

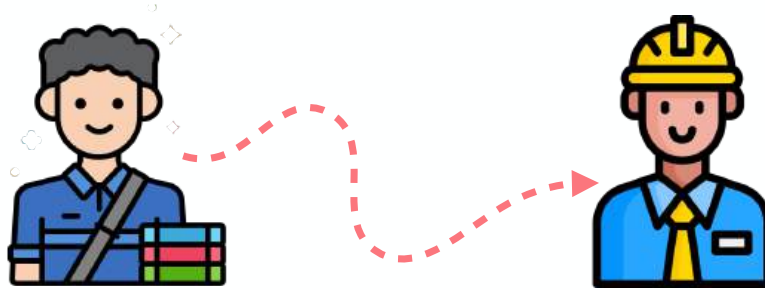
INVESTIGACIÓN OPERATIVA SUPERIOR

(investigación operativa 3)

Presencial

introducción

*nosotros nos **proponemos** ...*



"integrar métodos matemáticos, tecnología para poder implementar en empresas e industrias"

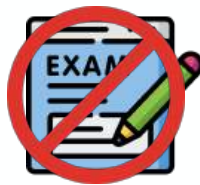
características del curso

4

*otorga
4 créditos*



*50% virtual
50% presencial*



*no hay
parcial*



*trabajo práctico
grupal*



*100%
promocionable**

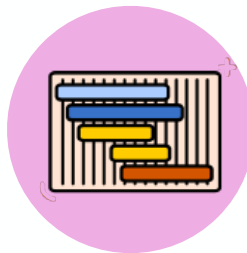
* entregando un paper individual.



gestión



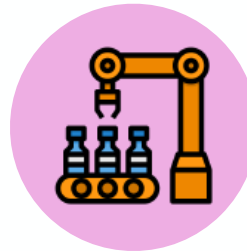
planificación



logística



manufatura

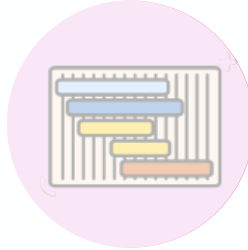


primer módulo: gestión

*introducción a
python*



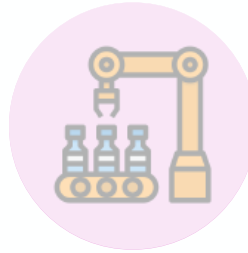
*introducción a apis,
bases de datos y
json*



*integración de
modelos a la empresa*



*introducción a la
ciencia de datos*



*modelos de
inteligencia artificial*

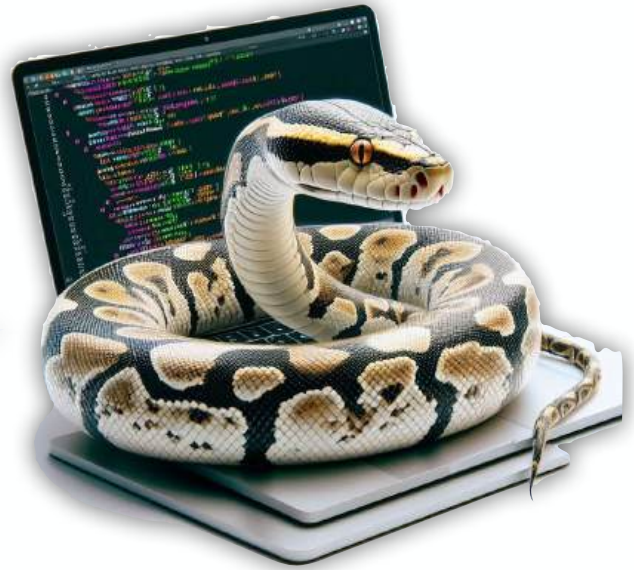
*optimización con
python*



¿preguntas?

una breve ...

introducción a python



*hoy es python ...
¿y mañana?*

¿herramientas?

