

Archivos

- son estructura de datos persistentes (o sea que perdura más allá de la vida de quien las genera)
- son estructuras de datos empleados para compartir información entre usuarios, procesos, equipos
- debemos recordar que estamos haciendo programas en una arquitectura Von Neumann. Por lo tanto, para que los datos puedan ser procesados primero se deben alojar en la MI y para sacar los datos y los resultados también deben pasar por dicha MI
- generalmente los archivos tienen grandes cantidades de datos, por esto copiar un archivo completo en MI puede ser una tarea costosa y/o imposible. Una solución será copiar los datos por partes para que a medida que necesitemos información la vamos subiendo a la MI y a medida que la modifiquemos resguardaremos una copia en el archivo
- el SO reserva una zona de memoria y establece un canal de comunicación entre el archivo y el programa. Estos son conocidos como **Buffers** (mini espacios de intercambio entre un programa y un archivo). El SO implementa un sistema de tráfico de intercambio, lectura y escritura en bloque (si un programa necesita un dato específico, trae todo el bloque de información donde se encuentra ese dato, debido a que si trae solo el dato específico resulta un costo operativo alto. También es probable que se necesite los datos cercanos a este)
- abrir un archivo es decirle al SO que defina un buffer. Este va a tener el nombre del archivo, dónde se ubica en las estructura de directorios, qué haremos con él, etc
- un mismo archivo puede abrir simultáneamente los archivos necesarios. Para diferenciarlos se le asigna un alias (nombre interno) a cada archivo para referenciarlos dentro del programa

Primitivas Generales

- abrimos un archivo, en Python, con la **función open()**. Ésta solicita al SO que establezca el vínculo del trabajo y nos devuelva la dirección del inicio del buffer del archivo. Guardamos la dirección en una variable (nombre interno del archivo desde ese momento, dicho nombre interno es una etiqueta que apunta al inicio del buffer. Una vez abierto solo referenciamos al archivo por su nombre interno o alias)

SINTAXIS

```
alias = open(nombre_completo_archivo, modo)
```

EJEMPLO

```
archivo = open("texto.txt")
```

- nombre y extensión con el que fue guardado
- localización completa o relativa en los directorios (path)
- el modo es por default de lectura (o sea 'r') pero puede ser también w

- cuando terminamos de usar el archivo, **ES IMPORTANTE** realizar la **operación de cierre** (operación que libera el espacio de buffer y completa el proceso de grabado si hay algo pendiente), usando el **método close()**.

El SO trae la información, por bloques sin preguntarnos y tampoco se molesta en salir a grabar un dato cuando lo solicitamos. Junta varios encargos de grabado antes de realizar la tarea. El cierre del archivo obliga a grabar todo lo que quedó pendiente

Modos de Apertura

Modo	Lee	Escribe	Existencia de Archivo	Posición Inicial de Lectura	Posición Inicial de Escritura
r	✓	-	Si el archivo no existe, da error.	Inicio	-
w	-	✓	Si el archivo no existe, lo crea. Si existe, le borra todo el contenido.	-	Inicio
a	-	✓	Si el archivo no existe, lo crea.	-	Final
r+	✓	✓	Si el archivo no existe, da error.	Inicio	Depende
w+	✓	✓	Si el archivo no existe, lo crea. Si existe, le borra todo el contenido.	Inicio	Depende

1. Modo r

- r = read = leer
- se asigna por default, aunque también se puede aclarar en el código

```
archivo = open("archivo.txt", "r")
```

- tres métodos de lectura
 - **read** → lee todo el archivo junto y lo devuelve en un string
 - **readlines** → lee cada línea y las devuelve como una lista. Cada línea tiene \n que indica el cambio de línea
 - **readline** → lee línea por línea cada vez que se llama la función. Cada línea contiene el caracter especial \n pero no de forma explícita, sino implícita (o sea se imprime con un espacio de nueva línea pero no vemos el carácter)

- modo r+

```
archivo = open("archivo.txt", "r+")
```

- tenemos tres comportamientos:

Comportamiento 1: Sólo Lectura

Si sólo leemos, el modo r+ se va a comportar como r

```
main.py
1 archivo = open("archivo.txt", "r+")
2 lineas = archivo.readlines()
3 print(lineas)
4 archivo.close()
```

```
['Este es un texto previo\n', 'Este también\n']
>
```

archivo.txt x notas.txt x arch1Par.txt x +

archivo.txt

```
1 Este es un texto previo
2 Este también
3
```

Comportamiento 2: Sólo Escritura

El modo **r+** cuando sólo escribe, empieza a escribir desde el comienzo del archivo. Es decir, va a pisar, caracter a caracter, el contenido previo de cada línea.



```
main.py
1 archivo = open("archivo.txt", "r+")
2 archivo.write("Texto nuevo que va a pisar todo")
3 archivo.close()
```

archivo.txt

```
1 Texto nuevo que va a pisar todombién
2
```

Comportamiento 3: Lectura y Escritura

- cuando leemos y escribimos usando este modo, va a pasar que independientemente de qué hagamos antes, todo lo que escribamos en el archivo siempre va a ir al final. Esto ocurre porque una vez que leemos el archivo, el puntero de posición con el que escribimos pasa a estar al final
- si le pedimos leer el archivo, siempre nos va a leer el contenido que teníamos antes de abrirlo, debido a que los write se guardan en el momento en el que se cierra el archivo

Ejemplo de lectura y luego escritura:
Como vemos, le pedimos leer sólo 1 línea, e incluso así cuando escribí, escribí al final del archivo.

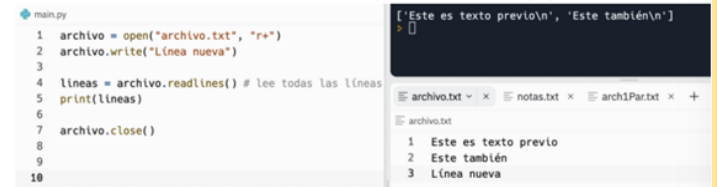


```
main.py
1 archivo = open("archivo.txt", "r+")
2 lineas = archivo.readline() # lee una sola línea
3 print(lineas)
4
5 archivo.write("Línea nueva")
6 archivo.close()
```

archivo.txt

```
1 Este es texto previo
2 Este también
3 Línea nueva
```

Ejemplo de escritura y luego lectura:
Lo mismo: escribimos una línea, que se agregó al final. Pero al momento de leer todas las del archivo, sólo nos devuelve aquellas que estaban en la copia de apertura original, no incluye la que agregamos.



```
main.py
1 archivo = open("archivo.txt", "r+")
2 archivo.write("Línea nueva")
3
4 lineas = archivo.readlines() # lee todas las líneas
5 print(lineas)
6
7 archivo.close()
```

archivo.txt

```
1 Este es texto previo
2 Este también
3 Línea nueva
```

2. Modo w

- w = write = escribir

```
archivo = open("archivo.txt", "w")
```

- dos métodos de lectura
 - **write** → recibe un string y los escribe en el archivo. Si no le agregamos el caracter especial (\n) imprime una palabra detrás de otra

```
archivo = open("archivo.txt", "w")
archivo.write("hola")
archivo.write("chau")
archivo.close()
```

```
archivo.txt
1 holachau
```

- **writelines** → recibe un string o una lista de strings. Si no le agregamos el caracter especial imprime una palabra detrás de otra

```
archivo = open("archivo.txt", "w")
archivo.writelines(["hola", "adios"])
archivo.writelines("chau")
archivo.close()
```

```
archivo.txt
1 holaadioschau
```

- modo w+

```
archivo = open("archivo.txt", "w+")
```

- tiene tres posibles comportamientos:

Comportamiento 1: Sólo Lectura

El modo sólo lectura para `w+` no funciona como uno pensaría, porque **lo primero que hace `w+` es borrar todo el archivo al abrirlo.**

Siempre empezamos con un archivo en blanco al abrir con `w+` (como pasa también con `w`, porque si no existe, lo crea; y si existe, lo vacía).

Este es el estado previo de nuestro archivo:

```
archivo.txt
1 Línea 1
2 Línea 2
3
```

Y este es el resultado de ejecutar el código de la izquierda:

```
main.py
1 archivo = open("archivo.txt", "w+")
2
3 lineas = archivo.readlines() # lee todas las líneas
4 print(lineas)
5
6 archivo.close()
7
8
```

```
[ ]
> []
```

archivo.txt x notas.txt x arch1Par.txt x +

archivo.txt

```
1
```

Comportamiento 2: Sólo Escritura

El estado previo del archivo es el siguiente:

```
archivo.txt
1 Línea 1
2 Línea 2
3 Línea 3
```

```
main.py
1 archivo = open("archivo.txt", "w+")
2
3 archivo.write("Línea nueva")
4
5 archivo.close()
6
7
8
```

```
[ ]
> []
```

archivo.txt x notas.txt x arch1Par.txt x +

archivo.txt

```
1 Línea nueva
```

Al abrirlo, se borra el archivo por completo. Luego, escribimos dentro del archivo vacío:

Comportamiento 3: Lectura y Escritura

Este de nuevo es el punto complicado.

Primero lectura y luego escritura

Como al abrir el archivo, se vacía, siempre tratar de leer con **w+** antes de escribir no va a devolver nada.

main.py

```
1 archivo = open("archivo.txt", "w+")
2
3 lines = archivo.readlines()
4 print(lines)
5
6 archivo.write("Línea nueva")
7
8 archivo.close()
9
10
```

```
[]
> []
```

archivo.txt x notas.txt x

archivo.txt

1 Línea nueva

Primer escritura y luego lectura

Lo mismo, el archivo se borra al abrirlo. Por lo que el resultado es idéntico al caso anterior.

Una forma de lograr leer sobre el archivo que acabamos de escribir sería volver a posicionarnos con el comando **seek(0)** en el inicio del archivo, pero no es algo que vayamos a ver en la materia.

3. Modo a

- a = append = agregar

```
archivo = open("archivo.txt", "a")
```

- el método de escritura es igual al de **w**, aunque se diferencia porque **a SIEMPRE AGREGA EL CONTENIDO AL FINAL DEL ARCHIVO**
- problema de escritura con append
 - ocurre cuando nuestro archivo no tiene una línea vacía extra
 - nuestro indicador de posición en el archivo va a agregar el contenido a lado del anterior (o sea sin salto de línea ambos strings se van a juntar por así decirlo). Para esto tenemos dos soluciones: la primera es poner el carácter especial antes o dentro de escribir el string y la segunda es procurar que todos lo que se agrega al archivo tenga un carácter especial (**\n**) al final, de esta forma vamos a terminar las líneas con un punto y coma

```
archivo = open("archivo.txt", "a")
archivo.writelines(["\nhola", "adios"])
archivo.writelines("chau")
archivo.close()
```

archivo.txt

1 Este es un texto previo

2 Este también

3 holaadioschau

4. Otra forma de abrir archivos

- usando el with open as (para este modo no es necesario usar close())

main.py

```
1 ~ with open("archivo.txt", "r") as archivo:
2     lines = archivo.readlines()
3     print(lines)
4
```

- el código dentro del bloque with mantiene el archivo abierto

Tipos de Archivo

Archivos Planos – Archivos de Texto (txt)

- archivo plano / archivo de texto plano / archivo de texto simple
- es un tipo de archivo informático que almacena datos estructurados pero sin usar ningún formato de codificación o estructura de datos compleja
- suelen llevar la extensión “.txt” al final (ejemplo: archivo.txt)

• Agrega oraciones en arch1.txt

```
vaca_txt=open('arch1.txt','r+')
t = vaca_txt.readline()
print(t)
```

```
t = input('Ingresá un texto con vaca: ')
while (t.lower()).count('vaca')==0:
```

```
    t = input('Ingresá un texto con vaca: ')
vaca_txt.write(t+'\n')
vaca_txt.close()
```

```
vaca_txt=open('arch1.txt')
todas = vaca_txt.readlines()
for linea in todas:
    print(linea.strip('\n'))
vaca_txt.close()
```

Nota:

En este ejemplo tenemos en la misma carpeta del programa un archivo: **arch1.txt** que contiene al momento de correr el programa lo siguiente:

```
La vaca LOCA
VACAYENDO gente al baileee
siga la vaca
vacaciones en faMilia...
es lila la vaca de MILKA
```

si ejecutamos el programa tendremos la siguiente salida

```
>_ Console x Output x +

La vaca LOCA

Ingresá un texto con vaca: señora vaca, señora vaca
La vaca LOCA
VACAYENDO gente al baileee
siga la vaca
vacaciones en faMilia...
es lila la vaca de MILKA
>_
```

Y el archivo **arch1.txt** contendrá ahora lo siguiente:

```
La vaca LOCA
VACAYENDO gente al baileee
siga la vaca
vacaciones en faMilia...
es lila la vaca de MILKA
señora vaca, señora vaca
```

- qué hicimos en el programa?

