

PANDAS:

ANÁLISIS DE DATOS ESTRUCTURADOS Y MANIPULACIÓN

TRABAJO PRÁCTICO FINAL
INTEGRADOR

TEMARIO

1. INTRODUCCIÓN A PANDAS	3
2. ESTRUCTURAS DE DATOS EN PANDAS	6
3. EJERCICIO PRÁCTICO: SERIES EN PANDAS	10
4. EJERCICIO PRÁCTICO: DATAFRAME EN PANDAS	15
5. CONSIDERACIONES FINALES	21

1. INTRODUCCIÓN A PANDAS

¿QUÉ ES Y PARA QUÉ SIRVE?

Pandas es una librería de Python especializada en el manejo y análisis de estructuras de datos. Su nombre proviene de la contracción "Panel Data", que se refiere a conjuntos de datos que incluyen observaciones de múltiples individuos a lo largo de varios períodos de tiempo. Esta herramienta es fundamental para los científicos de datos, ya que permite realizar operaciones complejas de manera sencilla y eficiente.

¿CUÁLES SON SUS CARACTERÍSTICAS PRINCIPALES?

- Definir nuevas estructuras de datos basadas en los arrays de la librería Numpy pero con nuevas funcionalidades.
- Leer y escribir ficheros fácilmente en distintos formatos (CSV, Excel y bases de datos SQL).
- Acceder a los datos mediante índices o nombres para filas y columnas.
- Ofrecer métodos para reordenar, dividir y combinar conjuntos de datos.
- Realizar operaciones de manera eficiente.
- Código abierto.

