



# **INTRODUCCIÓN A PYTHON**

**CLASE 12**

# Contenido

Introducción al pensamiento computacional.....	4
Introducción.....	5
Librería Pandas.....	5
Dataframe .....	5
Creación de un dataframe desde un diccionario.....	5
Creando un DF desde un archivo CSV o Excel.....	5
[df].iloc .....	6
[df].loc .....	6
Librería Matplotlib .....	7
Importar matplotlib .....	7
Creando un gráfico sencillo .....	7
Personalizando los gráficos.....	8
Personalizar las etiquetas .....	9
tipos de gráficos .....	9
Bibliografía .....	10


“

Este espacio curricular permite fortalecer el espíritu crítico y la actitud creativa del futuro graduado. Lo alienta a integrar los conocimientos previos, recreando la práctica en el campo real, al desarrollar un proyecto propio.

”



**PYTHON**



## Introducción al pensamiento computacional

Avanzamos con el uso de las librerías, en este apartado, vamos a completar los conocimientos de la librería pandas y complementarlo con MatPloylib para mostrar por medio de gráficos los datos obtenidos

# Introducción

Pandas es una librería que nos permite trabajar con los datos, ya lo usamos en clases anteriores generando series. En esta oportunidad, vamos a trabajar con un tipo nuevo de datos.

En complemento con lo anterior, veremos como graficar datos en pantalla usando las librerías Matplot.

## Librería Pandas

Una librería para trabajar con datos

### Dataframe

Es un conjunto de datos que se organiza como tablas, ordenado en filas y columnas. Esto significa que, todos los datos organizados en una fila pertenecen al mismo objeto. Los dataframe, tienen dos índices. Al estar organizado como matrices, tienen uno para las filas y otro para las columnas.

### Creación de un dataframe desde un diccionario

Para crear un DF puede parecer algo complicado, pero podemos ver cómo hacerlo partiendo de un diccionario.

Primero creamos un diccionario, donde el índice corresponde a una columna de la tabla y los datos en cada elemento de las filas

```
dic={"nombre":["Martin","Angela","Nahuel"], "Materia":["IPC","IPC","IPC"], "Notas":[10,10,9]}
```

luego, tomando esto como insumo creamos el df con las palabras reservadas incluidas en pandas.

```
df=pd.DataFrame(dic)
```

para comprobar que se generó correctamente, usando la instrucción print lo mostramos en pantalla.

	nombre	Materia	Notas
0	Martin	IPC	10
1	Angela	IPC	10
2	Nahuel	IPC	9

### Creando un DF desde un archivo CSV o Excel

La librería panda, nos brinda un potencial increíble para leer archivos externos. En este caso podemos hacerlo desde un archivo con extensión xls o desde un csv. Lo importante, es que la extracción es totalmente transparente.

Aquí vemos como importar desde un CSV

```
df=pd.read_csv('ejemplo.csv')
```

Como nota de color, recordemos que con este ejemplo el archivo debe estar en la misma

carpeta. En caso contrario, debemos colocar el path al archivo.

## [df].iloc

El acceso es tomando como referencia la ubicación, del dato en la tabla. Dicha ubicación se representa por las filas y las columnas.

- **df.iloc[filas, columnas]** : devuelve el valor de la celda
- **df.iloc[i]** : devuelve los datos de toda una fila

```
print (df.iloc[0])
```

Nombre	Martin
materia	IPC
nota	10

## [df].loc

Devuelve los datos de una fila entera, también permite devolver los datos de más de una fila.

```
print (df.loc[[0,1]])
```

	Nombre	materia	nota
0	Martin	IPC	10
1	Angela	IPC	10

# Librería Matplotlib

Es una librería de código abierto, que se puede usar en los programas de Python para generar fácil y rápidamente un gráfico.

Importar Matplot

Como todas las librerías, se puede descargar usando la instrucción pip

```
pip install matplotlib
```

## Importar matplotlib

No hay grandes cambios a lo que veníamos haciendo, matplotlib como toda librería se agrega a nuestro código usando **import**

```
import matplotlib.pyplot as mp
```

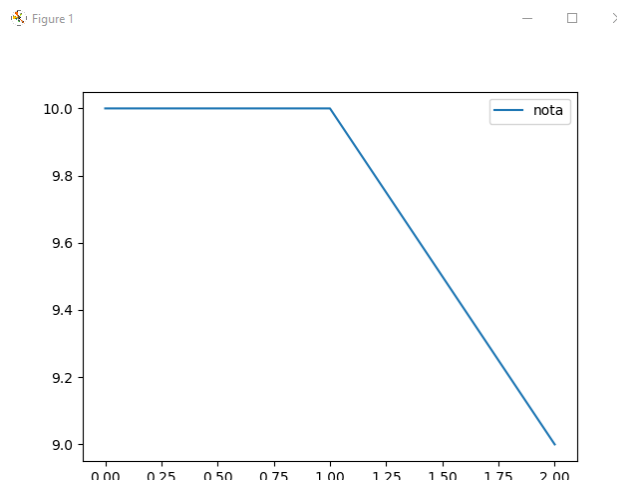
## Creando un gráfico sencillo

El Código, se puede ver a continuación

```
import pandas as pd
import matplotlib.pyplot as mp

df=pd.read_csv('ejemplo.csv')
print (df)

df.plot()
mp.show()
```



Claramente, no siempre el gráfico generado automáticamente tiene sentido. Vamos a personalizar un poco el código para mostrar un grafico con un poco más de sentido.