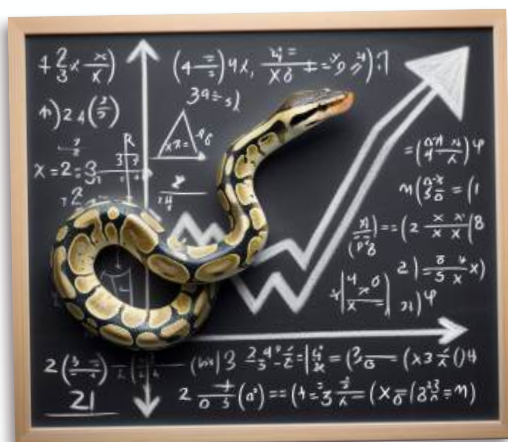


Visual Studio Code

¿alternativas?



python e investigación operativa



optimización con modelos LP/NLP

programación de restricciones

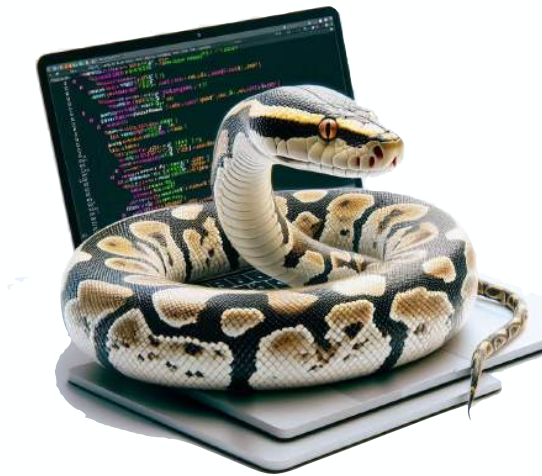
simulación

optimización con metaheurísticas

modelos de inteligencia artificial

modelos de decisión

etc ...



primeras instrucciones en python

comentarios

```
# comentario de una sola linea  
# definición de variables
```

variables

```
x = 1 # variable entera  
y = 1.5 # variable de punto flotante  
s = "x es mayor que y." # variable de texto
```

condicionales

```
# condicionales  
if x>y:  
    print(s)  
else:  
    print("X no es mayor que Y")
```

bucles

```
# bucles  
for i in range(10):  
    x = x * y
```

impresión

```
# impresión de resultados  
print("x =", x, "e y =", y)  
print(f"x = {x} e y = {y}")
```


colecciones en python



listas

(es un tipo de variable muy utilizada en python)

AGUSTIN	EMILIO	CAMILO	VALENTINA	LEONARDO	NICOLAS	LUCAS
[0]	[1]	[2]	[3]	[4]	[5]	[6]

¿quién está en la posición 4?

listas

	"MANZANA"	263.10	232	"MOÑO AZUL SA"		30-71597299-5
[0]	[1]	[2]	[3]	[4]	[5]	[6]

- indexadas
- ordenadas
- mutables
- duplicados
- heterogénea

listas ejemplos ...

```
a = ["a", "b", 1, 1.5, "a", "c", "d", 0] # una lista se inicializa siempre

a.append("hola") # agrego un elemento al final

a.clear()        # borro toda la lista
a.remove(1.5)    # borro el primer elemento igual a 1.5
a.pop(2)         # saco el **tercer** elemento de la lista

c = a.count("a") # cuento la cantidad de elementos iguales a "a"
d = len(a)       # obtengo la cantidad de elementos de la lista

# puedo utilizarlos en bucles para recorrer sus elementos

for i in a:
    print(i)
```

diccionarios

(almacena datos tipo clave:valor)

Clave		Valor
1	→	'hola'
'a'	→	256
'b'	→	1.5
'c'	→	'chau'

diccionarios

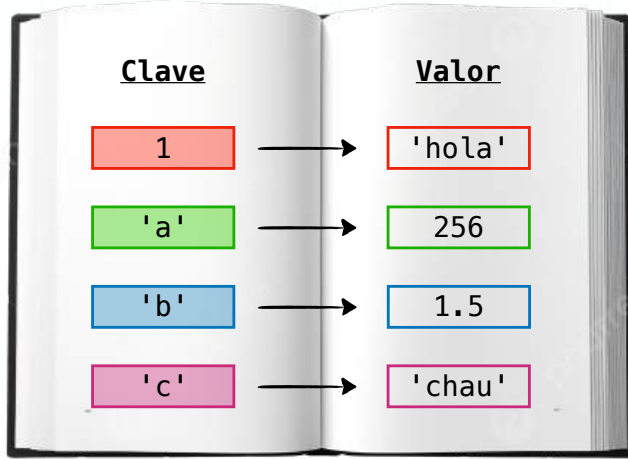
(almacena datos tipo clave:valor)