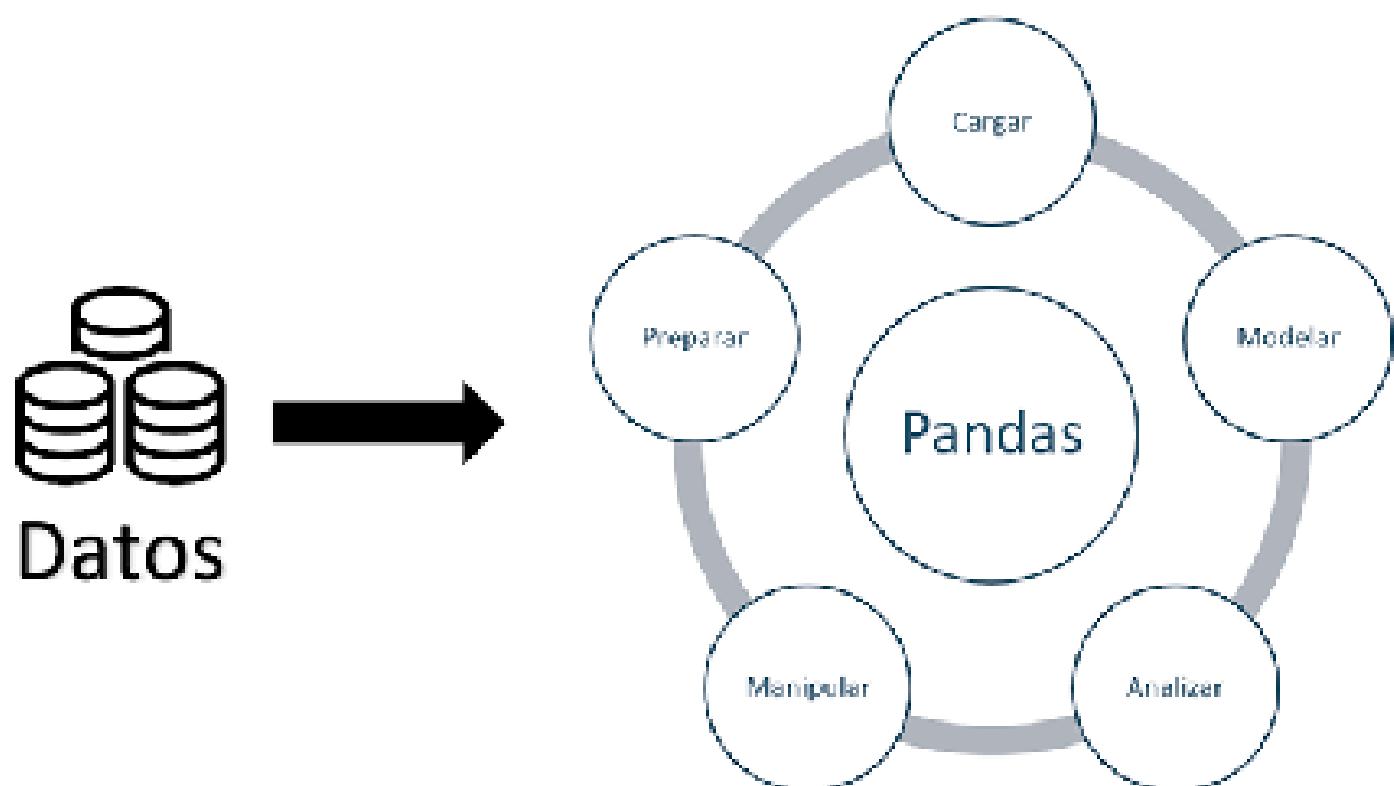


UTILIZACIÓN

Pandas puede ser utilizado para diversas actividades y procesos, entre ellos:

- Limpieza y tratamiento de datos.
- Análisis exploratorio de datos.
- Soporte en actividades de Machine Learning.
- Consultas y queries en bases de datos relacionales.
- Visualizaciones de datos.
- Web scrapping.

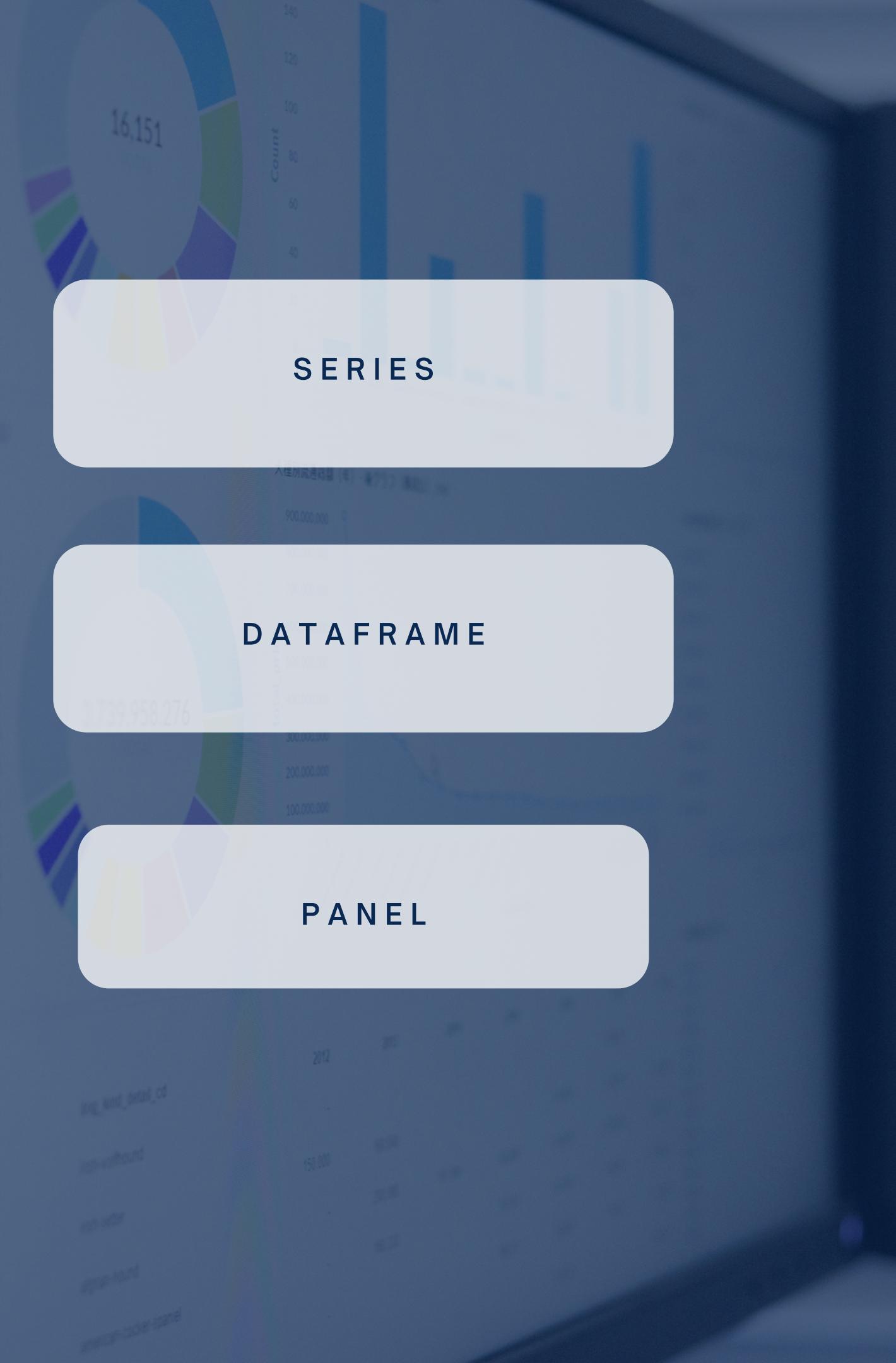
Además, también posee una excelente integración con otras bibliotecas utilizadas en Ciencia de Datos (Numpy, Scikit-Learn, Seaborn, Altair, Matplotlib, Plotly, Scipy, etc.)



