NIVEL 4

LIBRERÍAS - PANDAS

Universidad de los Andes

PANDAS

- ¿Qué es Pandas?
  - ✓ Estructuras de datos de Pandas
  - ✓ Lectura de datos en Pandas
  - ✓ Tipos de datos de un DataFrame
  - ✓ Atributos de un DataFrame
  - ✓ Métodos de un DataFrame
  - ✓ Manipulación de DataFrames
  - ✓ Gráficos para explorar los datos de un DataFrame

¿QUÉ ES PANDAS?

Pandas es una librería especializada para el análisis de datos que cuenta con las estructuras de datos que necesitamos para limpiar los datos en bruto y que sean aptos para el análisis (por ejemplo, tablas)

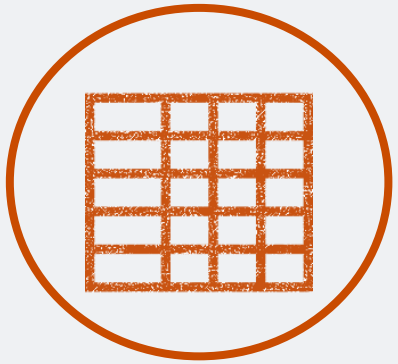
- ✓Dado que pandas lleva a cabo tareas importantes, como alinear datos para su comparación, fusionar conjuntos de datos, gestionar datos perdidos, etc., se ha convertido en una librería muy importante para procesar datos a alto nivel en Python (es decir, hacer análisis estadístico)
- ✓Pandas fue diseñada originalmente para gestionar datos financieros, y como alternativo al uso de hojas de cálculo (es decir, Microsoft Excel)

Pandas nos proporciona las estructuras de datos y funciones necesarias para el análisis de datos

ESTRUCTURAS DE DATOS EN PANDAS

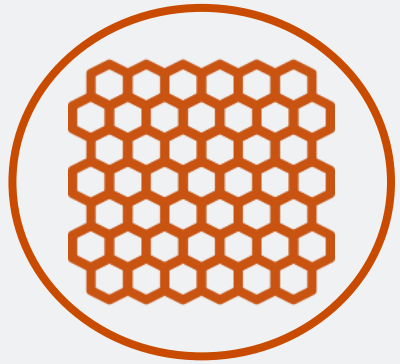
Series (1D)

Similar a la columna de una tabla



DataFrame (2D)