Clave		Valor
1	→	'hola'
'a'	→	256
'b'	→	
'c'	→	'chau'

Clave		Valor
1	→	'hola'
'a'	→	256
'b'	→	1.5

es posible anidar diccionarios

diccionarios *(almacena datos tipo clave:valor)*



A diagram illustrating a dictionary as an open book. The left page is labeled 'Clave' (Key) and the right page is labeled 'Valor' (Value). Four entries are shown, each with a colored box for the key and a corresponding colored box for the value, connected by an arrow. The entries are: 1 (red) points to 'hola' (red), 'a' (green) points to 256 (green), 'b' (blue) points to 1.5 (blue), and 'c' (pink) points to 'chau' (pink).

Clave	Valor
1	'hola'
'a'	256
'b'	1.5
'c'	'chau'

- *ordenadas* (a partir de versión 3.7)
- *mutables*
- *heterogéneas*
- *la clave es única*

diccionarios *ejemplos ...*

```
# inicialización de un diccionario
dict = {'parametro1': 30, 'parametro2': 45}

# obtener valores a partir de las claves
print(dict['parametro1'])

# actualizar/agregar datos
dict.update({'parametro2': 35})
dict.update({'parametro3': 95})

# eliminar elementos a partir de una clave
del dict['parametro3']
```

sets

hola@gmail.com
verde@fi.uba.ar
azul@fi.uba.ar
lunes@fi.uba.ar
martes@fi.uba.ar
simplex@fi.uba.ar
simulacion@fi.uba.ar
python@fi.uba.ar

- no ordenados
- mutables (eliminar/agregar)
- heterogéneas
- no duplicados
- interesantes operaciones de conjuntos

sets ejemplos ...

inicialización de sets, notar que se utilizan {}

```
alumnos = { 'hola@gmail.com', 'verde@fi.uba.ar', 'azul@fi.uba.ar',  
            'lunes@fi.uba.ar', 'martes@fi.uba.ar', 'simplex@fi.uba.ar',  
            'simulacion@fi.uba.ar' }
```

```
docentes = { 'lsmith@fi.uba.ar',  
             'ilnacucchio@gmail.com',  
             'jdabusti@fi.uba.ar' }
```

se agrega un alumno faltante

```
alumnos.add('python@fi.uba.ar')  
print(alumnos)
```

sets ejemplos ...

lista de emails completa de emails utilizando UNION

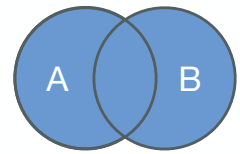
```
emails = alumnos.union(docentes)
print(emails)
```

lista de emails y se elimina un email de la lista de alumnos

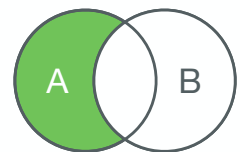
```
emails.clear()
alumnos.discard("verde@fi.uba.ar")
print(emails)
print(alumnos)
```

se genera una lista de docentes utilizando DIFFERENCE

```
d = emails.difference(alumnos)
print(d)
```



UNION



DIFFERENCE

DEMO 4

tuplas



384
08-OCT-2025
OC
CENCOSUD SA

1	CABLE TALLER 2x1.5	100	MT	850	ARS
2	TOMA SUPERFICIE 10A	4	UN	338	ARS
3	PROLONGADOR 6 TOMAS	10	UN	1580	ARS

- ordenados
- inmutables
- heterogéneas
- duplicados

tuplas ejemplos ...