VENTAJAS DE UTILIZAR PANDAS

- Facilidad de uso: su sintaxis es intuitiva, lo que permite aprender fácilmente.
- Flexibilidad: permite insertar o eliminar columnas con facilidad dentro de los DataFrames.
- Agrupación de datos: ofrece herramientas para realizar operaciones "split-apply-combine" sobre series de datos.
- Interfaz amigable: la documentación es completa y la interfaz es agradable, lo cual facilita su aprendizaje.
- Amplia comunidad: al ser una herramienta de código abierto, cuánto con una gran comunidad que contribuye a su desarrollo y mejora.

2. ESTRUCTURAS DE DATOS EN PANDAS



TIPOS DE DATOS

Pandas dispone de tres estructuras de datos diferentes: Series (estructura de una sola dimensión), DataFrame (estructura de dos dimensiones) y Panel (estructura de tres dimensiones).

Estas estructuras se construyen a partir de arrays de la librería Numpy, añadiendo nuevas funcionalidades.

SERIES

Son objetos de tipo array unidimensional, con un eje de etiquetas (index), que es responsable de identificar cada registro. Son homogéneas, es decir, sus elementos tienen que ser del mismo tipo, y su tamaño es inmutable (no se puede cambiar, aunque sí su contenido).

El índice asocia un nombre a cada elemento de la serie, a través de la cual se accede al elemento.

Representación de los tres tipos de estructuras de datos utilizadas en Pandas: Series, DataFrame y Panel.

The diagram illustrates the three types of Pandas structures:

- Series (1-D):** A table with a single column labeled "Marks". The index is "Index" and the values are "a" (21), "b" (16), "c" (28), "d" (26), and "e" (43).
- DataFrame (2-D):** A table with two columns labeled "Name" and "Marks". The index is "Index" and the values are "Name": "Rahul", "Deepak", "Varun", "Shivam", "Prateek" and "Marks": 21, 16, 28, 26, 43.
- Panel (3-D):** A stack of two DataFrames. The top DataFrame has the same structure as the one above. The bottom DataFrame has an index "Index" from 0 to 4, and the columns "Name" and "Marks". The values are "Name": "Rahul", "Deepak", "Varun", "Shivam", "Prateek" and "Marks": 21, 16, 28, 26, 43.

DATA FRAME

Son objetos bidimensionales, de tamaño variable. Definen un conjunto de datos estructurado en forma de tabla donde cada columna es un objeto de tipo Series, es decir, todos los datos de una misma columna son del mismo tipo, y las filas son registros que pueden contener datos de distintos tipos.

Un DataFrame contiene dos índices, uno para las filas y otro para las columnas, y se puede acceder a sus elementos mediante los nombres de filas y las columnas.