Código	Efecto
# pruebaarch1.py	
vaca_txt=open('arch1.txt','r+')	Abre arch1.txt para lectura y escritura, con la codificación por defecto del SO y define nombre interno o alias vaca_txt
t=vaca_txt.readline()	Lee desde el archivo la primer línea (todos los caracteres hasta encontrar un \n y se guarda en t
print(t)	Muestra la línea
t=input('Ingresá un texto con vaca: ')	Pide al usuario un nuevo texto con la palabra vaca
while (t.lower()).count('vaca')==0:	Valida que el texto incluya al menos vez
t=input('Ingresá un texto con vaca: ')	vaca
vaca_txt.write(t+'\n')	Graba en el archivo la línea leída
vaca_txt.close()	Cierra el archivo
vaca_txt=open('arch1.txt')	Reabre el archivo, pero ahora sólo lectura
todas=vaca_txt.readlines()	Carga la lista todas con las líneas del archivo. Cada línea es un elemento
for linea in todas:	Muestra el contenido de la lista quitando
print(linea.strip("\n"))	\n para evitar doble interlineado, uno sale naturalmente por el print()
vaca_txt.close()	Cierra el archivo

Archivos Planos – Comma Separated Values (csv)

- los datos se separan por comas o puntos y comas (este último lo usa excel)
- como al abrir un archivo se usa la función open(), se puede editar una string y automatizar la construcción del nombre del archivo

```
• Integración con bucle de Carga
completo=open('datosCompletos.csv','w')
b=[]

j in range(1,4):
    dat=open('datos'+str(j)+'.csv')
    a=dat.readlines()
    dat.close()
for i in range(len(a)):
    a[i]=a[i].strip('\n')
    a[i]=a[i].split(';')
    a[i][3]=int(a[i][3])

    b=b+a
b.sort(reverse=True,key=lambda b: (b[3],b[2]))
for elemento in b:
    print(elemento)
for elemento in b:
    elemento[3]=str(elemento[3])
    ele=';'.join(elemento)+'\n'
    completo.write(ele)

completo.close()
```

Manejo de Errores

- mientras más tarde se identifica el error, más costos genera el proceso de corrección
- cuando hablamos de sistemas y programas
 - el ingreso de datos en unos de los puntos con frecuencia de errores
 - el ingreso de datos es el lugar más económico donde se pueden detectar y salvar errores
- para disminuir la tasa de errores
 - simplificamos el formato de los datos o los adaptamos a sistemas de lectura automática (código de barra, qr, etc)
 - validar los ingresos / lectura (proceso por el que se verifica y rechaza un ingreso de datos, buscando detectar posibles errores)
- siempre hay posibilidades de que el usuario se equivoque al ingresar los datos
- no todos los errores se podrán detectar en el ingreso. Los que si no todos tendrán una ecuación costo-beneficio que justifique su validación. Pero cuando validamos?. Cuando se pueda y cuando convenga.

se recomienda validar el ingreso siempre que se pueda y cuando la ecuación costo-beneficio sea positiva

- para validar un ingreso debemos escribir una o varias condiciones que nos permitan testear esos datos o identificar la posible presencia de errores
- si detectamos un ingreso erróneo, podemos hacernos las siguientes preguntas:
 - rechazamos el ingreso y paramos la ejecución del programa?
 - continuamos la ejecución ignorando el ingreso?
 - deja de tratarse el caso que contiene un error?

- se pide la colaboración del usuario y se le solicita un reingreso?
- Lo más lógico sería solicitar un reingreso para validarlo y ver si tiene errores, si los tiene se le vuelve a pedir hasta que no haya más errores.

Mostraremos un ejemplo de cómo validar números pares:

```
# Valida el ingreso de un número par
par = int(input('Ingresá un número par: '))

while par < 0 or par % 2 != 0: #también te serviría la condición par<0 or par%2
    par = int(input('Ingresá un número par: '))
print('Tu número es',par)
```

Una ejecución ejemplo podría ser:

```
Ingresá un número par: 11
Ingresá un número par: 8
Tu número es 8
>>>
```

Retenemos al usuario en el proceso de ingreso, la condición para continuar, o sea iterar dentro del bucle, es que se cumpla lo que no debe pasar. O sea no debemos escribir la condición de ingreso válido. Otra opción es crear la condición que queremos cumplir y agregarle el **operador not** adelante para negar al mismo.

- un ejemplo, cuando solicitamos el ingreso de un número entero y lo casteamos en la misma línea, si el usuario se equivoca e ingresa un dato erróneo Python enloquece. Para manejar este error tendríamos que validar que el valor ingresado sea compatible con el número solicitado y recién castear.

```
par=int(input('Ingresá un número par: '))
```

Ahora cuando ingresamos un número negativo lo toma como no válido, este problema también se da con los métodos de string como isdecimal() o isdigit()

```
# Solicita el ingreso de un número entero
numero = input('Ingresá un número entero: ')
while not numero.isnumeric():
    print('El valor ingresado no es válido.')
    numero = input('Ingresá un número entero: ')
print('Tu número entero es ',int(numero))
```

Manejo de Excepciones

- es un evento que ocurre durante la ejecución de programas que interrumpen el flujo normal de la ejecución. O sea es un objeto de Python que representa un error y contiene información del mismo
- Vemos que el error se da en la celda [1] de la línea 1
- tipo de error o nombre de la excepción → `NameError`
- breve descripción del error → el nombre 'Name' no está definido, Python no sabe a qué nos referimos cuando llamamos a Name
- Python viene con excepciones incorporadas y tiene la posibilidad de crear excepciones autodefinidas. Estas son las más usadas:

```
In [1]: name

-----
NameError                                Traceback (most recent call last)
Cell In [1], line 1
----> 1 name

NameError: name 'name' is not defined
```


Excepción	Causa del error
<code>AssertionError</code>	Falla <code>assert</code> .
<code>AttributeError</code>	Falla la referencia de atributos.
<code>EOFError</code>	Se genera cuando la función <code>input()</code> alcanza la condición de fin de archivo (EOF).
<code>FloatingPointError</code>	Falla una operación de punto flotante.
<code>GeneratorExit</code>	Se genera cuando se llama al método <code>close()</code> de un generador.
<code>ImportError</code>	Se genera cuando no se encuentra el módulo importado.

<code>IndexError</code>	Se genera cuando el índice de una secuencia está fuera de rango.
<code>KeyError</code>	Se genera cuando una <code>key</code> no se encuentra en un diccionario.
<code>KeyboardInterrupt</code>	Se genera cuando el usuario presiona la tecla de interrupción (Ctrl+C o Eliminar)
<code>MemoryError</code>	Se genera cuando una operación se queda sin memoria.
<code>NameError</code>	Se genera cuando una variable no se encuentra en el ámbito local o global.
<code>OSError</code>	Se genera cuando la operación del sistema causa un error relacionado con el sistema.
<code>ReferenceError</code>	Se genera cuando se utiliza un proxy de referencia débil para acceder a un atributo del referente recolectado como basura.

try / except

- sentencia estructurada usada para capturar y manejar excepciones
- Python ejecuta el código e intenta hacer una acción con la instrucción `try`, si tiene éxito el programa prosigue de lo contrario si surge un error, la sentencia lo captura y lo trata en la cláusula `except`
- `try` → cláusula que expresa lo que esperamos que falle
- `except` → cláusula donde planeamos manejar las fallas específicas en el código

```
try:
    # Corre este código, algo puede fallar.
except:
    # Se ejecuta este código cuando ocurre una excepción en el bloque de try.

Mostraremos cómo resolver el problema del ingreso de un entero utilizando esta facilidad:

numero = None
while type(numero) is not int:
    try:
        numero = input('Ingresá un número entero: ')
        numero = int(numero)
        print('Tu número entero es ', int(numero))
    except ValueError:
        print('El valor ingresado no es válido.')
```

Una ejecución posible sería:
Ingresá un número entero: 1
El valor ingresado no es válido.
Ingresá un número entero: 0.6
El valor ingresado no es válido.
Ingresá un número entero: -7
Tu número entero es -7
>>>

- A tener en cuenta:
 - el manejo de excepciones dentro de un bucle es para darle al usuario los intentos que necesite. Sólo salimos del bucle cuando se puede ejecutar el bloque vinculado a la cláusula `try`
 - hay situaciones donde hay múltiples excepciones. Si decidimos usarlos debemos consignarlos, de lo contrario capturara todos los errores posibles

Caso 1: No se indica ninguna excepción en particular

```
try:
    # Corre este código, algo puede fallar.
except:
    # Se ejecuta este código cuando ocurre una excepción en el bloque de try.
```

Caso 2: Se indican distintas excepciones en un mismo bloque `except`.

```
try:
    # Corre este código, algo puede fallar.
except Excepción_1, Excepción_2:
    # Se ejecuta este código cuando ocurre cualquiera de las dos excepciones.
```

Caso 3: Se indican distintas excepciones en bloques diferentes de `except`.

```
try:
    # Corre este código, algo puede fallar.
except Excepción_1:
    # Se ejecuta este código cuando ocurre Excepcion_1.
except Excepción_2:
    # Se ejecuta este código cuando ocurre Excepcion_2.
```

⚠ ¡Atención!

También podríamos querer “no hacer nada” frente a una excepción, pero evitando que el programa se bloquee. Para ello, existe pass.

```
try:
    # Corre este código, algo puede fallar.
except:
    pass # No hagas nada.
```

Quizá pass puede parecer poco útil en un primer momento, pero puede volverse tu mejor amigo. Por ejemplo, mientras estás pensando la estructura del código, y querés ir probando algunas líneas cuando aún no está terminado, y evitar que el programa se interrumpa.

```
if numero % 15 == 0:
    pass # Completar.
elif numero % 3 == 0:
    pass # Completar.
elif numero % 5 == 0:
    pass # Completar.
else:
    pass # Completar.
```

try con else

- else es una sentencia usada cuando deseamos ejecutar un bloque de código específico sólo si no ocurre ninguna excepción en el bloque try

```
try:
    # Corre este código, algo puede fallar.
except:
    # Se ejecuta este código cuando ocurre una excepción en el bloque de try.
else:
    # Se ejecuta este código cuando no existe una excepción.
```

- ¿Por qué es una mala idea poner más código en el bloque try?
 - hay posibilidad de encontrarnos con excepciones que no anticipamos
 - reduce la legibilidad. Cuanto más código está en la cláusula try, menos claridad tiene su estructura. Lo ideal sería que fuera lo más chico posible

try con finally

- usado cuando queremos ejecutar una acción por más que haya ocurrido, o no, un error en un programa

```
try:
    # Corre este código, algo puede fallar.
except:
    # Se ejecuta este código cuando ocurre una excepción en el bloque de try.
else:
    # Se ejecuta este código cuando no existe una excepción.
finally:
    # Siempre se ejecuta y bloquea el código.
```

- siempre se ejecutará
- finaliza el programa
- si se produce una excepción no controlada ejecutará el código antes que ocurra
- si tenemos un return en el bloque try, finally interrumpe ese retorno para ejecutar primero su propio código

```
def probar_finally():
    try:
        return 2
    finally:
        print("Código del bloque finally")

probar_finally()
```

Miremos su ejecución:

Código del bloque finally

```
2
>>>
```

cláusula raise

- permite lanzar una excepción si se produce cierta condición
 - útil en situaciones donde necesitamos generar una excepción personalizada (ejemplo recibir datos incorrectos o cualquier error de validación)
- ```
x = 10

if x > 5:
 raise Exception(f'No debe exceder 5. El valor es: {x}').

Miremos su ejecución:

Exception: No debe exceder 5. El valor es: 10
>>>
```
- tenemos que usarlo de forma razonable, no es correcto lanzar una excepción para atajar inmediatamente con un except. Lanzar una excepción sirve para elevar el error a una función previa y manejar el error desde ahí

## Ejemplo de como usar declaraciones

- Crear una función con parámetros x e y que calcule el cociente entre ambos. Deberá usar:
- try para probar el cálculo
  - except en caso de que ocurra una excepción con los números ingresados
  - else para brindar la respuesta en un formato adecuado
- finally para mostrar al usuario que este bloque siempre se ejecuta.

|                                                                                                                                                                                                                                                                                                                 |                                                                                                                                                                                                                                                                                                                                                                                                          |
|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <pre>def calcular_division(x, y):     try:         cociente = x / y     except ZeroDivisionError:         pass     else:         pass     finally:         pass</pre>                                                                                                                                           | realizamos un cálculo de división e identificamos cualquier excepción que surja cuando el usuario proporciona valores no válidos (ejemplo de este caso sería el denominador cero donde obtendremos ZeroDivisionError, esta excepción que debe ser considerada con la cláusula except)                                                                                                                    |
| <pre>def calcular_division(x, y):     try:         cociente = x / y     except ZeroDivisionError:         print("No se puede dividir por cero.")     else:         print("El cociente es: ", cociente)     finally:         print("El bloque finally siempre se ejecuta")</pre>                                 | dentro la cláusula except escribimos el mensaje para informarle al usuario que los datos ingresados son válidos. Devolvemos el cociente en la cláusula else y agregamos el mensaje indicado en la consigna                                                                                                                                                                                               |
| <p><b>Caso 1:</b> x = 15, y = 5</p> <p>El cociente es: 5.0<br/>El bloque finally siempre se ejecuta</p> <p>&gt;&gt;&gt;</p> <p>En este caso, Python intenta hacer la división 15 / 5, y como es posible, ejecuta el bloque else. El bloque finally, se ejecuta siempre.</p> <p><b>Caso 2:</b> x = 15, y = 0</p> | <pre>No se puede dividir por cero. El bloque finally siempre se ejecuta  &gt;&gt;&gt;</pre> <p>En este caso, Python intenta hacer la división 15 / 0 pero aparece el mensaje de error ZeroDivisionError, que es capturado por la cláusula except que le indica cómo tratarlo: en este caso, mostrándole al usuario el mensaje "No se puede dividir por cero". El bloque finally, se ejecuta siempre.</p> |

# Aclaraciones

- independientemente cómo se ejecute el bloque de try-except y sus variaciones, al finalizar, el programa sigue con su flujo de control (siempre y cuando no explote)

```
def calcular_division(x, y):
 print(f"\nDividiendo {x} sobre {y}")
 try:
 cociente = x / y
 except ZeroDivisionError:
 print("No se puede dividir por cero.")
 else:
 print("El cociente es: ", cociente)
 finally:
 print("El bloque finally siempre se ejecuta")

print("Este texto siempre se va a imprimir después del bloque try-except")

calcular_division(15, 0)
calcular_division(15, 5)
```

```
Dividiendo 15 sobre 0
No se puede dividir por cero.
El bloque finally siempre se ejecuta
Este texto siempre se va a imprimir después del bloque try-except

Dividiendo 15 sobre 5
El cociente es: 3.0
El bloque finally siempre se ejecuta
Este texto siempre se va a imprimir después del bloque try-except
> []
```

El manejo de errores con try-except es una forma donde nuestro programa puede ser más robusto y resistente a los errores. Al mismo tiempo, no todas las validaciones requieren de uno (hay validaciones que se pueden hacer con las herramientas de control que conocemos, if, elif, else)

## Pandas

- es una biblioteca de Python usada para el análisis y la manipulación de los datos en bruto
- puede ayudar al usuario a limpiar, transformar y analizar datos rápido y eficientemente
- su estructura de datos principal es el DataFrame (df). Una tabla bidimensional compuesta por filas y columnas, semejantes a una hoja de cálculo de Excel. Usados para almacenar y manipular grandes cantidades de datos. Proporciona funcionalidades (selección y filtrado de datos, agregación de datos y relación de operaciones matemáticas y estadísticas)

## Pasos

|                                                                                                                                                                                                                                                                                                                                                                                                                                            |                                               |
|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------|
| <pre>import pandas as pd</pre>                                                                                                                                                                                                                                                                                                                                                                                                             | importamos pandas con su alias pd             |
| <pre>pd.__version__<br/><br/><u>Resultado:</u><br/>{ "type": "string" }</pre>                                                                                                                                                                                                                                                                                                                                                              | vemos la versión                              |
| <pre>data = { 'animal': ['cat', 'cat', 'snake', 'dog', 'dog', 'cat', 'snake', 'cat',<br/>                'dog', 'dog'],<br/>        'age': [2.5, 3, 0.5, None, 5, 2, 4.5, None, 7, 3],<br/>        'visits': [1, 3, 2, 3, 2, 3, 1, 1, 2, 1],<br/>        'priority': ['yes', 'yes', 'no', 'yes', 'no', 'no', 'no', 'yes', 'no',<br/>                    'no'] }<br/><br/>labels = ['a', 'b', 'c', 'd', 'e', 'f', 'g', 'h', 'i', 'j']</pre> | tenemos el diccionario data y la lista labels |

```
df = pd.DataFrame(data, index=labels)
df # Esto lo ponemos para mostrarlo
```

Resultado:

|   | animal | age | visits | priority |
|---|--------|-----|--------|----------|
| a | cat    | 2.5 | 1      | yes      |
| b | cat    | 3.0 | 3      | yes      |
| c | snake  | 0.5 | 2      | no       |
| d | dog    | NaN | 3      | yes      |
| e | dog    | 5.0 | 2      | no       |
| f | cat    | 2.0 | 3      | no       |
| g | snake  | 4.5 | 1      | no       |
| h | cat    | NaN | 1      | yes      |
| i | dog    | 7.0 | 2      | no       |
| j | dog    | 3.0 | 1      | no       |

creamos un df a partir del diccionario y usando la lista como índice

```
df.info()
```

Resultado:

```
<class 'pandas.core.frame.DataFrame'>
Index: 10 entries, a to j
Data columns (total 4 columns):
Column Non-Null Count Dtype
--- -
0 animal 10 non-null object
1 age 8 non-null float64
2 visits 10 non-null int64
3 priority 10 non-null object
dtypes: float64(1), int64(1), object(2)
memory usage: 400.0+ bytes
```

**df.describe()** # Otra forma

|       | age      | visits    |
|-------|----------|-----------|
| count | 8.000000 | 10.000000 |
| mean  | 3.437500 | 1.900000  |
| std   | 2.007797 | 0.875595  |
| min   | 0.500000 | 1.000000  |
| 25%   | 2.375000 | 1.000000  |
| 50%   | 3.000000 | 2.000000  |
| 75%   | 4.625000 | 2.750000  |
| max   | 7.000000 | 3.000000  |

mostramos información

El método info() devuelve información del df junto con sus índices y columnas, los valores no nulos y el uso de memoria.

Otra forma es usando la función describe() que devuelve una serie con resumen descriptivo que incluye: número de datos, la suma, el mínimo, la media, la desviación típica y los cuartiles

```
df.head(3)
```

Resultado:

|   | animal | age | visits | priority |
|---|--------|-----|--------|----------|
| a | cat    | 2.5 | 1      | yes      |
| b | cat    | 3.0 | 3      | yes      |
| c | snake  | 0.5 | 2      | no       |

# Otra forma

```
df.iloc[:3]
```

Resultado:

|   | animal | age | visits | priority |
|---|--------|-----|--------|----------|
| a | cat    | 2.5 | 1      | yes      |
| b | cat    | 3.0 | 3      | yes      |
| c | snake  | 0.5 | 2      | no       |

El método head() devuelve las primeras filas especificadas del df, si no lo especificamos nos devuelve las primeras cinco.

El método iloc[] en este ejemplo nos devuelve las primeras tres filas

```
df.loc[:, ['animal', 'age']]
```

# Otra manera

Resultado:

|   | animal | age |
|---|--------|-----|
| a | cat    | 2.5 |
| b | cat    | 3.0 |
| c | snake  | 0.5 |
| d | dog    | NaN |
| e | dog    | 5.0 |
| f | cat    | 2.0 |
| g | snake  | 4.5 |
| h | cat    | NaN |

```
df[['animal', 'age']]
```

Resultado:

|   | animal | age |
|---|--------|-----|
| a | cat    | 2.5 |
| b | cat    | 3.0 |
| c | snake  | 0.5 |
| d | dog    | NaN |
| e | dog    | 5.0 |
| f | cat    | 2.0 |
| g | snake  | 4.5 |
| h | cat    | NaN |

Usado para mostrar columnas de un df.

El método loc[] devuelve un df con los elementos de las filas de la lista filas y las columnas de la lista columnas. Si queremos que muestre todas las columnas usamos : (dos puntos).

También podemos usar doble corchete con el nombre de las columnas a especificar, por default nos va a traer todo

```
df.loc[df.index[[3, 4, 8]], ['animal', 'age']]
```

Resultado:

|   | animal | age |
|---|--------|-----|
| d | dog    | NaN |
| e | dog    | 5.0 |
| i | dog    | 7.0 |

seleccionamos las filas [3, 4, 8] y las columnas 'animal' y 'age' usando el método index

```
df[df['visits'] > 3] #No hay ninguno ja
```

Resultado:

```
Empty DataFrame
Columns: [animal, age, visits, priority]
Index: []
```

Ponemos un condición dentro de los corchetes para filtrar los elementos de nuestra tabla.

En el ejemplo seleccionamos las filas que tengan n° de visitas mayores que tres

```
df[df['age'].isnull()]
```

Resultado:

|   | animal | age | visits | priority |
|---|--------|-----|--------|----------|
| d | dog    | NaN | 3      | yes      |
| h | cat    | NaN | 1      | yes      |

La función isnull() devuelve los valores NaN, ésta es usada como condición, donde devuelve True si el valor de la columna es nulo o NaN

```
df[(df['animal'] == 'cat') & (df['age'] < 3)]
```

Resultado:

|   | animal | age | visits | priority |
|---|--------|-----|--------|----------|
| a | cat    | 2.5 | 1      | yes      |
| f | cat    | 2.0 | 3      | no       |

Ejemplo de una condición más compleja, donde decimos que queremos las filas donde el animal sea cat y la edad menor a tres

```
df[df['age'].between(2, 4)]
```

Resultado:

|   | animal | age | visits | priority |
|---|--------|-----|--------|----------|
| a | cat    | 2.5 | 1      | yes      |
| b | cat    | 3.0 | 3      | yes      |
| f | cat    | 2.0 | 3      | no       |
| j | dog    | 3.0 | 1      | no       |

La función `between()` elige los elementos ubicados en el rango `[a, b]`, sería equivalente a columna `>= a` & columna `<= b`. Como ejemplo le pedimos seleccionar las filas con edad entre dos y cuatro inclusive

```
df.loc['f', 'age'] = 1.5
```

Para cambiar el valor de una fila en específica, usamos el método `loc[ ]`, especificamos las columnas y el `=` como si fuera una variable. En este ejemplo cambiamos la edad de la fila f a 1.5

```
df['visits'].sum()
```

Resultado:

19

La función `sum()` es usada para sumar todos los valores de la columna seleccionada. En este ejemplo estamos sumando la cantidad de visitas de todas las filas del df, como resultado vemos que son 19

```
df.groupby('animal')['age'].mean()
```

Resultado:

| animal |          |
|--------|----------|
| cat    | 2.333333 |
| dog    | 5.000000 |

```
snake 2.500000
Name: age, dtype: float64
```

En el ejemplo usamos dos métodos. El primero, `groupby()`, agrupa las filas según el animal y el segundo, `mean()`, calcula el promedio de la edad

```
df.loc['k'] = ['dog', 5.5, 2, 'no']
df
```