Similar a una tabla con columnas y filas

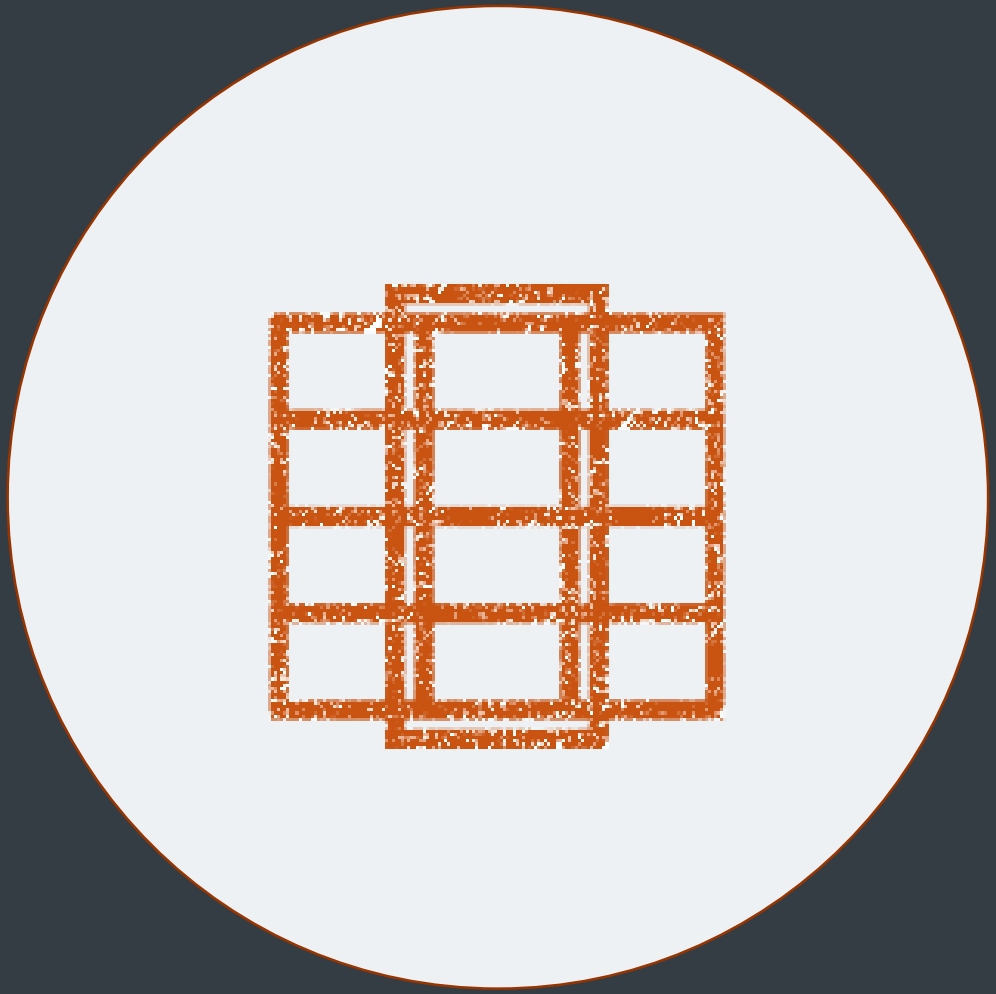


Panel (3D)

Contenedor 3D. No es muy utilizado

TIPOS DE DATOS DE PANDAS

Tipo en Pandas	Tipo equivalente en Python	Descripción
object	string	Es el tipo más general. Será asignado a la columna si la columna tiene valores de varios tipos (números y strings)
int64	int	Valores numéricos
float64	float	Valores numéricos con decimales. Si una columna contiene números y NaNs, pandas lo pondrá por defecto en float64.
datetime64, timedelta[ns]	N/A	Valores destinados a contener datos de tiempo



SERIES EN PANDAS

¿QUÉ ES UNA SERIE?

Temperaturas en Bogotá
19
18
20
14
20
20
20
20
19

- ✓Muchos registros, ordenados según algún criterio
- ✓El mismo tipo de dato para todas las columnas
- ✓Un nombre que describe la serie

¿QUÉ ES UNA SERIE?