The diagram shows a DataFrame with the following structure:

	Nombre	Edad	Grado	Correo
1	María	18	Economía	maria@gmail.com
2	Luis	22	Medicina	luis@yahoo.es
3	Carmen	20	Arquitectura	carmen@gmail.com
4	Antonio	21	Economía	antonio@gmail.com

Annotations point to specific parts of the table:

- Nombres Filas:** Points to the index row (1, 2, 3, 4).
- Nombres Columnas:** Points to the header row (Nombre, Edad, Grado, Correo).
- Columnas:** Points to the "Correo" column.
- Filas:** Points to the "Nombre" column.

DataFrame con información sobre los alumnos de un curso. Cada fila corresponde a un alumno y cada columna a una variable.

PANEL

Es una estructura tridimensional que pueden almacenar datos en tres dimensiones. Su uso en las versiones más recientes de Pandas es menor que las de otras estructuras más simples y eficientes como Series y DataFrames.

¿POR QUÉ TRABAJAR CON ESTAS ESTRUCTURAS?

Es posible trabajar con la creación de cada una de estas estructuras utilizando los métodos de Pandas sobre estructuras nativas de Python (cómo listas, arrays y diccionarios). También se puede trabajar con la lectura y escritura de varios tipos de archivos de datos, cómo: CSV, hojas de cálculo en Excel, Parquet, SQL, HTML, JSON, XML, y más.

3. EJERCICIO PRÁCTICO: SERIES EN PANDAS

SERIES

MANIPULACIÓN DE DATOS: SELECCIÓN Y MANIPULACIÓN

```
import pandas as pd

turnos = {
    'Nombre': ['Micaela', 'Miguel', 'Luis', 'Nina'],
    'Servicio': ['Manicuria', 'Pedicuria', 'Peluqueria', 'Depilacion'],
    'Fecha': ['14-07-2026', '18-07-2026', '29-07-2025', '01-08-2025'],
    'Hora': ['15:00', '16:30', '14:50', '18:00'],
    'Localidad' : ['Adrogué', 'Lomas de Zamora', 'Temperley', 'Lanus']
}

serie = pd.Series(turnos)

localidad_series = pd.Series(turnos['Localidad'])
print("Serie de Localidad:")
print(localidad_series)
```

OUTPUT

Serie de Localidad:

0	Adrogué
1	Lomas de Zamora
2	Temperley
3	Lanus

dtype: object

SERIES

MANIPULACIÓN DE DATOS: SELECCIÓN Y MANIPULACIÓN

```
import pandas as pd

turnos = {
    'Nombre': ['Micaela', 'Miguel', 'Luis', 'Nina'],
    'Servicio': ['Manicuria', 'Pedicuria', 'Peluqueria', 'Depilacion'],
    'Fecha': ['14-07-2026', '18-07-2026', '29-07-2025', '01-08-2025'],
    'Hora': ['15:00', '16:30', '14:50', '18:00'],
    'Localidad' : ['Adrogué', 'Lomas de Zamora', 'Temperley', 'Lanus']
}

serie = pd.Series(turnos)

fechas = pd.Series(turnos['Fecha'], name='Fechas')
horas = pd.Series(turnos['Hora'], name='Horas')

print("Fecha del último turno:", fechas[3])
print("Hora del primer turno:", horas[0])
```

OUTPUT

Fecha del último turno: 01-08-2025
Hora del primer turno: 15:00

SERIES

ANÁLISIS DE DATOS: MÉTODOS ESTADÍSTICOS Y DESCRIPTIVOS

```
import pandas as pd

turnos = {
    'Nombre': ['Micaela', 'Miguel', 'Luis', 'Nina'],
    'Servicio': ['Manicuria', 'Pedicuria', 'Peluqueria', 'Depilacion'],
    'Fecha': ['14-07-2026', '18-07-2026', '29-07-2025', '01-08-2025'],
    'Hora': ['15:00', '16:30', '14:50', '18:00'],
    'Localidad' : ['Adrogue', 'Lomas de Zamora', 'Temperley', 'Lanus']
}

serie = pd.Series(turnos)

localidades = pd.Series(turnos['Localidad'])

print("Conteo de localidades:\n", localidades.value_counts())
```