inicialización de la tupla (y sub-tuplas)

```
oc = (384, 'OC', 'CENCOSUD SA',  
      ((1, 'CABLE TALLER 2x1.5', 100, 'MT', 850, 'ARS'),  
       (2, 'TOMA SUPERFICIE 10A', 4, 'UN', 338, 'ARS'),  
       (3, 'PROLONGADOR 6 TOMAS', 10, 'UN', 1580, 'ARS')))
```

oc[3][0].count('ARS') # cantidad de elementos dentro de una tupla

oc.index('CENCOSUD SA') # obtener el índice de un elemento

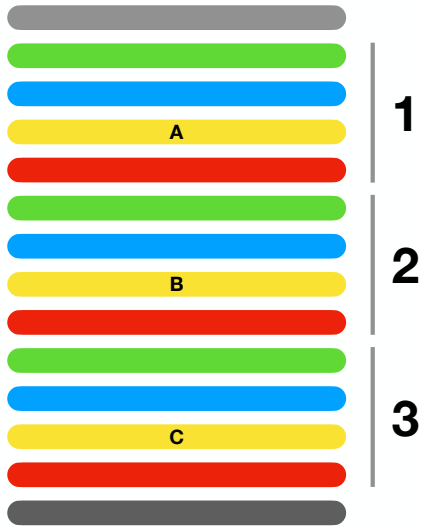
oc[3][0][1] # obtener un elemento específico de la tupla

DEMO 5

diferencias

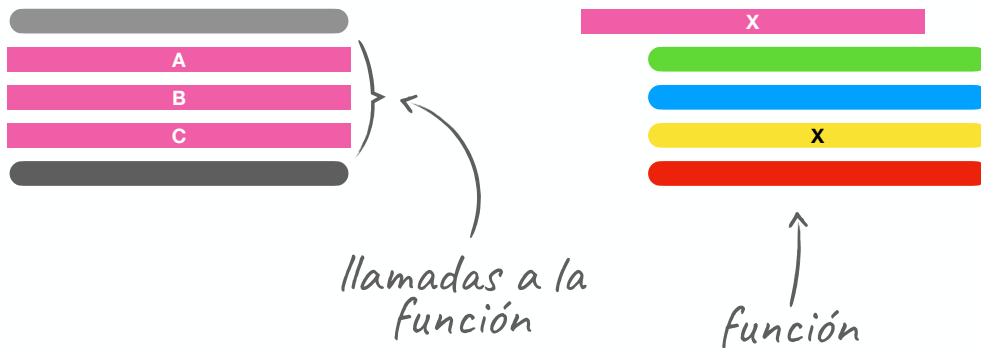
	<i>mutable</i>	<i>ordenado</i>	<i>indexable</i>	<i>duplicados</i>
set	✓ (agregar y borrar)	✗	✗	✗
lista	✓	✓	✓	✓
tupla	✗	✓	✓	✓
diccionario	✓	✓ (a partir de versión 3.7)	✓	✗

funciones



- ordenar código (prolijidad)
- evitar código duplicado
- mejorar el mantenimiento

funciones



funciones ejemplos ...

función simple

```
def saludar():  
    print("hola")
```

```
saludar()
```

función con dos parámetros obligatorios de entrada y resultado

```
def suma(x, y):  
    r = x + y  
    return r
```

```
print(suma(3,4))
```

DEMO 6



bibliotecas
(en python)

instalar/importar bibliotecas

instalar una biblioteca

```
!pip install biblioteca
```

importar módulos de una biblioteca

```
from biblioteca import *
```

```
from biblioteca import módulos
```

```
import biblioteca.módulo as alias
```

importar una biblioteca con todos sus módulos

```
import biblioteca
```

```
import biblioteca as alias
```



numpy

*una biblioteca que mejora el
trabajo con números*

*si vamos a trabajar con
números y/o datos casi seguro
que la usamos ...*

*construido en C sobre un código muy optimizado
gran cantidad de funciones para el manejo de arrays
puede trabajar con arrays muy grandes
las ndarrays utilizan menos memoria que las listas
puede realizar operaciones de manera mucho más rápida y eficaz*

numpy

```
# importamos las bibliotecas

import time
import numpy as np

# se crean las listas y ndarrays

s = 2000000
list1 = list(range(s))
list2 = list(range(s))
array1 = np.arange(s)
array2 = np.arange(s)
```

numpy

mide el tiempo que tarda en sumar dos elementos de las listas

```
start_list = time.time()
result = [(x + y) for x, y in zip(list1, list2)]
print("Listas:")
print((time.time() - start_list) * 1000)
```

mide el tiempo que tarda en sumar dos elementos de los ndarrays

```
start_array = time.time()
result = array1 + array2
print("NumPy ndarray:")
print((time.time() - start_array) * 1000)
```