Resultado:

|   | animal | age | visits | priority |
|---|--------|-----|--------|----------|
| a | cat    | 2.5 | 1      | yes      |
| b | cat    | 3.0 | 3      | yes      |
| c | snake  | 0.5 | 2      | no       |
| d | dog    | NaN | 3      | yes      |
| e | dog    | 5.0 | 2      | no       |
| f | cat    | 1.5 | 3      | no       |
| g | snake  | 4.5 | 1      | no       |
| h | cat    | NaN | 1      | yes      |
| i | dog    | 7.0 | 2      | no       |
| j | dog    | 3.0 | 1      | no       |
| k | dog    | 5.5 | 2      | no       |

Usamos `loc` para agregar una nueva fila al df junto a los valores que tiene

```
df = df.drop('k')
df
```

Resultado:

|   | animal | age | visits | priority |
|---|--------|-----|--------|----------|
| a | cat    | 2.5 | 1      | yes      |
| b | cat    | 3.0 | 3      | yes      |
| c | snake  | 0.5 | 2      | no       |
| d | dog    | NaN | 3      | yes      |
| e | dog    | 5.0 | 2      | no       |
| f | cat    | 1.5 | 3      | no       |
| g | snake  | 4.5 | 1      | no       |
| h | cat    | NaN | 1      | yes      |
| i | dog    | 7.0 | 2      | no       |
| j | dog    | 3.0 | 1      | no       |

Usamos la función `drop()` para borrar la fila agregada recientemente



|                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |        |                            |        |          |          |   |                            |     |                                                                                                                                                                                          |      |   |     |     |   |      |   |        |     |   |       |   |     |     |   |      |   |     |     |   |       |   |     |     |   |       |   |        |     |   |       |   |       |     |   |      |   |     |     |   |       |   |     |     |   |       |                                                                                                                                                                                                                                                                         |
|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|----------------------------|--------|----------|----------|---|----------------------------|-----|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------|---|-----|-----|---|------|---|--------|-----|---|-------|---|-----|-----|---|------|---|-----|-----|---|-------|---|-----|-----|---|-------|---|--------|-----|---|-------|---|-------|-----|---|------|---|-----|-----|---|-------|---|-----|-----|---|-------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <pre>df['animal'].value_counts()</pre> <p><u>Resultado:</u></p> <table><tr><td>cat</td><td>4</td><td>snake</td><td>2</td></tr><tr><td>dog</td><td>4</td><td>Name: animal, dtype: int64</td><td></td></tr></table>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              | cat    | 4                          | snake  | 2        | dog      | 4 | Name: animal, dtype: int64 |     | <p>La función value_counts() devuelve la cantidad de veces que aparece un valor para una columna específica.</p> <p>En este ejemplo queremos contar la cantidad de animales en el df</p> |      |   |     |     |   |      |   |        |     |   |       |   |     |     |   |      |   |     |     |   |       |   |     |     |   |       |   |        |     |   |       |   |       |     |   |      |   |     |     |   |       |   |     |     |   |       |                                                                                                                                                                                                                                                                         |
| cat                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            | 4      | snake                      | 2      |          |          |   |                            |     |                                                                                                                                                                                          |      |   |     |     |   |      |   |        |     |   |       |   |     |     |   |      |   |     |     |   |       |   |     |     |   |       |   |        |     |   |       |   |       |     |   |      |   |     |     |   |       |   |     |     |   |       |                                                                                                                                                                                                                                                                         |
| dog                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            | 4      | Name: animal, dtype: int64 |        |          |          |   |                            |     |                                                                                                                                                                                          |      |   |     |     |   |      |   |        |     |   |       |   |     |     |   |      |   |     |     |   |       |   |     |     |   |       |   |        |     |   |       |   |       |     |   |      |   |     |     |   |       |   |     |     |   |       |                                                                                                                                                                                                                                                                         |
| <pre>df.sort_values(by=['age', 'visits'], ascending=[False, True])</pre> <p><u>Resultado:</u></p> <table><tr><td></td><td>animal</td><td>age</td><td>visits</td><td>priority</td></tr><tr><td>i</td><td>dog</td><td>7.0</td><td>2</td><td>no</td></tr><tr><td>e</td><td>dog</td><td>5.0</td><td>2</td><td>no</td></tr><tr><td>g</td><td>snake</td><td>4.5</td><td>1</td><td>no</td></tr><tr><td>j</td><td>dog</td><td>3.0</td><td>1</td><td>no</td></tr><tr><td>b</td><td>cat</td><td>3.0</td><td>3</td><td>yes</td></tr><tr><td>a</td><td>cat</td><td>2.5</td><td>1</td><td>yes</td></tr><tr><td>f</td><td>cat</td><td>1.5</td><td>3</td><td>no</td></tr><tr><td>c</td><td>snake</td><td>0.5</td><td>2</td><td>no</td></tr><tr><td>h</td><td>cat</td><td>NaN</td><td>1</td><td>yes</td></tr><tr><td>d</td><td>dog</td><td>NaN</td><td>3</td><td>yes</td></tr></table>                         |        | animal                     | age    | visits   | priority | i | dog                        | 7.0 | 2                                                                                                                                                                                        | no   | e | dog | 5.0 | 2 | no   | g | snake  | 4.5 | 1 | no    | j | dog | 3.0 | 1 | no   | b | cat | 3.0 | 3 | yes   | a | cat | 2.5 | 1 | yes   | f | cat    | 1.5 | 3 | no    | c | snake | 0.5 | 2 | no   | h | cat | NaN | 1 | yes   | d | dog | NaN | 3 | yes   | <p>La función sort_values() ordena la tabla usando ciertos criterios como parámetros (by, ascending). En by ponemos una lista de las columnas y en ascending la lista de booleanos donde indicamos si queremos que el orden sea ascendente o no</p>                     |
|                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                | animal | age                        | visits | priority |          |   |                            |     |                                                                                                                                                                                          |      |   |     |     |   |      |   |        |     |   |       |   |     |     |   |      |   |     |     |   |       |   |     |     |   |       |   |        |     |   |       |   |       |     |   |      |   |     |     |   |       |   |     |     |   |       |                                                                                                                                                                                                                                                                         |
| i                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              | dog    | 7.0                        | 2      | no       |          |   |                            |     |                                                                                                                                                                                          |      |   |     |     |   |      |   |        |     |   |       |   |     |     |   |      |   |     |     |   |       |   |     |     |   |       |   |        |     |   |       |   |       |     |   |      |   |     |     |   |       |   |     |     |   |       |                                                                                                                                                                                                                                                                         |
| e                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              | dog    | 5.0                        | 2      | no       |          |   |                            |     |                                                                                                                                                                                          |      |   |     |     |   |      |   |        |     |   |       |   |     |     |   |      |   |     |     |   |       |   |     |     |   |       |   |        |     |   |       |   |       |     |   |      |   |     |     |   |       |   |     |     |   |       |                                                                                                                                                                                                                                                                         |
| g                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              | snake  | 4.5                        | 1      | no       |          |   |                            |     |                                                                                                                                                                                          |      |   |     |     |   |      |   |        |     |   |       |   |     |     |   |      |   |     |     |   |       |   |     |     |   |       |   |        |     |   |       |   |       |     |   |      |   |     |     |   |       |   |     |     |   |       |                                                                                                                                                                                                                                                                         |
| j                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              | dog    | 3.0                        | 1      | no       |          |   |                            |     |                                                                                                                                                                                          |      |   |     |     |   |      |   |        |     |   |       |   |     |     |   |      |   |     |     |   |       |   |     |     |   |       |   |        |     |   |       |   |       |     |   |      |   |     |     |   |       |   |     |     |   |       |                                                                                                                                                                                                                                                                         |
| b                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              | cat    | 3.0                        | 3      | yes      |          |   |                            |     |                                                                                                                                                                                          |      |   |     |     |   |      |   |        |     |   |       |   |     |     |   |      |   |     |     |   |       |   |     |     |   |       |   |        |     |   |       |   |       |     |   |      |   |     |     |   |       |   |     |     |   |       |                                                                                                                                                                                                                                                                         |
| a                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              | cat    | 2.5                        | 1      | yes      |          |   |                            |     |                                                                                                                                                                                          |      |   |     |     |   |      |   |        |     |   |       |   |     |     |   |      |   |     |     |   |       |   |     |     |   |       |   |        |     |   |       |   |       |     |   |      |   |     |     |   |       |   |     |     |   |       |                                                                                                                                                                                                                                                                         |
| f                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              | cat    | 1.5                        | 3      | no       |          |   |                            |     |                                                                                                                                                                                          |      |   |     |     |   |      |   |        |     |   |       |   |     |     |   |      |   |     |     |   |       |   |     |     |   |       |   |        |     |   |       |   |       |     |   |      |   |     |     |   |       |   |     |     |   |       |                                                                                                                                                                                                                                                                         |
| c                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              | snake  | 0.5                        | 2      | no       |          |   |                            |     |                                                                                                                                                                                          |      |   |     |     |   |      |   |        |     |   |       |   |     |     |   |      |   |     |     |   |       |   |     |     |   |       |   |        |     |   |       |   |       |     |   |      |   |     |     |   |       |   |     |     |   |       |                                                                                                                                                                                                                                                                         |
| h                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              | cat    | NaN                        | 1      | yes      |          |   |                            |     |                                                                                                                                                                                          |      |   |     |     |   |      |   |        |     |   |       |   |     |     |   |      |   |     |     |   |       |   |     |     |   |       |   |        |     |   |       |   |       |     |   |      |   |     |     |   |       |   |     |     |   |       |                                                                                                                                                                                                                                                                         |
| d                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              | dog    | NaN                        | 3      | yes      |          |   |                            |     |                                                                                                                                                                                          |      |   |     |     |   |      |   |        |     |   |       |   |     |     |   |      |   |     |     |   |       |   |     |     |   |       |   |        |     |   |       |   |       |     |   |      |   |     |     |   |       |   |     |     |   |       |                                                                                                                                                                                                                                                                         |
| <pre>df['priority'] = df['priority'].map({'yes': True, 'no': False})</pre> <p><u>Resultado:</u></p> <table><tr><td></td><td>animal</td><td>age</td><td>visits</td><td>priority</td></tr><tr><td>a</td><td>cat</td><td>2.5</td><td>1</td><td>True</td></tr><tr><td>b</td><td>cat</td><td>3.0</td><td>3</td><td>True</td></tr><tr><td>c</td><td>snake</td><td>0.5</td><td>2</td><td>False</td></tr><tr><td>d</td><td>dog</td><td>NaN</td><td>3</td><td>True</td></tr><tr><td>e</td><td>dog</td><td>5.0</td><td>2</td><td>False</td></tr><tr><td>f</td><td>cat</td><td>1.5</td><td>3</td><td>False</td></tr><tr><td>g</td><td>snake</td><td>4.5</td><td>1</td><td>False</td></tr><tr><td>h</td><td>cat</td><td>NaN</td><td>1</td><td>True</td></tr><tr><td>i</td><td>dog</td><td>7.0</td><td>2</td><td>False</td></tr><tr><td>j</td><td>dog</td><td>3.0</td><td>1</td><td>False</td></tr></table> |        | animal                     | age    | visits   | priority | a | cat                        | 2.5 | 1                                                                                                                                                                                        | True | b | cat | 3.0 | 3 | True | c | snake  | 0.5 | 2 | False | d | dog | NaN | 3 | True | e | dog | 5.0 | 2 | False | f | cat | 1.5 | 3 | False | g | snake  | 4.5 | 1 | False | h | cat   | NaN | 1 | True | i | dog | 7.0 | 2 | False | j | dog | 3.0 | 1 | False | <p>La función map() transforma los valores de la columna entera. Le pasamos un diccionario con elementos del tipo valor viejo - valor nuevo.</p> <p>En este ejemplo tenemos una columna llamada priority y le queremos cambiar los valores sí y no por true y false</p> |
|                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                | animal | age                        | visits | priority |          |   |                            |     |                                                                                                                                                                                          |      |   |     |     |   |      |   |        |     |   |       |   |     |     |   |      |   |     |     |   |       |   |     |     |   |       |   |        |     |   |       |   |       |     |   |      |   |     |     |   |       |   |     |     |   |       |                                                                                                                                                                                                                                                                         |
| a                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              | cat    | 2.5                        | 1      | True     |          |   |                            |     |                                                                                                                                                                                          |      |   |     |     |   |      |   |        |     |   |       |   |     |     |   |      |   |     |     |   |       |   |     |     |   |       |   |        |     |   |       |   |       |     |   |      |   |     |     |   |       |   |     |     |   |       |                                                                                                                                                                                                                                                                         |
| b                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              | cat    | 3.0                        | 3      | True     |          |   |                            |     |                                                                                                                                                                                          |      |   |     |     |   |      |   |        |     |   |       |   |     |     |   |      |   |     |     |   |       |   |     |     |   |       |   |        |     |   |       |   |       |     |   |      |   |     |     |   |       |   |     |     |   |       |                                                                                                                                                                                                                                                                         |
| c                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              | snake  | 0.5                        | 2      | False    |          |   |                            |     |                                                                                                                                                                                          |      |   |     |     |   |      |   |        |     |   |       |   |     |     |   |      |   |     |     |   |       |   |     |     |   |       |   |        |     |   |       |   |       |     |   |      |   |     |     |   |       |   |     |     |   |       |                                                                                                                                                                                                                                                                         |
| d                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              | dog    | NaN                        | 3      | True     |          |   |                            |     |                                                                                                                                                                                          |      |   |     |     |   |      |   |        |     |   |       |   |     |     |   |      |   |     |     |   |       |   |     |     |   |       |   |        |     |   |       |   |       |     |   |      |   |     |     |   |       |   |     |     |   |       |                                                                                                                                                                                                                                                                         |
| e                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              | dog    | 5.0                        | 2      | False    |          |   |                            |     |                                                                                                                                                                                          |      |   |     |     |   |      |   |        |     |   |       |   |     |     |   |      |   |     |     |   |       |   |     |     |   |       |   |        |     |   |       |   |       |     |   |      |   |     |     |   |       |   |     |     |   |       |                                                                                                                                                                                                                                                                         |
| f                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              | cat    | 1.5                        | 3      | False    |          |   |                            |     |                                                                                                                                                                                          |      |   |     |     |   |      |   |        |     |   |       |   |     |     |   |      |   |     |     |   |       |   |     |     |   |       |   |        |     |   |       |   |       |     |   |      |   |     |     |   |       |   |     |     |   |       |                                                                                                                                                                                                                                                                         |
| g                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              | snake  | 4.5                        | 1      | False    |          |   |                            |     |                                                                                                                                                                                          |      |   |     |     |   |      |   |        |     |   |       |   |     |     |   |      |   |     |     |   |       |   |     |     |   |       |   |        |     |   |       |   |       |     |   |      |   |     |     |   |       |   |     |     |   |       |                                                                                                                                                                                                                                                                         |
| h                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              | cat    | NaN                        | 1      | True     |          |   |                            |     |                                                                                                                                                                                          |      |   |     |     |   |      |   |        |     |   |       |   |     |     |   |      |   |     |     |   |       |   |     |     |   |       |   |        |     |   |       |   |       |     |   |      |   |     |     |   |       |   |     |     |   |       |                                                                                                                                                                                                                                                                         |
| i                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              | dog    | 7.0                        | 2      | False    |          |   |                            |     |                                                                                                                                                                                          |      |   |     |     |   |      |   |        |     |   |       |   |     |     |   |      |   |     |     |   |       |   |     |     |   |       |   |        |     |   |       |   |       |     |   |      |   |     |     |   |       |   |     |     |   |       |                                                                                                                                                                                                                                                                         |
| j                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              | dog    | 3.0                        | 1      | False    |          |   |                            |     |                                                                                                                                                                                          |      |   |     |     |   |      |   |        |     |   |       |   |     |     |   |      |   |     |     |   |       |   |     |     |   |       |   |        |     |   |       |   |       |     |   |      |   |     |     |   |       |   |     |     |   |       |                                                                                                                                                                                                                                                                         |
| <pre>df['animal'] = df['animal'].replace('snake', 'python')</pre> <p><u>Resultado:</u></p> <table><tr><td></td><td>animal</td><td>age</td><td>visits</td><td>priority</td></tr><tr><td>a</td><td>cat</td><td>2.5</td><td>1</td><td>True</td></tr><tr><td>b</td><td>cat</td><td>3.0</td><td>3</td><td>True</td></tr><tr><td>c</td><td>python</td><td>0.5</td><td>2</td><td>False</td></tr><tr><td>d</td><td>dog</td><td>NaN</td><td>3</td><td>True</td></tr><tr><td>e</td><td>dog</td><td>5.0</td><td>2</td><td>False</td></tr><tr><td>f</td><td>cat</td><td>1.5</td><td>3</td><td>False</td></tr><tr><td>g</td><td>python</td><td>4.5</td><td>1</td><td>False</td></tr><tr><td>h</td><td>cat</td><td>NaN</td><td>1</td><td>True</td></tr><tr><td>i</td><td>dog</td><td>7.0</td><td>2</td><td>False</td></tr><tr><td>j</td><td>dog</td><td>3.0</td><td>1</td><td>False</td></tr></table>        |        | animal                     | age    | visits   | priority | a | cat                        | 2.5 | 1                                                                                                                                                                                        | True | b | cat | 3.0 | 3 | True | c | python | 0.5 | 2 | False | d | dog | NaN | 3 | True | e | dog | 5.0 | 2 | False | f | cat | 1.5 | 3 | False | g | python | 4.5 | 1 | False | h | cat   | NaN | 1 | True | i | dog | 7.0 | 2 | False | j | dog | 3.0 | 1 | False | <p>Usamos replace() para reemplazar el valor por uno nuevo (también se puede usar map())</p>                                                                                                                                                                            |
|                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                | animal | age                        | visits | priority |          |   |                            |     |                                                                                                                                                                                          |      |   |     |     |   |      |   |        |     |   |       |   |     |     |   |      |   |     |     |   |       |   |     |     |   |       |   |        |     |   |       |   |       |     |   |      |   |     |     |   |       |   |     |     |   |       |                                                                                                                                                                                                                                                                         |
| a                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              | cat    | 2.5                        | 1      | True     |          |   |                            |     |                                                                                                                                                                                          |      |   |     |     |   |      |   |        |     |   |       |   |     |     |   |      |   |     |     |   |       |   |     |     |   |       |   |        |     |   |       |   |       |     |   |      |   |     |     |   |       |   |     |     |   |       |                                                                                                                                                                                                                                                                         |
| b                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              | cat    | 3.0                        | 3      | True     |          |   |                            |     |                                                                                                                                                                                          |      |   |     |     |   |      |   |        |     |   |       |   |     |     |   |      |   |     |     |   |       |   |     |     |   |       |   |        |     |   |       |   |       |     |   |      |   |     |     |   |       |   |     |     |   |       |                                                                                                                                                                                                                                                                         |
| c                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              | python | 0.5                        | 2      | False    |          |   |                            |     |                                                                                                                                                                                          |      |   |     |     |   |      |   |        |     |   |       |   |     |     |   |      |   |     |     |   |       |   |     |     |   |       |   |        |     |   |       |   |       |     |   |      |   |     |     |   |       |   |     |     |   |       |                                                                                                                                                                                                                                                                         |
| d                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              | dog    | NaN                        | 3      | True     |          |   |                            |     |                                                                                                                                                                                          |      |   |     |     |   |      |   |        |     |   |       |   |     |     |   |      |   |     |     |   |       |   |     |     |   |       |   |        |     |   |       |   |       |     |   |      |   |     |     |   |       |   |     |     |   |       |                                                                                                                                                                                                                                                                         |
| e                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              | dog    | 5.0                        | 2      | False    |          |   |                            |     |                                                                                                                                                                                          |      |   |     |     |   |      |   |        |     |   |       |   |     |     |   |      |   |     |     |   |       |   |     |     |   |       |   |        |     |   |       |   |       |     |   |      |   |     |     |   |       |   |     |     |   |       |                                                                                                                                                                                                                                                                         |
| f                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              | cat    | 1.5                        | 3      | False    |          |   |                            |     |                                                                                                                                                                                          |      |   |     |     |   |      |   |        |     |   |       |   |     |     |   |      |   |     |     |   |       |   |     |     |   |       |   |        |     |   |       |   |       |     |   |      |   |     |     |   |       |   |     |     |   |       |                                                                                                                                                                                                                                                                         |
| g                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              | python | 4.5                        | 1      | False    |          |   |                            |     |                                                                                                                                                                                          |      |   |     |     |   |      |   |        |     |   |       |   |     |     |   |      |   |     |     |   |       |   |     |     |   |       |   |        |     |   |       |   |       |     |   |      |   |     |     |   |       |   |     |     |   |       |                                                                                                                                                                                                                                                                         |
| h                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              | cat    | NaN                        | 1      | True     |          |   |                            |     |                                                                                                                                                                                          |      |   |     |     |   |      |   |        |     |   |       |   |     |     |   |      |   |     |     |   |       |   |     |     |   |       |   |        |     |   |       |   |       |     |   |      |   |     |     |   |       |   |     |     |   |       |                                                                                                                                                                                                                                                                         |
| i                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              | dog    | 7.0                        | 2      | False    |          |   |                            |     |                                                                                                                                                                                          |      |   |     |     |   |      |   |        |     |   |       |   |     |     |   |      |   |     |     |   |       |   |     |     |   |       |   |        |     |   |       |   |       |     |   |      |   |     |     |   |       |   |     |     |   |       |                                                                                                                                                                                                                                                                         |
| j                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              | dog    | 3.0                        | 1      | False    |          |   |                            |     |                                                                                                                                                                                          |      |   |     |     |   |      |   |        |     |   |       |   |     |     |   |      |   |     |     |   |       |   |     |     |   |       |   |        |     |   |       |   |       |     |   |      |   |     |     |   |       |   |     |     |   |       |                                                                                                                                                                                                                                                                         |
|                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |        |                            |        |          |          |   |                            |     |                                                                                                                                                                                          |      |   |     |     |   |      |   |        |     |   |       |   |     |     |   |      |   |     |     |   |       |   |     |     |   |       |   |        |     |   |       |   |       |     |   |      |   |     |     |   |       |   |     |     |   |       |                                                                                                                                                                                                                                                                         |