OUTPUT

```
Conteo de localidades:
Adrogue           1
Lomas de Zamora   1
Temperley          1
Lanus              1
Name: count, dtype: int64
```

SERIES

VISUALIZACIÓN DE DATOS: INTEGRACIÓN CON MATPLOTLIB

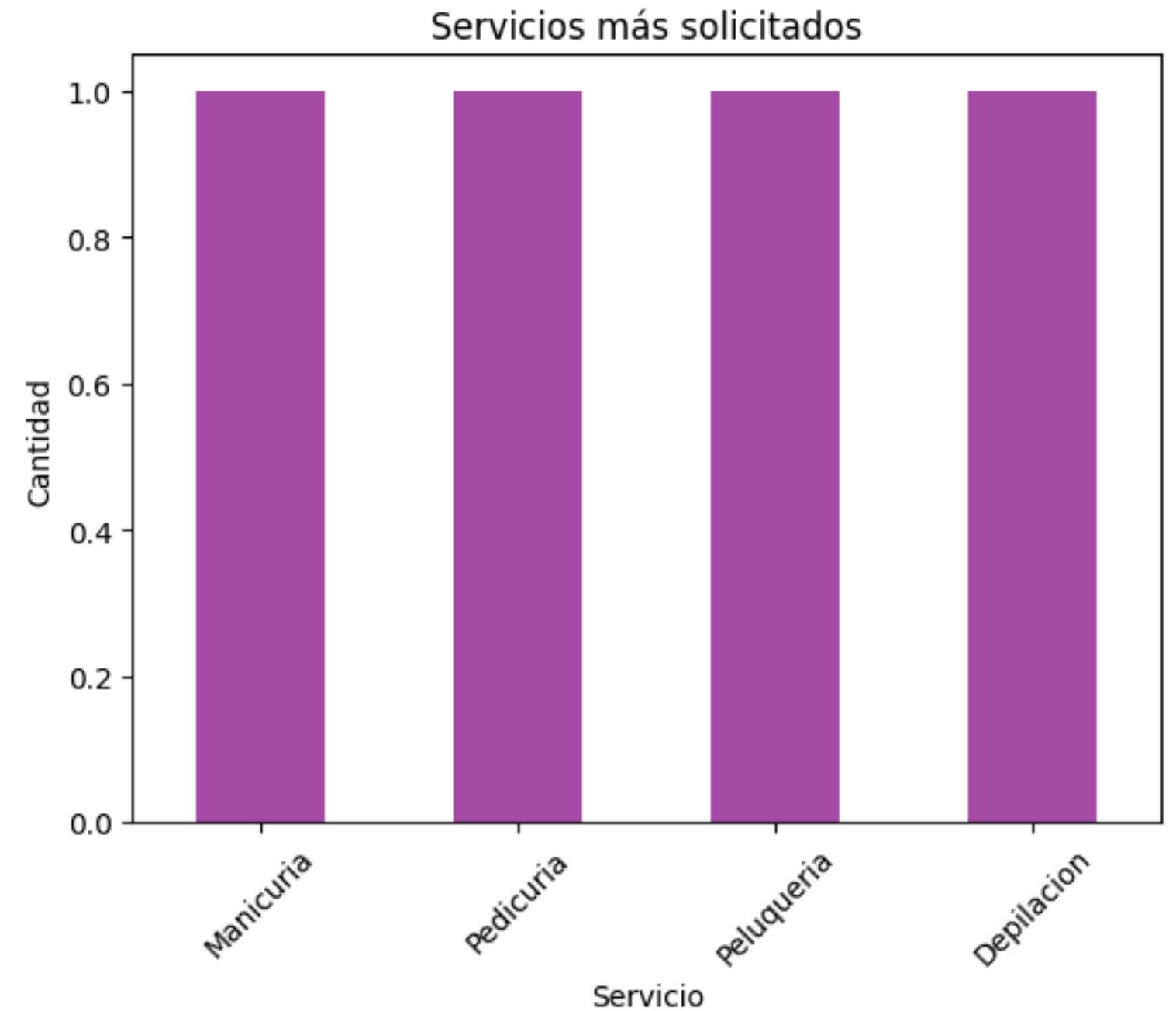
```
import pandas as pd

turnos = {
    'Nombre': ['Micaela', 'Miguel', 'Luis', 'Nina'],
    'Servicio': ['Manicuria', 'Pedicuria', 'Peluqueria', 'Depilacion'],
    'Fecha': ['14-07-2026', '18-07-2026', '29-07-2025', '01-08-2025'],
    'Hora': ['15:00', '16:30', '14:50', '18:00'],
    'Localidad' : ['Adrogué', 'Lomas de Zamora', 'Temperley', 'Lanus']
}

serie = pd.Series(turnos)

import matplotlib.pyplot as plt
# Gráfico de barras para servicios
servicios.value_counts().plot(kind='bar', color='purple', alpha=0.7)
plt.title('Servicios más solicitados')
plt.xlabel('Servicio')
plt.ylabel('Cantidad')
plt.xticks(rotation=45)
plt.show()
```