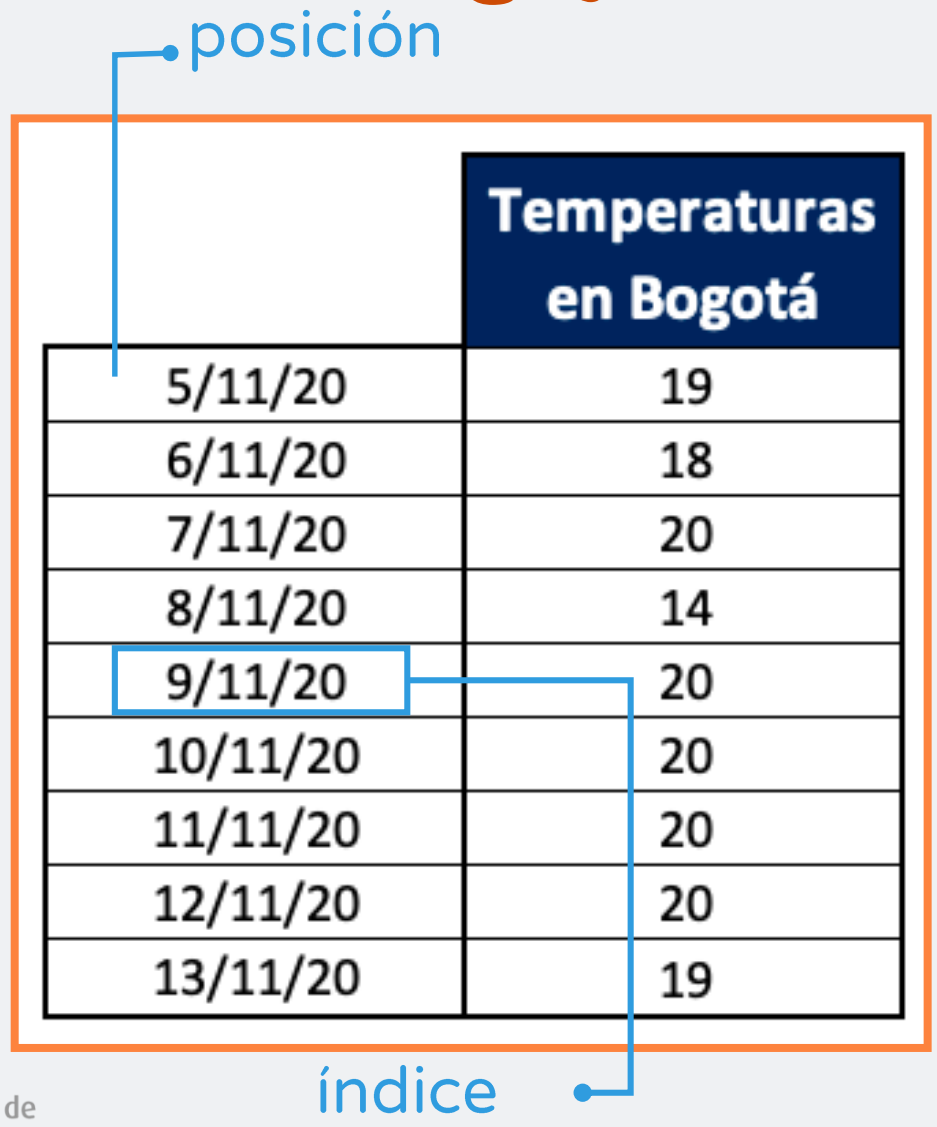
Posición

Índice o etiqueta

	Temperaturas en Bogotá
0	19
1	18
2	20
3	14
4	20
5	20
6	20
7	20
8	19

- ✓Muchos registros, ordenados según algún criterio
- ✓El mismo tipo de dato para todas las columnas
- ✓Un nombre que describe la serie

¿QUÉ ES UNA SERIE?



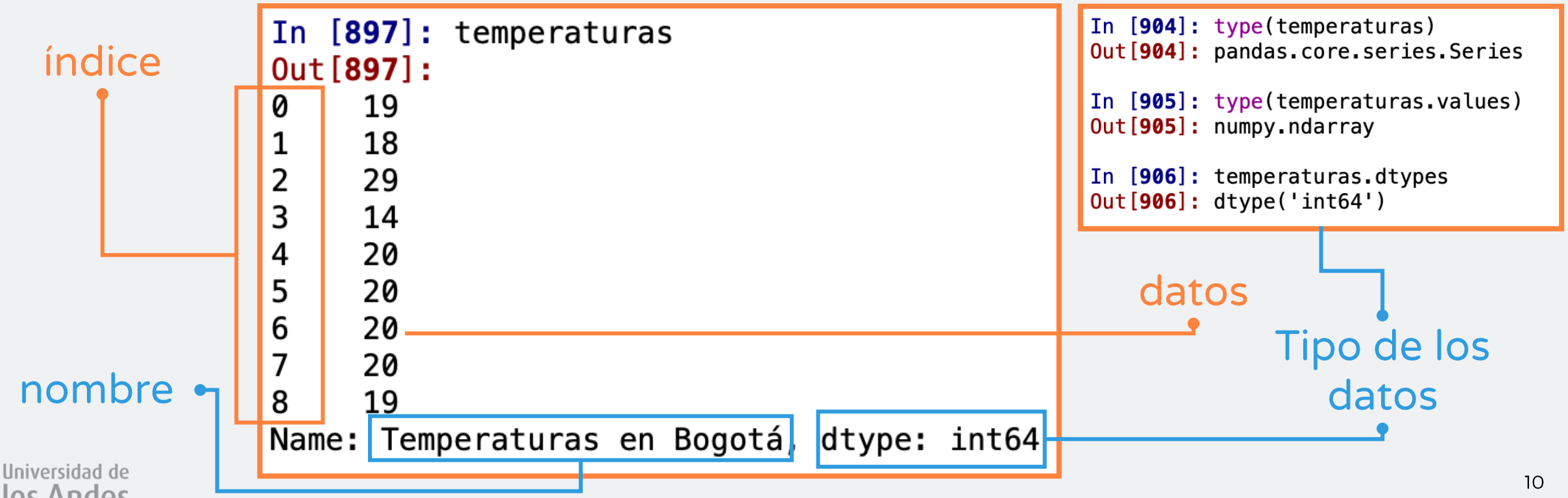
	Temperaturas en Bogotá
5/11/20	19
6/11/20	18
7/11/20	20
8/11/20	14
9/11/20	20
10/11/20	20
11/11/20	20
12/11/20	20
13/11/20	19