# Funciones de Pandas

| Función    | Definición                                                                                                  | Ejemplo de Uso                                    |
|------------|-------------------------------------------------------------------------------------------------------------|---------------------------------------------------|
| read_csv() | Lee un archivo CSV y lo carga en un DataFrame                                                               | <code>df = pd.read_csv('data.csv')</code>         |
| head()     | Muestra las primeras filas del DataFrame. Se le puede pasar un número para pedirle una N cantidad de filas. | <code>df.head()</code><br><code>df.head(7)</code> |
| tail()     | Muestra las últimas filas del                                                                               | <code>df.tail()</code>                            |

|          |                                                               |                           |
|----------|---------------------------------------------------------------|---------------------------|
| dropna() | Elimina filas con valores faltantes del DataFrame             | <code>df.dropna()</code>  |
| fillna() | Rellena los valores faltantes del DataFrame con un valor dado | <code>df.fillna(0)</code> |

|           |                                                  |                                              |
|-----------|--------------------------------------------------|----------------------------------------------|
| groupby() | Agrupar el DataFrame según una o varias columnas | <code>grouped = df.groupby('columna')</code> |
| sum()     | Calcula la suma de los valores en el DataFrame   | <code>total = df['columna'].sum()</code>     |
| mean()    | Calcula la media de los valores en el DataFrame  | <code>average = df['columna'].mean()</code>  |
| max()     | Encuentra el valor máximo en el DataFrame        | <code>max_value = df['columna'].max()</code> |

|                |                                                                |                                                          |
|----------------|----------------------------------------------------------------|----------------------------------------------------------|
| count()        | Devuelve el número de elementos no nulos en el objeto          | <code>non_null_count = df['columna'].count()</code>      |
| map()          | Aplica una función o un diccionario a los elementos del objeto | <code>mapped_values = df['columna'].map(func)</code>     |
| value_counts() | Devuelve una serie que contiene recuentos de valores únicos    | <code>value_counts = df['columna'].value_counts()</code> |