OUTPUT



4. EJERCICIO PRÁCTICO: DATAFRAME EN PANDAS

DATAFRAME

CREACIÓN DESDE DICCIONARIOS

```
import pandas as pd

turnos = {
    'Nombre': ['Micaela', 'Miguel', 'Luis', 'Nina'],
    'Servicio': ['Manicuria', 'Pedicuria', 'Peluqueria', 'Depilacion'],
    'Fecha': ['14-07-2026', '18-07-2026', '29-07-2025', '01-08-2025'],
    'Hora': ['15:00', '16:30', '14:50', '18:00']
}

turnos = pd.DataFrame(turnos)

print(turnos)
```

OUTPUT

	Nombre	Servicio	Fecha	Hora
0	Micaela	Manicuria	14-07-2026	15:00
1	Miguel	Pedicuria	18-07-2026	16:30
2	Luis	Peluqueria	29-07-2025	14:50
3	Nina	Depilacion	01-08-2025	18:00

DATAFRAME

MANIPULACIÓN DE DATOS: SELECCIÓN Y MODIFICACIÓN

```
import pandas as pd

turnos = {
    'Nombre': ['Micaela', 'Miguel', 'Luis', 'Nina'],
    'Servicio': ['Manicuria', 'Pedicuria', 'Peluqueria', 'Depilacion'],
    'Fecha': ['14-07-2026', '18-07-2026', '29-07-2025', '01-08-2025'],
    'Hora': ['15:00', '16:30', '14:50', '18:00']
}

turnos = pd.DataFrame(turnos)

servicio = turnos['Servicio']
print(servicio)
```

OUTPUT

```
0      Manicuria
1      Pedicuria
2      Peluqueria
3      Depilacion
Name: Servicio, dtype: object
```

DATAFRAME

MANIPULACIÓN DE DATOS: SELECCIÓN Y MODIFICACIÓN

```
import pandas as pd

turnos = [
    'Nombre': ['Micaela', 'Miguel', 'Luis', 'Nina'],
    'Servicio': ['Manicuria', 'Pedicuria', 'Peluqueria', 'Depilacion'],
    'Fecha': ['14-07-2026', '18-07-2026', '29-07-2025', '01-08-2025'],
    'Hora': ['15:00', '16:30', '14:50', '18:00']
]

turnos = pd.DataFrame(turnos)

turnos['Localidad'] = ['Adrogué', 'Lomas de Zamora', 'Temperley', 'Lanus']
print(turnos)
```

OUTPUT

	Nombre	Servicio	Fecha	Hora	Localidad
0	Micaela	Manicuria	14-07-2026	15:00	Adrogué
1	Miguel	Pedicuria	18-07-2026	16:30	Lomas de Zamora
2	Luis	Peluqueria	29-07-2025	14:50	Temperley
3	Nina	Depilacion	01-08-2025	18:00	Lanus