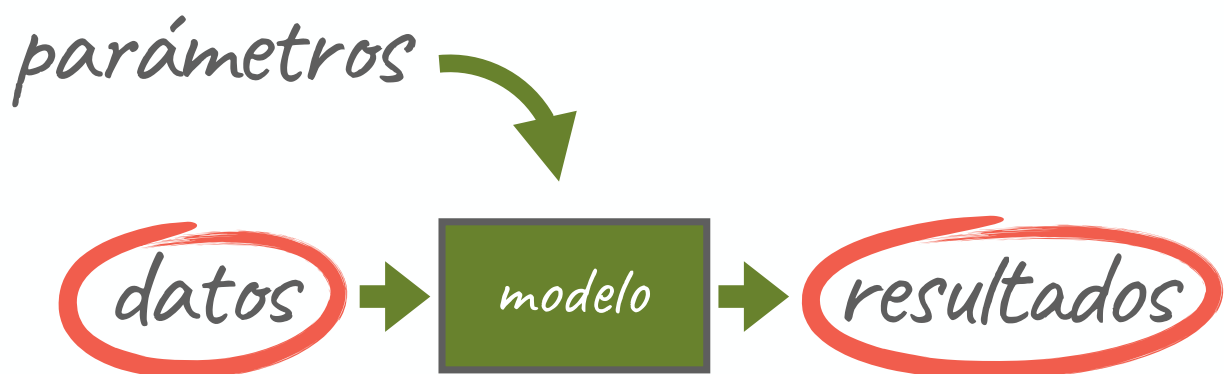
DEMO 7



pandas

*perfecto para el trabajo con
datos en forma tabular*

*¿por qué nos interesan los datos
en investigación operativa?*



pandas

creación de un dataframe con pandas

```
import pandas as pd
```

```
data = {"temp_p": [98.5, 97.4, 97.8, 98.1, 98.3, 97.9],  
        "presion": [2.3, 2.3, 2.1, 2.3, 2.4, 2.1]}
```

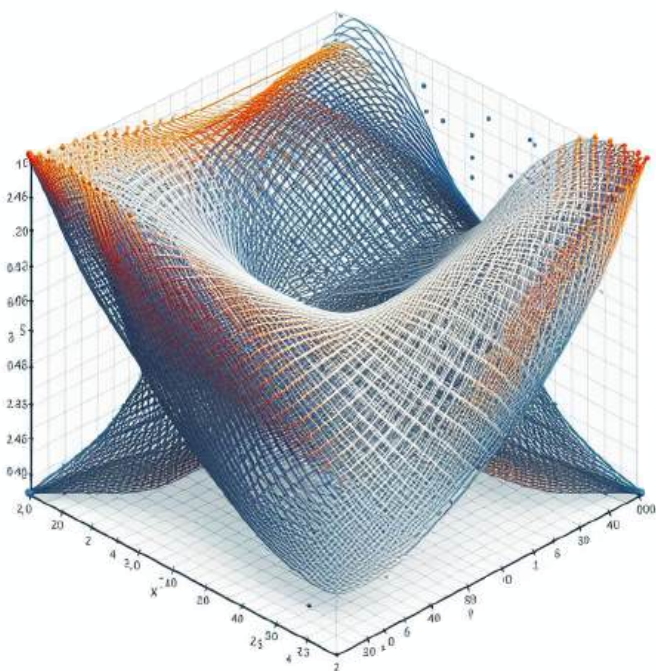
```
df = pd.DataFrame(data, index = ["lote_102", "lote_103", "lote_104",  
                                "lote_105", "lote_106", "lote_107"])
```

```
print(df)
```