|            |                                                              |                                   |
|------------|--------------------------------------------------------------|-----------------------------------|
| shape      | Devuelve la forma (número de filas y columnas) del DataFrame | <code>shape = df.shape</code>     |
| info()     | Muestra información sobre el DataFrame                       | <code>df.info()</code>            |
| describe() | Genera estadísticas descriptivas del DataFrame               | <code>df.describe()</code>        |
| columns    | Devuelve los nombres de las columnas del DataFrame           | <code>columns = df.columns</code> |
| dtypes     | Devuelve los tipos de datos de las columnas del DataFrame    | <code>dtypes = df.dtypes</code>   |

|           |                                                                   |                                                     |
|-----------|-------------------------------------------------------------------|-----------------------------------------------------|
| min()     | Encuentra el valor mínimo en el DataFrame                         | <code>min_value = df['columna'].min()</code>        |
| unique()  | Devuelve los valores únicos en una columna del DataFrame          | <code>unique_values = df['columna'].unique()</code> |
| nunique() | Devuelve el número de valores únicos en una columna del DataFrame | <code>num_unique = df['columna'].nunique()</code>   |

|               |                                                            |                                                                                                                             |
|---------------|------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------|
| sort_values() | Ordena el DataFrame por una o varias columnas              | <code>df.sort_values('columna')</code><br><code>df.sort_values(by=['columna1', 'columna2'], ascending=[True, False])</code> |
| merge()       | Combina dos DataFrames en función de una o varias columnas | <code>merged_df = pd.merge(df1, df2, on='columna')</code>                                                                   |
| to_csv()      | Guarda el DataFrame en un archivo CSV                      | <code>df.to_csv('output.csv', index=False)</code>                                                                           |
| size()        | Devuelve el número total de elementos en el objeto         | <code>total_elements = df.size</code>                                                                                       |

|               |                                                           |                                         |
|---------------|-----------------------------------------------------------|-----------------------------------------|
| isnull()      | Devuelve una máscara booleana que indica valores nulos    | is_null_mask = df['columna'].isnull()   |
| notnull()     | Devuelve una máscara booleana que indica valores no nulos | not_null_mask = df['columna'].notnull() |
| reset_index() | Restablece los índices del DataFrame                      | df_reset = df.reset_index()             |

|        |                                                           |                                         |
|--------|-----------------------------------------------------------|-----------------------------------------|
| loc[]  | Permite la indexación y selección de datos por etiquetas. | df.loc[2]<br><br>df.loc[1:3, 'columna'] |
| iloc[] | Permite la indexación y selección de datos por posición.  | df.iloc[2]<br><br>df.iloc[1:3, 0]       |
| get()  | Devuelve el valor correspondiente a una clave dada        | value = df.get('columna')               |

## loc vs iloc

- son métodos de indexación usados para acceder a filas y columnas específicas
- loc → se basa en etiquetas, usa las etiquetas de filas y columnas para acceder a los datos
- iloc → se basa en números, usa las posiciones de los valores para acceder a los datos

|   | padron | nombre   | apellido | edad | carrera     | mail_fiuba     | ingreso_en_pandemia |
|---|--------|----------|----------|------|-------------|----------------|---------------------|
| 1 | 109234 | Agustín  | Ramírez  | 18   | Química     | AgustinRamirez | si                  |
| 2 | 110456 | María    | Portillo | 19   | Civil       | MariaPortillo  | no                  |
| 3 | 102890 | Gustavo  | Feliche  | 20   | Mecánica    | None           | si                  |
| 4 | 100450 | Federico | Monte    | 35   | Informática | FedericoMonte  | si                  |
| 5 | 100430 | Felipe   | Michael  | 25   | Informática | FelipeMichael  | no                  |
| 6 | 100455 | Manuela  | Gomez    | 20   | Informática | ManuelaGomez   | no                  |
| 7 | 105689 | Laura    | Ramírez  | 24   | Sistemas    | None           | si                  |

Para el dataframe de la foto, usando **loc** para obtener nombre y edad se ve así:

```
nombres_edades_v2 = df.loc[:, ['nombre', 'edad']]
nombres_edades_v2
```

Pero usando **iloc** se ve así:

```
nombres_edades_v2 = df.iloc[:, [1,3]]
nombres_edades_v2
```

Los dos puntos (:) marcan que queremos mostrar todas las filas, pero se puede modificar para mostrar menos.  
Por ejemplo:

```
:10 imprime del índice 0 al 10.
```

## Matplotlib

- es una biblioteca de Python usada para crear gráficos/plots
- es una forma rápida de graficar datos en diferentes formatos que pueden ser compartidos o publicados
- pyplot le proporciona un interfaz, está diseñada siguiendo el estilo matlab (tiene comandos y argumentos similares a éste)

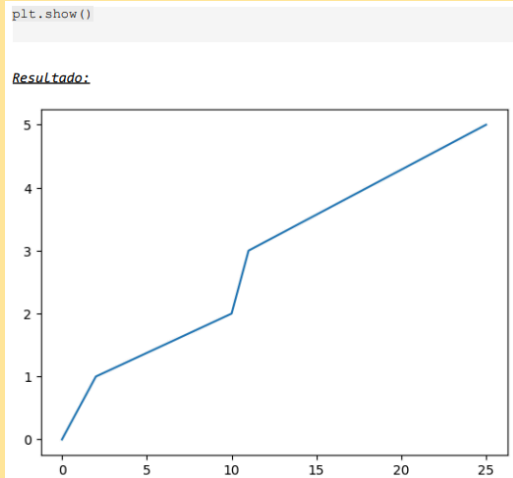
import matplotlib.pyplot as plt

importamos la librería junto al alias plt

# Función Plot y Show

- recibe dos vectores que tienen el mismo tamaño, estos representan una serie de puntos en el plano cartesiano. El primer arreglo son coordenadas X y el segundo son coordenadas Y
- tiene como función unir todos los puntos, con líneas, para que podamos ver el gráfico de la función
- `plt.plot()` → método usado para crear un gráfico para verificar resultados
- `plt.subplots()` → para gráficos más complejos con más finos
- `show` → función con la que creamos la imagen de todos los gráficos definidos

```
Grafico elemental
x = [0,2,10,11,18,25]
y = [0,1,2,3,4,5]
fig = plt.figure()
plt.plot(x, y)
```



o también

```
fig, ax = plt.subplots()
ax.plot(x, y)
plt.show()
```

- cuál se diferencia? → los outputs

1. `plt.figure()` crea una **figura pero sin axes** (por eso dice `<Figure size 640x480 with 0 Axes>`, se tiene una figura y 0 axes).
2. `plt.subplots()` permite **crear ambos** (por eso dice `<Figure size 640x480 with 1 Axes>`, `<Axes: >`, hay figura y axes).

## FIGURA

- es el marco que delimita la zona donde se traza los gráficos
- una figura puede tener muchos axes

## AXES

- son los gráficos, o sea las áreas donde los puntos se pueden especificar en términos de coordenadas
- un axes determinado solo puede estar contenido en una única figura

**NO CONFUNDIR AXES CON AXIS.** Los axes son ejes cartesianos encargados de establecer los límites, la escala y las dimensiones del gráfico

un axes puede tener dos axes (si es un gráfico plano) o tres (si es un gráfico tridimensional)

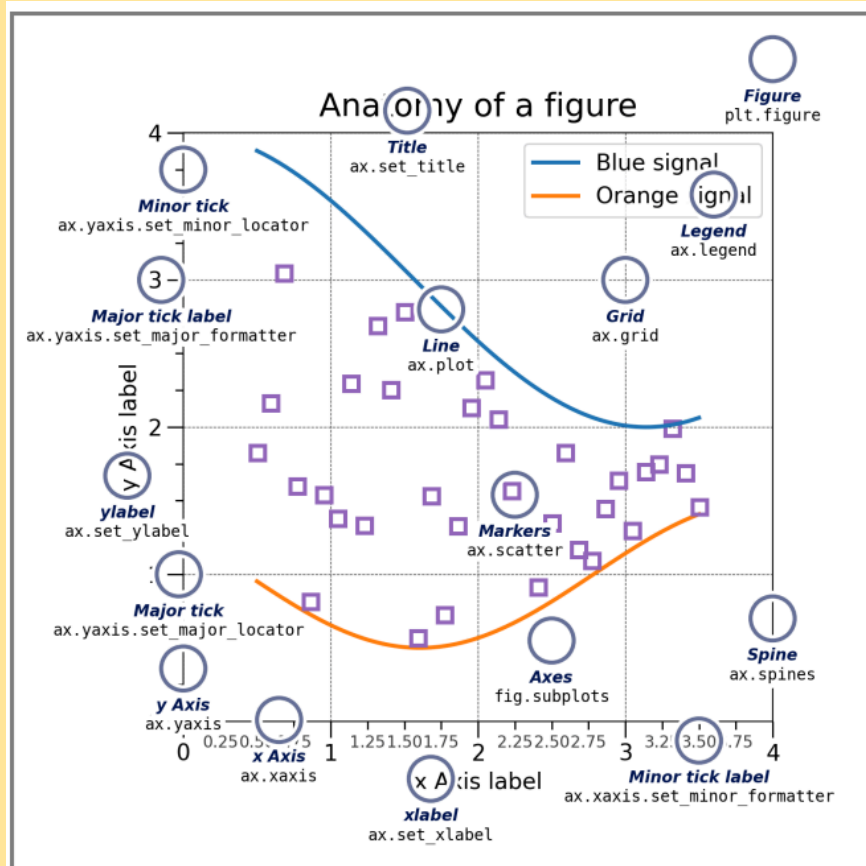
```
Grafico elemental explicado:
fig = plt.figure() # Se crea una figura vacía sin Axes

plt.plot(x, y) # Agregar plotteo (gráfico de línea)
plt.show() # Mostrar
```

```
Grafico elemental explicado:
fig, ax = plt.subplots() # Se crea una figura con un único Axes.

ax.plot(x, y) # Agregar los plotteos individuales (gráfico de línea)
plt.show() # Mostrar
```

# Partes de una Figura



## ¿Cómo cambiar el aspecto de los gráficos?

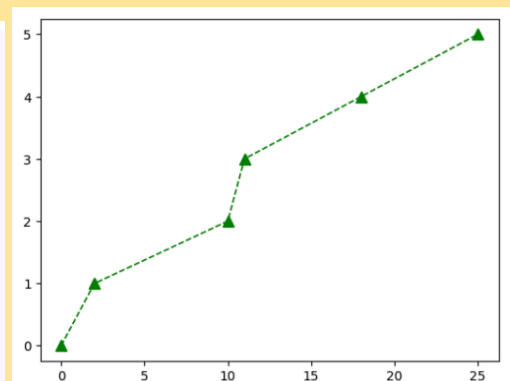
- definir los parámetros dentro de `ax.plot()`
- podemos establecer el tipo de línea y puntos, el grosor, color, etc
- color → para el color
- marker → forma de los puntos marcadores (^, o, v)
- linestyle → estilo de línea (solid/-, dashed/--, dotted/:)
- markersize, linewidth → establece el tamaño del marcador y el espesor de la línea (se usa números, si no le asignamos uno se establecen los predefinidos)

```
x = [0,2,10,11,18,25] # Tiempo (min)
y = [0,1,2,3,4,5] # Distancia (m)

fig, ax = plt.subplots()

ax.plot(x, y, color='green', marker='^', linestyle='--', markersize=8,
 linewidth=1.2)

plt.show()
```