DATAFRAME

ANÁLISIS DE DATOS:
ESTADÍSTICAS
DESCRIPTIVAS

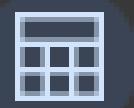
```
import pandas as pd

turnos = {
    'Nombre': ['Micaela', 'Miguel', 'Luis', 'Nina'],
    'Servicio': ['Manicuria', 'Pedicuria', 'Peluqueria', 'Depilacion'],
    'Fecha': ['14-07-2026', '18-07-2026', '29-07-2025', '01-08-2025'],
    'Hora': ['15:00', '16:30', '14:50', '18:00']
}

turnos = pd.DataFrame(turnos)

turnos.describe()
```

OUTPUT

	Nombre	Servicio	Fecha	Hora	
count	4	4	4	4	
unique	4	4	4	4	
top	Micaela	Manicuria	14-07-2026	15:00	
freq	1	1	1	1	

DATAFRAME

VISUALIZACIÓN DE DATOS: INTEGRACIÓN CON MATPLOTLIB

```

import pandas as pd

turnos = {
    'Nombre': ['Micaela', 'Miguel', 'Luis', 'Nina'],
    'Servicio': ['Manicuria', 'Pedicuria', 'Peluqueria', 'Depilacion'],
    'Fecha': ['14-07-2026', '18-07-2026', '29-07-2025', '01-08-2025'],
    'Hora': ['15:00', '16:30', '14:50', '18:00'],
}

turnos = pd.DataFrame(turnos)

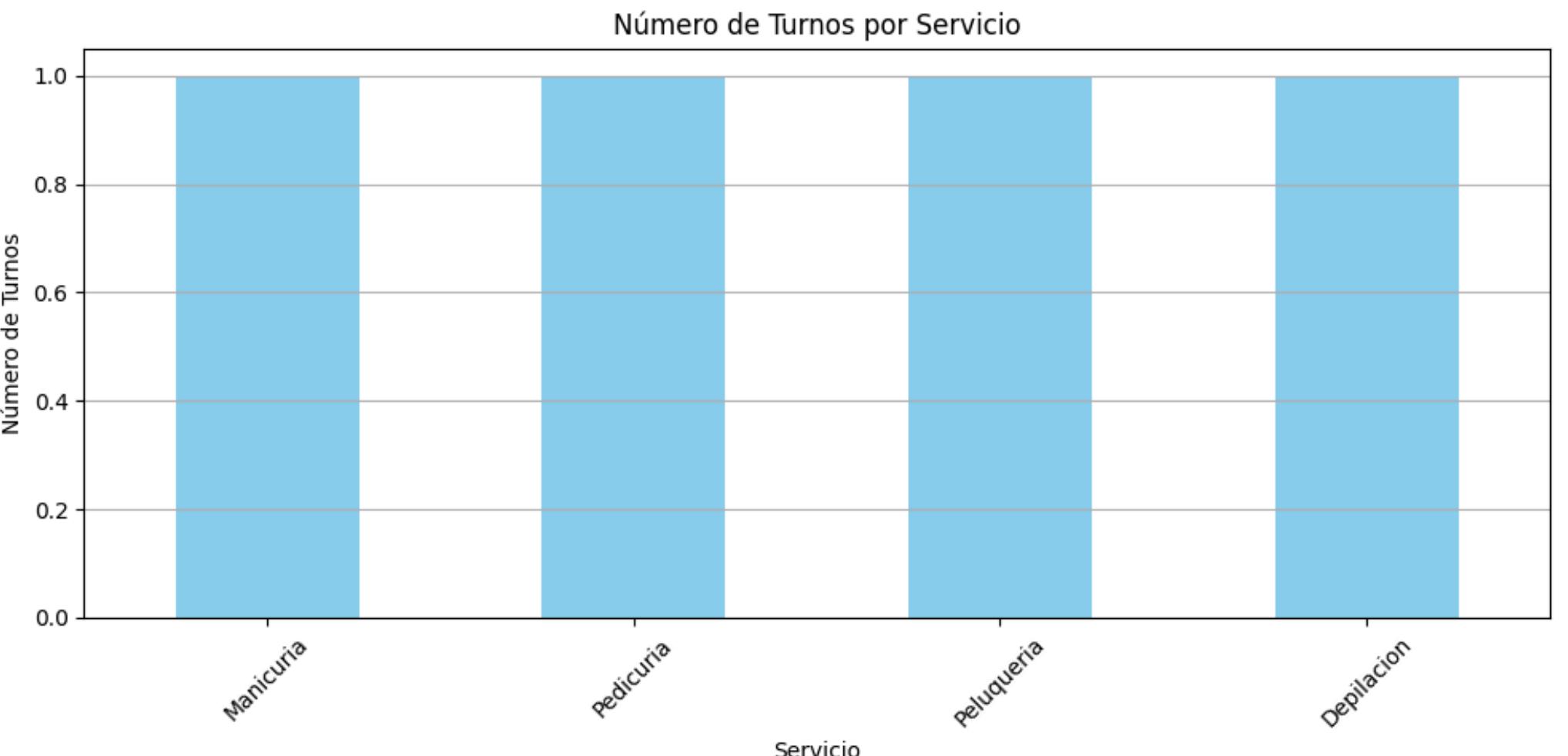
import matplotlib.pyplot as plt
turnos_por_servicio = turnos['Servicio'].value_counts()

plt.figure(figsize=(10, 5))
turnos_por_servicio.plot(kind='bar', color='skyblue')
plt.title('Número de Turnos por Servicio')
plt.xlabel('Servicio')
plt.ylabel('Número de Turnos')
plt.xticks(rotation=45)
plt.grid(axis='y')

plt.tight_layout()
plt.show()