funciones para la visualización de los datos

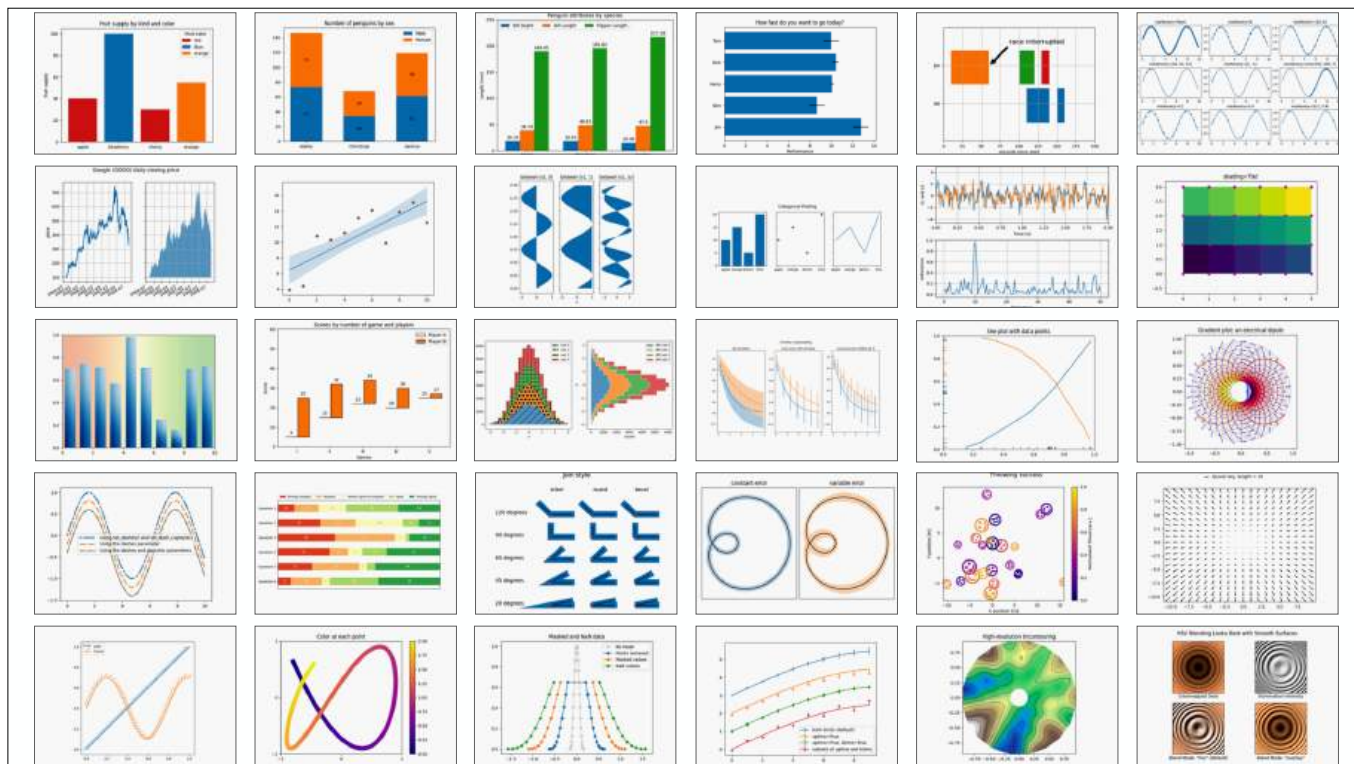
```
print(df.loc["lote_104"])  
print(df.head(2))  
print(df.tail())  
print(df.info())
```