## Grilla o cuadrícula

- usado para leer cada punto
- se agrega usando `ax.grid()`
- podemos modificarla usando parámetros similares a los anteriores pero en la función `ax.grid()`

```
x = [0,2,10,11,18,25] # Tiempo (min)
y = [0,1,2,3,4,5] # Distancia (m)

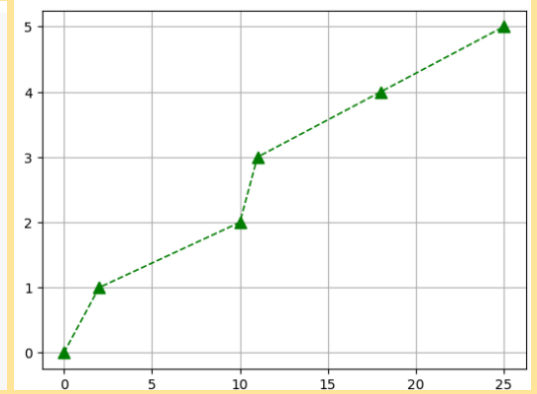
fig, ax = plt.subplots()

ax.plot(x, y, color='green', marker='^', linestyle='--', markersize=8,
linewidth=1.2)

Grilla preestablecida

ax.grid()

plt.show()
```



```
#Gráfica con la grilla preestablecida

x = [0,2,10,11,18,25] # Tiempo (min)
y = [0,1,2,3,4,5] # Distancia (m)

fig, ax = plt.subplots()

ax.plot(x, y, color='green', marker='^', linestyle='--', markersize=8,
linewidth=1.2)

#Grilla modificada

ax.grid(axis = 'y', color = 'gray', linestyle = 'dashed')

plt.show()
```

## Títulos

- es importante agregarle esto a los gráficos para que se pueda entender y explicar qué significa cada eje
- usamos las funciones `ax.set_xlabel()`, `ax.set_ylabel()`, `ax.set_title()`. Cada una recibe un string que se usará como etiqueta para los ejes x/y o el título

```
x = [0,2,10,11,18,25] # Tiempo (min)
y = [0,1,2,3,4,5] # Distancia (m)

fig, ax = plt.subplots()

ax.plot(x, y, color='green', marker='^', linestyle='--', markersize=8,
linewidth=1.2)

Mostrar el título del gráfico

ax.set_title("Gráfico de posición")

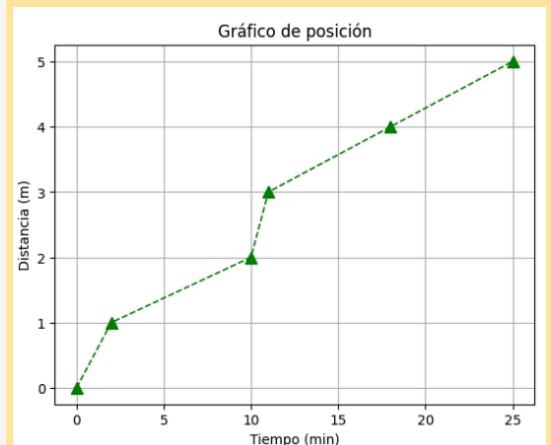
Mostrar el título de los ejes

ax.set_xlabel('Tiempo (min)')
ax.set_ylabel('Distancia (m)')

Grilla preestablecida

ax.grid()

plt.show()
```



# Referencias

- usado cuando el gráfico tiene más de una línea
- lo ponemos dentro de `ax.plot()` donde definimos la referencia como label y luego colocamos `ax.legend()`

```
x = [0,2,10,11,18,25] # Tiempo (min)
y = [0,1,2,3,4,5] # Distancia (m)

fig, ax = plt.subplots()

ax.plot(x, y, label='Objeto 1', color='green', marker='^', linestyle='--',
 markersize=8, linewidth=1.2)

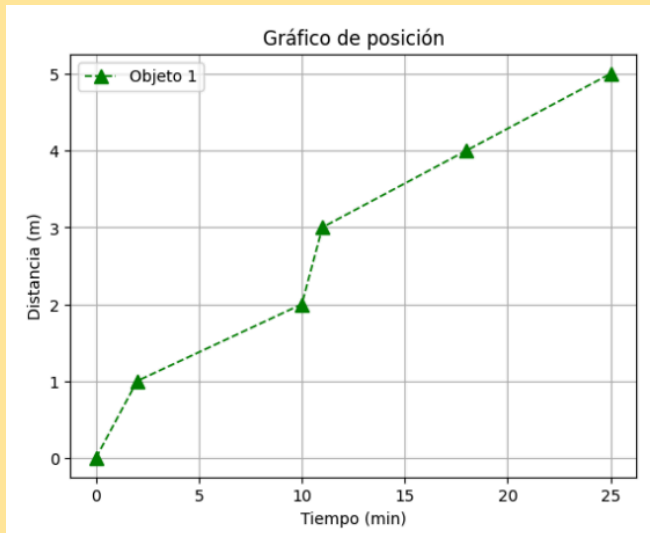
Mostrar el título del gráfico
ax.set_title("Gráfico de posición")

Mostrar el título de los ejes
ax.set_xlabel('Tiempo (min)')
ax.set_ylabel('Distancia (m)')

Agregar la referencia
ax.legend()

Grilla preestablecida
ax.grid()

plt.show()
```



## Características de los Ejes

- podemos establecer los límites de los ejes usando `ax.set_xlim()` y `ax.set_ylim()`

```
x = [0,2,10,11,18,25] # Tiempo (min)
y = [0,1,2,3,4,5] # Distancia (m)

fig, ax = plt.subplots()

ax.plot(x, y, label='Objeto 1', color='green', marker='^', linestyle='--',
 markersize=8, linewidth=1.2)

Mostrar el título del gráfico
ax.set_title("Gráfico de posición")

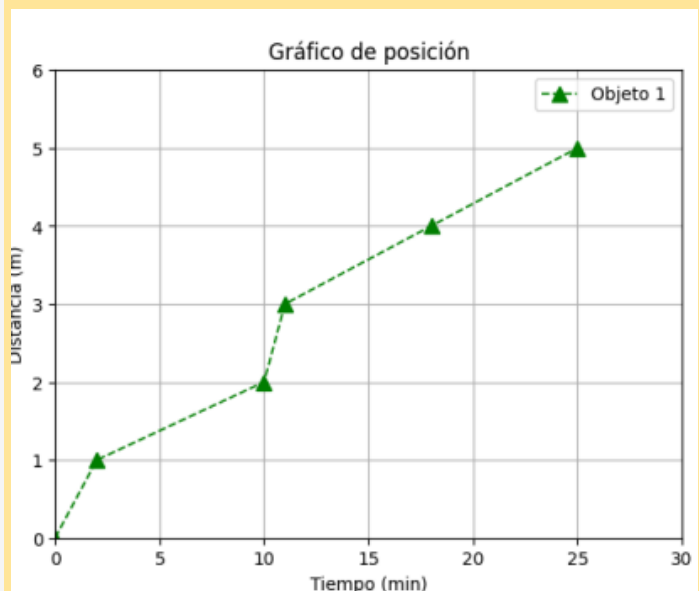
Mostrar el título de los ejes
ax.set_xlabel('Tiempo (min)')
ax.set_ylabel('Distancia (m)')

Establecer los límites de los ejes
ax.set_xlim(0, 30)
ax.set_ylim(0, 6)

Agregar la referencia
ax.legend()

Grilla preestablecida
ax.grid()

plt.show()
```



# Tipos de Gráficos

## Gráfico de Línea

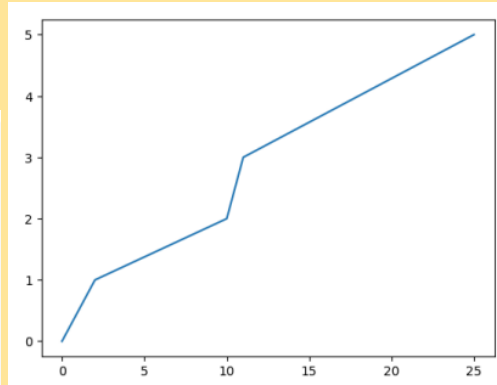
- permite visualizar cambios en los valores a lo largo de un rango continuo (tendencias) (ejemplo: tiempo o distancia)
- lo creamos usando la función `ax.plot()`

```
x = [0,2,10,11,18,25]
y = [0,1,2,3,4,5]

fig, ax = plt.subplots()

ax.plot(x, y)

plt.show()
```



## Gráfico de Dispersión / De Puntos

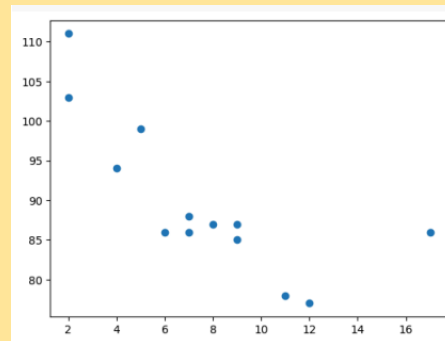
- permite visualizar la relación entre las variables
- la creamos usando la función `ax.scatter()`

```
x = [5,7,8,7,2,17,2,9,4,11,12,9,6]
y = [99,86,87,88,111,86,103,87,94,78,77,85,86]

fig, ax = plt.subplots()

ax.scatter(x, y)

plt.show()
```



## Gráfico de Barras

- permite visualizar proporciones, comparando dos o más valores entre sí
- usamos la función `ax.bar()` para crearlos
- primero recibe un arreglo con las etiquetas de las barras y otro con la altura de cada una de las barras

```
peso = [340, 115, 200, 200, 270]
ingredientes = ['chocolate', 'manteca', 'azúcar', 'huevo', 'harina']

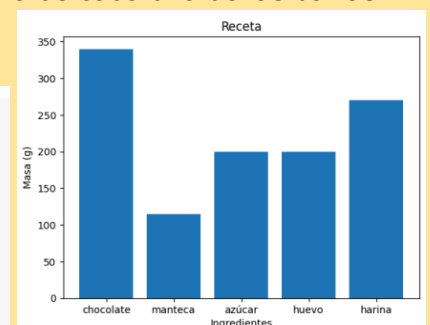
fig, ax = plt.subplots()

ax.bar(ingredientes, peso)
```

```
ax.set_xlabel('Ingredientes')
ax.set_ylabel('Masa (g)')

ax.set_title("Receta")

plt.show()
```



- usamos la función `ax.barh()` si queremos las barras horizontales (no olvidarnos de cambiar los títulos de los ejes)

```
peso = [340, 115, 200, 200, 270]
ingredientes = ['chocolate', 'manteca', 'azúcar', 'huevo', 'harina']
```



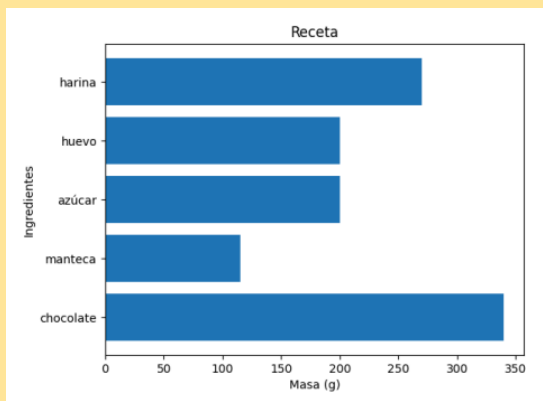
```
fig, ax = plt.subplots()

ax.barh(ingredientes, peso)

ax.set_ylabel('Ingredientes')
ax.set_xlabel('Masa (g)')

ax.set_title("Receta")

plt.show()
```



## Gráfico de Tortas

- al igual que el de barras, este permite visualizar y comparar proporciones pero de manera circular y como parte de un todo
- se crea con la función `ax.pie()`
- usando el parámetro `labels` se le asigna una etiqueta a cada porción y el parámetro `autopct` para que indique como se mostrará el porcentaje

el parámetro `autopct` que indica cómo se mostrará el porcentaje: `%1.1f%` le indica que el porcentaje tendrá un decimal, mientras que `%1.2f%` tendrá 2 decimales.

```
peso = [340, 115, 200, 200, 270]

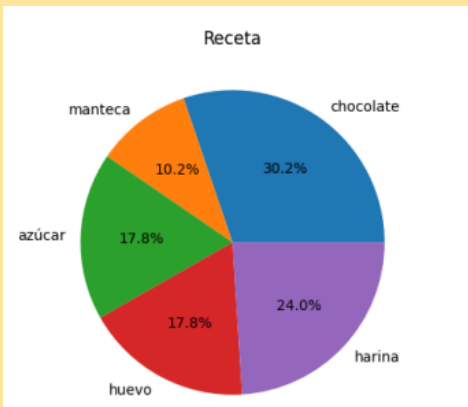
ingredientes = ['chocolate', 'manteca', 'azúcar', 'huevo', 'harina']

fig, ax = plt.subplots()

ax.pie(peso, labels= ingredientes, autopct='%1.1f%%')

ax.set_title("Receta")

plt.show()
```