```

OUTPUT



5. CONSIDERACIONES FINALES

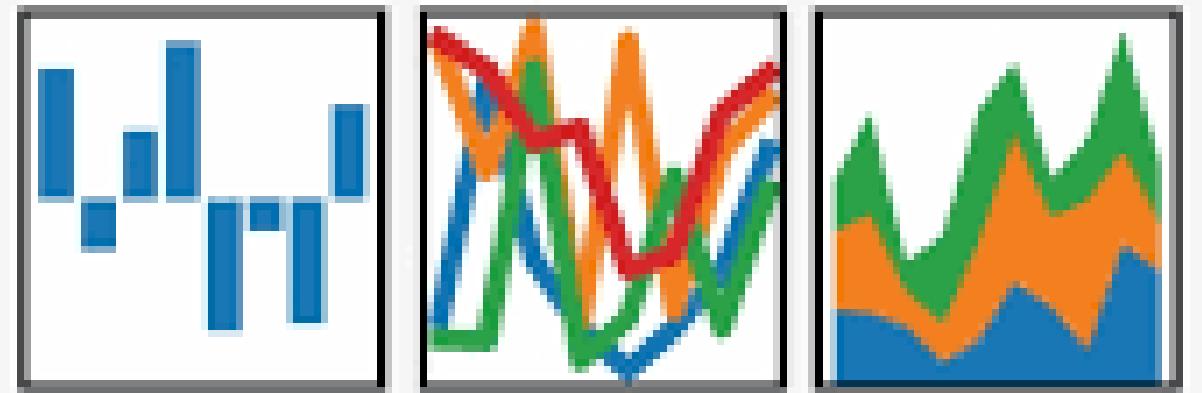
CONCLUSIÓN

Pandas es una herramienta indispensable para el análisis y manipulación de datos estructurados destacándose por su eficiencia, flexibilidad y facilidad de uso.

Sus estructuras de datos (series y DataFrames) permiten trabajar con diversos tipos de datos de manera intuitiva, facilitando el aprendizaje y el uso para analistas de todos los niveles.

La integración con otras bibliotecas de Python potencia sus capacidades, permitiendo análisis complejos y visualizaciones efectivas. Además, su optimización para manejar grandes volúmenes de datos asegura un rendimiento ágil y preciso.

En conclusión, dominar Pandas se traduce en una ventaja competitiva significativa para los profesionales del análisis de datos.



$$y_{it} = \beta' x_{it} + \mu_i + \epsilon_{it}$$

pandas

MUCHAS GRACIAS

INTEGRANTES: MOLINA FLORENCIA - MIGUEL SALAS
PROGRAMACIÓN AVANZADA
COMISIÓN 1 (2025)