Para ello, agregamos al plot tres palabras

1. Kind: el tipo de grafico
2. X= valor del eje X
3. Y= valor del eje Y

En el siguiente código, creamos un grafico de barras con el nombre y las notas tomando el siguiente dataframe

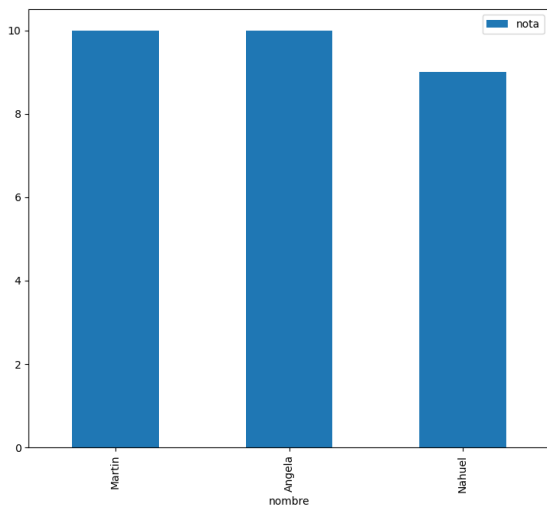
	nombre	materia	nota
0	Martin	IPC	10
1	Angela	IPC	10
2	Nahuel	IPC	9

Considerando el siguiente Código, veremos a continuación el grafico que genera.

```
import pandas as pd
import matplotlib.pyplot as mp

df=pd.read_csv('ejemplo.csv')
print (df)

df.plot(kind='bar',x="nombre",y='nota')
mp.show()
```



Como se puede observar, el grafico ya tiene un poco más de sentido.

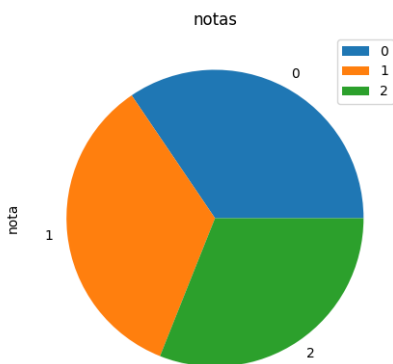
Gráfico de torta

Cambiamos tipo de gráfico, siguiendo el Código que se muestra a continuación

```
import pandas as pd
import matplotlib.pyplot as mp

df=pd.read_csv('ejemplo.csv')
print (df)

df.plot.pie(y='nota',figsize=(5,5))
mp.show()
```



## Personalizando los gráficos

Se puede modificar los gráficos para que la experiencia sea mas positiva.

Agregando títulos

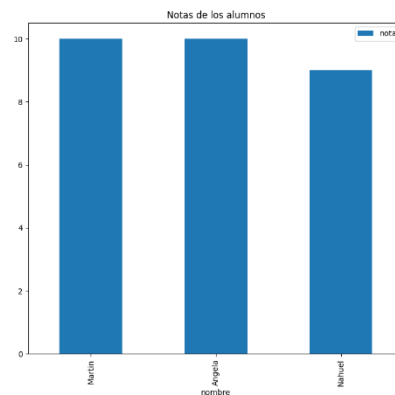
```
df.plot(kind='bar',x='nombre',y='nota',title='Notas de los alumnos')
```

## Importando con index

Podemos crear un index personalizado cuando importantes desde el archivo externo

```
df=pd.read_csv('ejemplo.csv',
index_col='nombre')

df.plot(kind='bar',y='nota',title='Notas de los alumnos')
```



cambiando el color

se puede modificar los colores de las barras, y también de las series



```
df.plot(kind='hist',y='nota',title='Notas de los alumnos',color=['red'])
```



## tipos de gráficos

- **'line'**: gráfico de líneas (predeterminado)
- **'bar'**: diagrama de barra vertical
- **'barh'**: diagrama de barra horizontal
- **'hist'**: histograma
- **'box'**: diagrama de caja
- **'kde'**: Diagrama de estimación de la densidad del kernel
- **'area'**: gráfico de área
- **'pie'**: diagrama circular
- **'scatter'**: gráfico de dispersión
- **'hexbin'**: diagrama de hexbin

## Personalizar las etiquetas

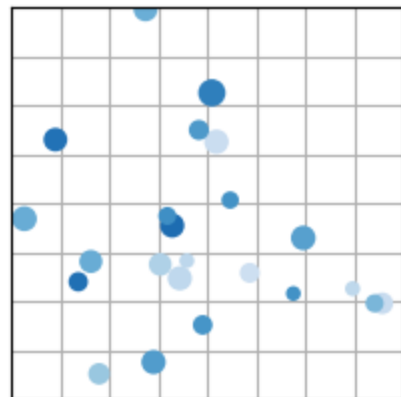
Podemos cambiar las Etiquetas por un dato

- La nota como etiqueta

```
df.plot(kind='pie',y='nota',title='Notas de los alumnos',labels=df['nota'])
```

- Las materias como etiqueta

```
df.plot(kind='pie',y='nota',title='Notas de los alumnos',labels=df['materia'])
```



# Bibliografía

Python para Todos” de Raúl González Duque. (<http://mundogeek.net/tutorial-python/>).

Miller, Curtis. (2016). Hands-On: Data Analysis with Numpy and pandas. (1ed).Packt>. Birmingham

recursos

John Hunter, Darren Dale, Eric Firing, Michael Droettboom. (2012).  
[https://matplotlib.org/3.1.0/gallery/style\\_sheets/style\\_sheets\\_reference.html](https://matplotlib.org/3.1.0/gallery/style_sheets/style_sheets_reference.html).  
matplotlib.or,. consultado en 2022

Pandas.pydata.org. (2022). [https://pandas.pydata.org/pandas-docs/stable/user\\_guide/visualization.html](https://pandas.pydata.org/pandas-docs/stable/user_guide/visualization.html). Consultado en 2022