## Gráficos Múltiples

- sirve para graficar varias curvas en un mismo gráfico

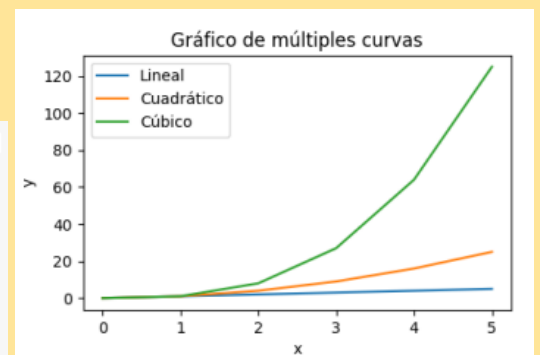
```
Valores que se desean graficar
x = [0, 1, 2, 3, 4, 5]
y_linear = [0, 1, 2, 3, 4, 5]
y_quadratic = [0, 1, 4, 9, 16, 25]
y_cubic = [0, 1, 8, 27, 64, 125]

fig, ax = plt.subplots(figsize=(5, 3))

ax.plot(x, y_linear, label='Lineal')
ax.plot(x, y_quadratic, label='Cuadrático')
ax.plot(x, y_cubic, label='Cúbico')
```

```
ax.set_title("Gráfico de múltiples curvas")
ax.set_xlabel('x')
ax.set_ylabel('y')
ax.legend()

plt.show()
```

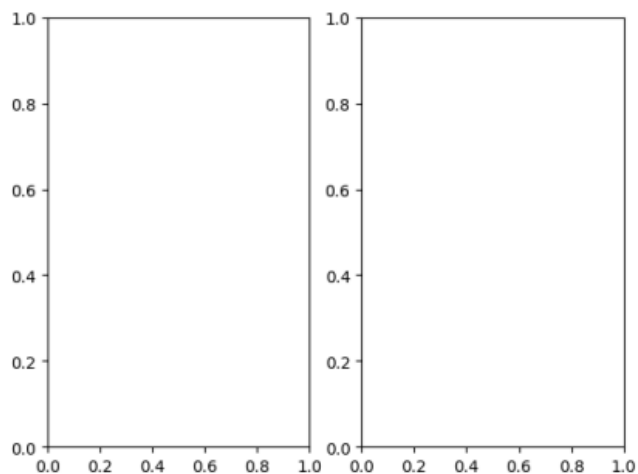


Note que estamos agregando nuevos datos al mismo axes, por lo que siempre usamos `ax.plot()` pero con distintos valores de `y`. Además, se estableció un tamaño de la figura con `figsize=(width, height)`.

## Grilla de Gráficos

- sirve para ver varios axes en una misma figura
- necesitamos definir cuantas columnas (`ncols`) y filas (`nrows`) de gráficos necesitamos

```
fig, ax = plt.subplots(nrows=1, ncols=2) # o simplemente plt.subplots(1,2)
```



De manera análoga, podemos representar las 3 curvas de la figura titulada "Gráfico de múltiples curvas" pero viendo 3 filas en una única columna de gráficos:

```
Valores que se desean graficar
x = [0, 1, 2, 3, 4, 5]
x_linear = [0, 1, 2, 3, 4, 5]
x_quadratic = [0, 1, 4, 9, 16, 25]
x_cubic = [0, 1, 8, 27, 64, 125]

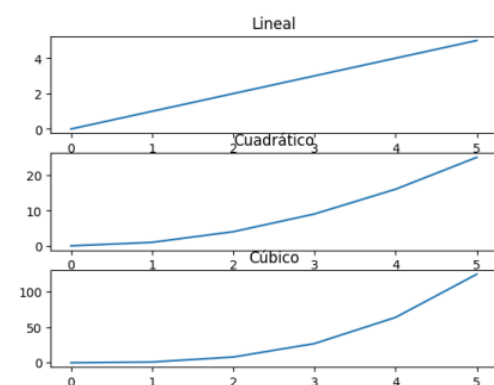
fig, ax = plt.subplots(nrows=3, ncols=1)
```

```
ax[0].plot(x, x_linear)
ax[0].set_title('Lineal')

ax[1].plot(x, x_quadratic)
ax[1].set_title('Cuadrático')

ax[2].plot(x, x_cubic)
ax[2].set_title('Cúbico')

plt.show()
```



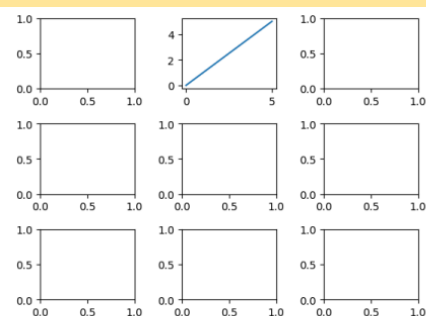
Entonces, ahora en vez de graficar usando `ax.plot()` unicamente, tengo que **indicar la posición del axes con los números dentro del corchete**, ¿pero si tengo varias columnas y filas?

```
fig, ax = plt.subplots(nrows=3, ncols=3)

fig.subplots_adjust(wspace=0.5, hspace=0.5) # Con esto indicamos el espacio libre entre los subplots

ax[0, 1].plot(x, x_linear)

plt.show()
```



Note que dentro del corchete, primero se indica la fila y luego la columna: `ax[fila, columna]`.

## Funciones de Gráficas

```
x = [0, 1, 2, 3, 4, 5]
```

```

x_linear = [0, 1, 2, 3, 4, 5]
x_quadratic = [0, 1, 4, 9, 16, 25]
x_cubic = [0, 1, 8, 27, 64, 125]

fig, ax=plt.subplots(3)

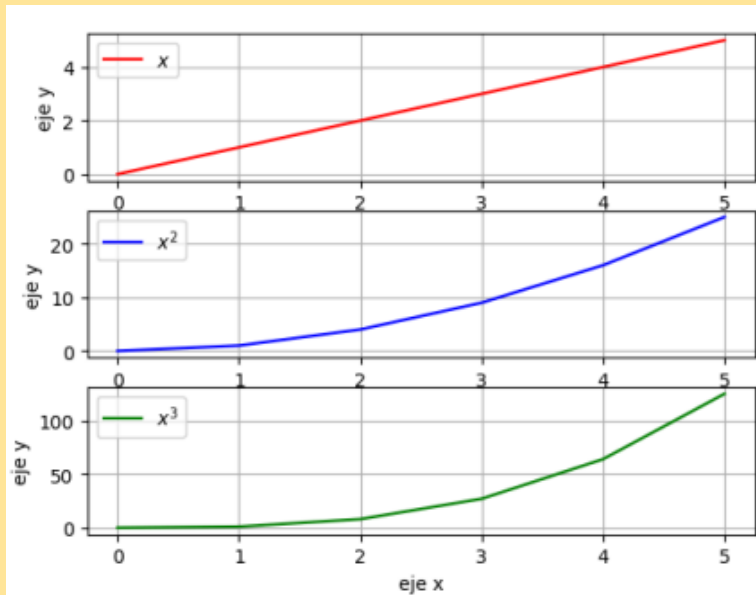
ax[0].plot(x,x_linear,label="x",color="r")
ax[0].set_xlabel("eje x")
ax[0].set_ylabel("eje y")
ax[0].legend()
ax[0].grid()

ax[1].plot(x,x_quadratic,label="x^2",color="b")
ax[1].set_xlabel("eje x")
ax[1].set_ylabel("eje y")
ax[1].legend()
ax[1].grid()

ax[2].plot(x,x_cubic,label="x^3",color="g")
ax[2].set_xlabel("eje x")
ax[2].set_ylabel("eje y")
ax[2].legend()
ax[2].grid()

plt.show()

```



Para evitar lo anterior, definimos una función a la que le debemos entregar los valores a graficar:

```

def create_easy_graph(x, y, label, ax, xlabel, ylabel, title, color):
 """Crea un gráfico a partir de vectores con valores de los ejes x e y.

 Recibe además:
 - El texto para el label
 - El subplot a donde graficar
 - Un label para el eje x
 - Un label para el eje y
 - Un título para el gráfico
 - Un color

 El color y el eje pueden ser None. En ese caso toman valores por default"""

 if color == None:

```

```

 color = "blue"

 # Si sólo haremos un gráfico, no necesito indicarle la posición
 if ax == None:
 fig, ax = plt.subplots()

 # Definimos el gráfico
 ax.plot(x, y, label=label, color=color)
 ax.set_xlabel(xlabel)
 ax.set_ylabel(ylabel)
 ax.set_title(title)

 return ax

```

Y por otro lado:

```

fig , ax = plt.subplots(3)

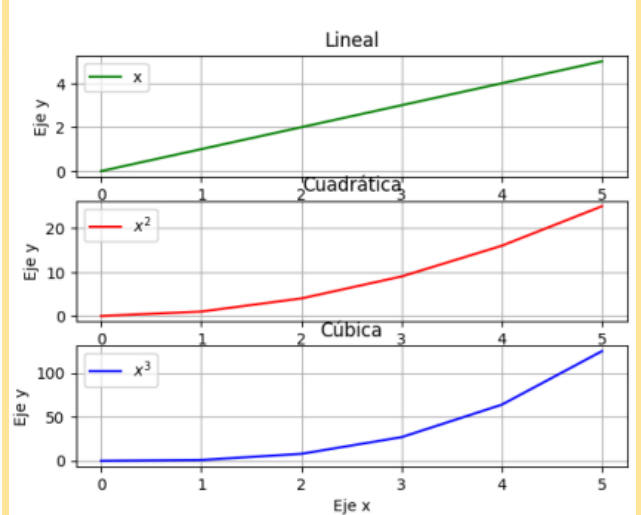
En vez de copiar y pegar el código, llamo a la función easy_graph():
create_easy_graph(x, x_linear, "x", ax[0], "Eje x", "Eje y", "Lineal",
color="green")

create_easy_graph(x, x_quadratic, "x^2", ax[1], "Eje x", "Eje y",
"Cuadrática", color="red")

create_easy_graph(x, x_cubic, "x^3", ax[2], "Eje x", "Eje y", "Cúbica", None)

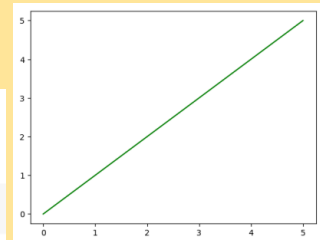
Hacemos un for para agregar la cuadrícula y las referencias en cada axes:
for axes in fig.axes[:]:
 axes.grid()
 axes.legend()

```



Como comentamos dentro de la función, también podemos usar `easy_graph()` para un único gráfico:

```
create_easy_graph(x, x_linear, "x", None, "", "", "", "green")
```



## Gráficos usando DataFrames (df)

- usamos los datos contenidos en el df (no es necesario que provenga de vectores de valores)

```
import pandas as pd

data = {'animal': ['cat', 'snake', 'dog'],
 'age': [2.5, 3, 7],
 'visits': [1, 3, 2],
 'priority': ['yes', 'yes', 'no']}

df = pd.DataFrame(data)

df
```

|   | animal | age | visits | priority |
|---|--------|-----|--------|----------|
| 0 | cat    | 2.5 | 1      | yes      |
| 1 | snake  | 3.0 | 3      | yes      |
| 2 | dog    | 7.0 | 2      | no       |

```
Determino las columnas del DataFrame que queremos graficar
x_values = df['animal']
y_values = df['age']

fig, ax = plt.subplots()

ax.bar(x_values, y_values)

ax.set_xlabel('Animal')
ax.set_ylabel('Edad (años)')

ax.set_title("Mascotas")

plt.show()
```

