add year and month
    df_mes['Año'] = año
    df_mes['Mes'] = mes

    df_mensuales.append(df_mes)
```

Concatenate all months

```
df_temps = pd.concat(df_mensuales, ignore_index=True).sort_values(by=['Año', 'Mes'])
```

```
df_temps.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 27639 entries, 0 to 27638
Data columns (total 9 columns):
 #   Column  Non-Null Count  Dtype  
---  --
 0   Lon     27639 non-null  float64
 1   Lat     27639 non-null  float64
 2   Clave   27639 non-null  object  
 3   Edo     27639 non-null  object  
 4   Est     27635 non-null  object  
 5   Tmin    27639 non-null  float64
 6   Tmax    27639 non-null  float64
 7   Año     27639 non-null  int64   
 8   Mes     27639 non-null  int64   
dtypes: float64(4), int64(2), object(3)
memory usage: 1.9+ MB
```

```
df_temps.to_csv('temperature_012024_092025.csv', index=False, encoding='utf-8')
```