DEMO 8



matplotlib

*perfecto para el trabajo con
datos en forma tabular*



matplotlib

importamos la biblioteca

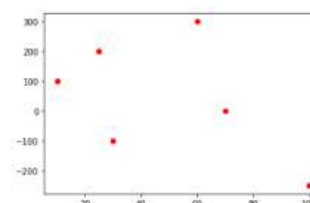
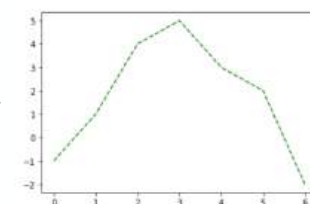
```
import matplotlib.pyplot as plt
%matplotlib inline
```

gráfico lineal simple

```
plt.plot([-1,1,4,5,3,2,-2], "g--")
```

gráfico de coordenadas

```
plt.plot([10,25,30,60,70,100], [100,200,-100,300,0,-250])
```



matplotlib

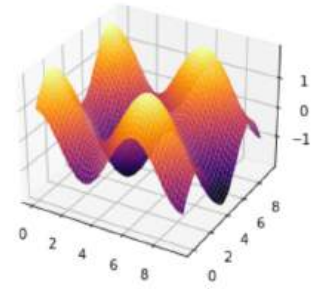
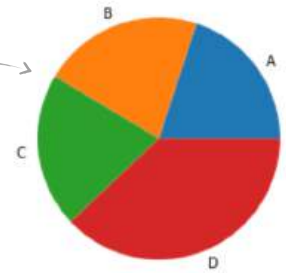
Gráfico de torta

```
plt.pie([121,130,125,230], labels=["A","B","C","D"])
```

Gráfico en 3D

```
from mpl_toolkits.mplot3d import Axes3D
import numpy as np
```

```
fig = plt.figure()
ax = fig.add_subplot(111, projection="3d")
x = np.linspace(0, 3*np.pi)
y = np.linspace(0, 3*np.pi)
X,Y = np.meshgrid(x,y)
Z = np.cos(X) + np.sin(Y)
ax.plot_surface(X, Y, Z, cmap="inferno")
```



DEMO 9



*hay millones de bibliotecas
para realizar distintas cosas*

¿dónde aprender más?