- ✓Una posición por serie
- ✓Los índices o etiquetas pueden ser de cualquier tipo

CREACIÓN DE SERIES

```
import pandas as pd
```

```
datos = [19, 18, 29, 14, 20, 20, 20, 20, 19]
temperaturas = pd.Series(datos, name="Temperaturas en Bogotá")
```



CREACIÓN DE SERIES

```
datos = [19, 18, 29, 14, 20, 20, 20, 20, 19]
fechas = ["5/11/20", "6/11/20", "7/11/20", "8/11/20", "9/11/20", "10/11/20",
          "11/11/20", "12/11/20", "13/11/20"]
temperaturas = pd.Series(datos, index=fechas, name="Temperaturas en Bogotá")
```

índice

```
In [909]: temperaturas
Out[909]:
5/11/20     19
6/11/20     18
7/11/20     29
8/11/20     14
9/11/20     20
10/11/20    20
11/11/20    20
12/11/20    20
13/11/20    19
Name: Temperaturas en Bogotá, dtype: int64
```

CREACIÓN DE SERIES

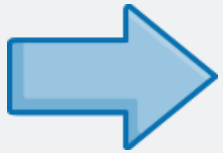
```
parejas = {'5/11/20': 19, '6/11/20': 18, '7/11/20': 29,
           '8/11/20': 14,  '9/11/20': 20, '10/11/20': 20,
           '11/11/20': 20, '12/11/20': 20, '13/11/20': 19}
temperaturas = pd.Series(parejas)
```

```
In [920]: temperaturas
Out[920]:
5/11/20     19
6/11/20     18
7/11/20     29
8/11/20     14
9/11/20     20
10/11/20    20
11/11/20    20
12/11/20    20
13/11/20    19
dtype: int64
```

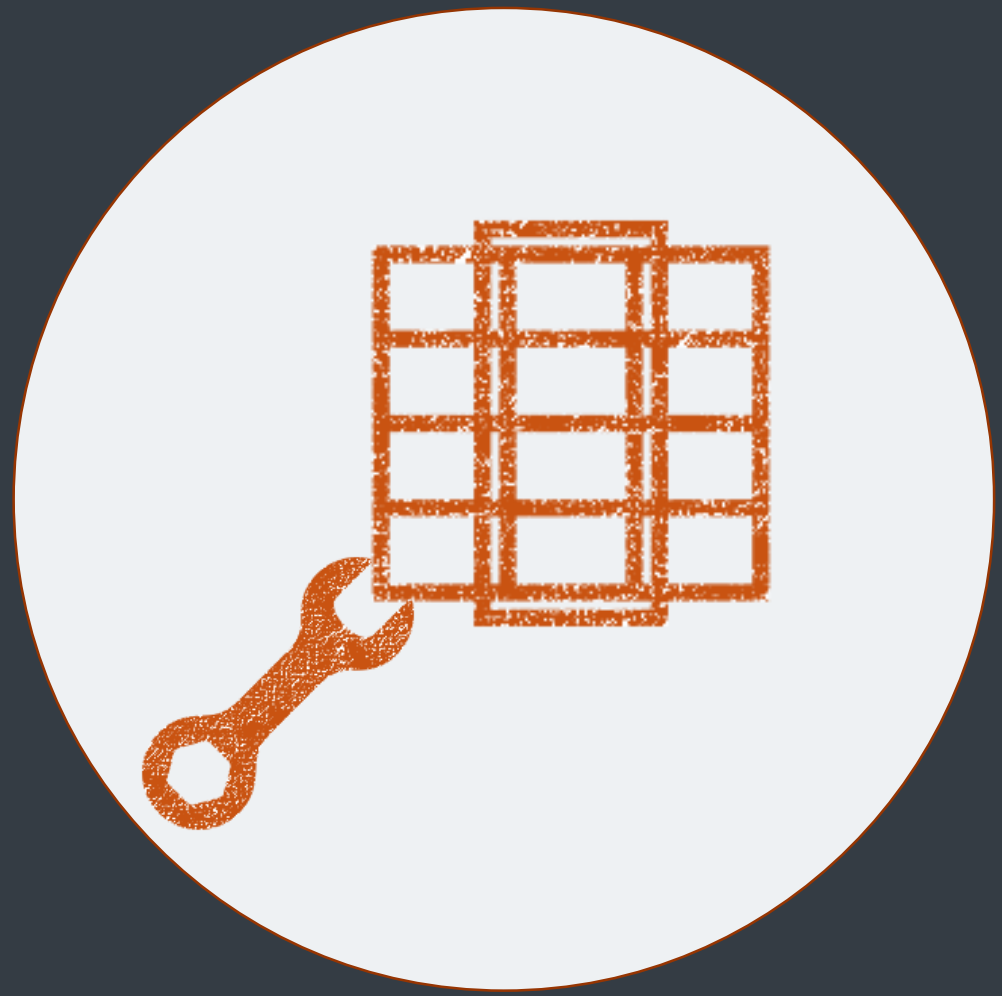
RECONSTRUIR ÍNDICES

```
temperaturas = temperaturas.reset_index(drop = True)
```

```
In [909]: temperaturas
Out[909]:
5/11/20     19
6/11/20     18
7/11/20     29
8/11/20     14
9/11/20     20
10/11/20    20
11/11/20    20
12/11/20    20
13/11/20    19
Name: Temperaturas en Bogotá, dtype: int64
```



```
In [914]: temperaturas
Out[914]:
0    19
1    18
2    29
3    14
4    20
5    20
6    20
7    20
8    19
Name: Temperaturas en Bogotá, dtype: int64
```