youtube

datacamp

libros

* se incluyen links al final de la presentación

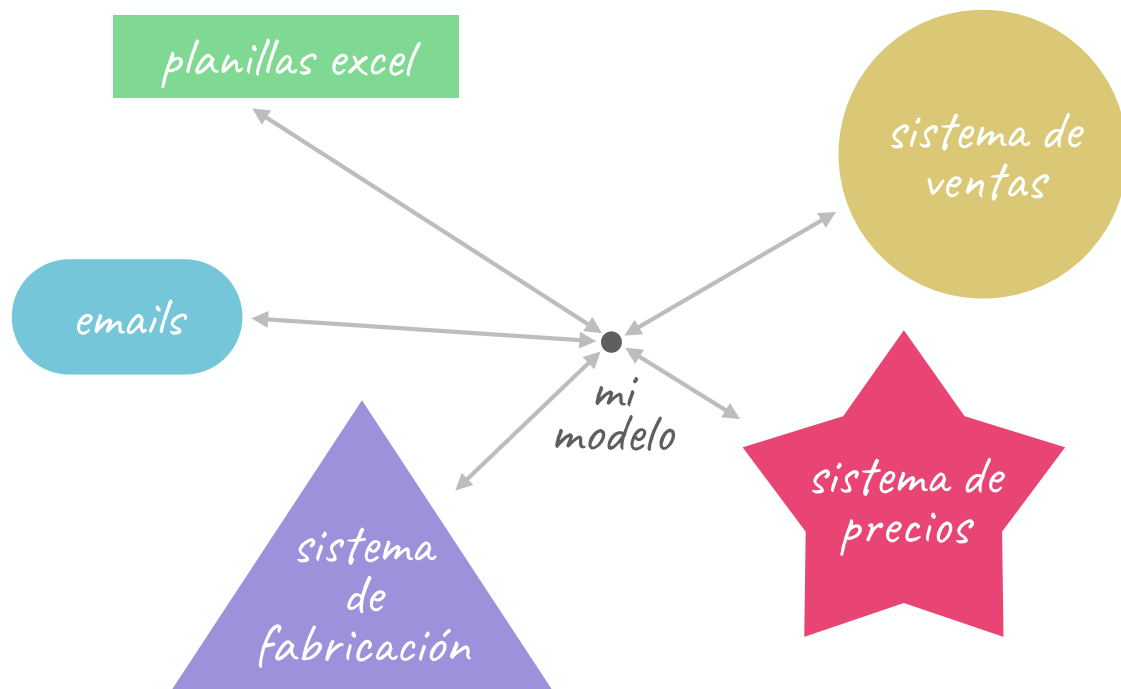
¿ChatGPT?
claro que **sí**



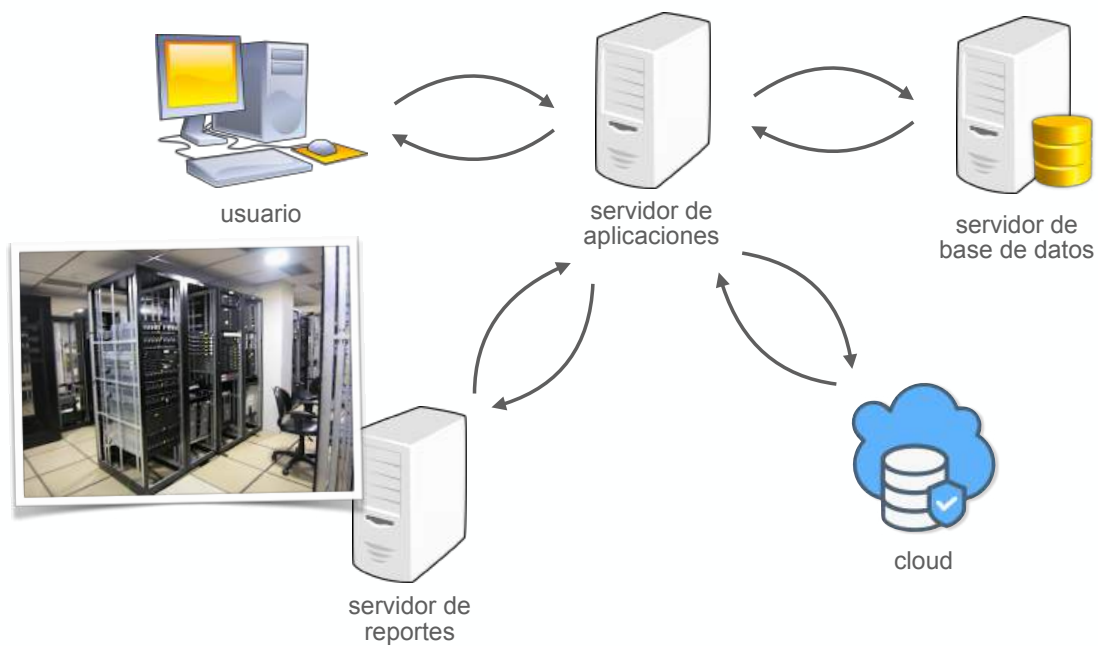
¿preguntas?



integración de modelos a la empresa



¿dónde están los datos?





¿qué es una base de datos?

Una base de datos es un conjunto de datos estructurados y se utiliza para administrar de forma electrónica grandes cantidades de información. Wikipedia.



¿qué es un archivo csv?

Los archivos CSV son un tipo de documento para representar datos en forma de tabla, en las que las columnas se separan por comas. Wikipedia.

ejemplo uso de csv

```
# importamos la biblioteca de pandas
import pandas as pd

# leemos el archivo
df = pd.read_csv('data.csv')

# imprime el dataframe
print(df.to_string())
```

DEMO 10

¿preguntas?



próxima clase:

¡clase virtual sincrónica!

*siempre vamos a utilizar el
mismo link de google meet
por favor utilizar el email de
@fi.uba.ar para ingresar*

*clase bien tecnológica
interfases entre aplicaciones
inteligencia artificial en investigación
operativa*

bibliografía y otros ...

[Python] herramientas:

<https://visualstudio.microsoft.com/es/>
<https://www.anaconda.com/>
<https://www.datawars.io/>
<https://colab.google/>

[Python] para aprender más:

<https://www.w3schools.com>
<https://www.datacamp.com/>
<https://www.youtube.com/>

[Python] bibliotecas:

<https://pandas.pydata.org/>
<https://numpy.org/>
<https://matplotlib.org/>



Inglés



Español

INVESTIGACIÓN OPERATIVA SUPERIOR

¡muchas gracias!