OPERACIONES SOBRE SERIES

TENEMOS TRES SERIES

serie_2

```
In [132]: print(serie_2)
1     2
2     4
3     8
4    16
5    32
dtype: int64
```

serie_3

```
In [133]: print(serie_3)
1      1
2      8
3     27
4     64
5    125
dtype: int64
```

serie_4

```
In [134]: print(serie_4)
1        1
2       16
3       81
4      256
5      625
6     1296
dtype: int64
```

ÍNDICES VS POSICIONES

serie_2

```
In [132]: print(serie_2)
1     2
2     4
3     8
4    16
5    32
dtype: int64
```

serie_3

```
In [133]: print(serie_3)
1     1
2     8
3    27
4    64
5   125
dtype: int64
```

serie_4

```
In [134]: print(serie_4)
1       1
2      16
3      81
4     256
5     625
6    1296
dtype: int64
```

¡Este es el índice y NO las posiciones!

EXTRAER VALORES DE UNA SERIE

serie_2

```
In [132]: print(serie_2)
1     2
2     4
3     8
4    16
5    32
dtype: int64
```

1. Extraer un valor usando el **índice**: get(indice)

```
In [159]: print(serie_2.get(2))
4
```

2. Extraer valores usando los **índices**: loc[inicio: fin]

```
In [157]: print(serie_2.loc[2:4])
2     4
3     8
4    16
```

3. Extraer valores usando las **posiciones**: iloc[inicio: fin]

```
In [158]: print(serie_2.iloc[2:4])
3     8
4    16
```

Series

OTRAS OPERACIONES

values / to_list( )

```
In [136]: serie_2.values
Out[136]: array([ 2,  4,  8, 16, 32])

In [137]: serie_2.to_list()
Out[137]: [2, 4, 8, 16, 32]
```

```
In [139]: numeros = serie_2.to_list()

In [140]: for num in numeros:
     ...:     print(num)
     ...:
2
4
8
16
32
```

copy( )

```
In [141]: copia = serie_2.copy()

In [142]: print(copia)
1     2
2     4
3     8
4    16
5    32
dtype: int64
```

OPERACIONES ESTADÍSTICAS

serie_2

```
In [132]: print(serie_2)
1     2
2     4
3     8
4    16
5    32
```

max( ), min( ), mean( ), std( ), median(), …

```
serie_2.max() --> 32
serie_2.min() --> 2
serie_2.mean() --> 12.4
serie_2.std() --> 12.198360545581526
serie_2.median() --> 8.0
```

MÁS OPERACIONES

Atributos, operaciones aritméticas, conversión, iteración, operaciones binarias, aplicación de funciones, cálculos y estadística descriptiva, ajuste de índices, manejo de datos faltantes, mezcla de series, gráficas, etc...

https://pandas.pydata.org/pandas-docs/stable/reference/series.html