Contenido

[**1. Fundamentos de Python** 1](#_Toc203565209)

[**2. Librerías esenciales** 1](#_Toc203565210)

[**3. Flujo de trabajo y preprocesamiento** 2](#_Toc203565211)

[**4. Lógica y estadística básica** 3](#_Toc203565212)

[**5. Buenas prácticas y razonamiento** 3](#_Toc203565213)

# 1. Fundamentos de Python

* ¿Diferencias entre lista, tupla, set y diccionario?

Una lista es un objeto mutable, admite duplicados

Una tupla no es mutable, admite duplicados

Un conjunto es mutable, no admite duplicados

Un diccionario tiene por elementos a un par llave:valor y la llave no puede ser duplicada, el valor si, es mutable

* ¿Qué hace \*args y \*\*kwargs?\*

Ambos se usan para **pasar una cantidad variable de argumentos** a una función.

\*args Permite pasar **cualquier número de argumentos posicionales** (como una lista o tupla).

def sumar\_todo(\*args):

print(args) # args es una tupla

return sum(args)

print(sumar\_todo(1, 2, 3, 4)) # Salida: 10

\*\*kwargs (keyword arguments):

Permite pasar **cualquier número de argumentos nombrados** (como un diccionario).

🔧 Se recibe como un **diccionario** dentro de la función.

def imprimir\_info(\*\*kwargs):

print(kwargs) # kwargs es un diccionario

for clave, valor in kwargs.items():

print(f"{clave}: {valor}")

imprimir\_info(nombre="Ana", edad=30, ciudad="México")

salida:

{'nombre': 'Ana', 'edad': 30, 'ciudad': 'México'}

nombre: Ana

edad: 30

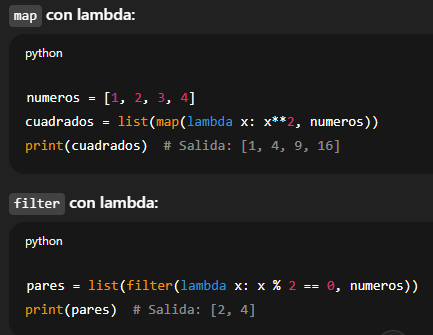
ciudad: México

* ¿Qué es una función lambda y cuándo la usarías?

Una **función lambda** en Python es una función **anónima** (sin nombre) y **compacta**, usada generalmente para operaciones simples y rápidas. Se define con la palabra clave lambda.

sumar = lambda x, y: x + y

print(sumar(3, 5)) # Salida: 8

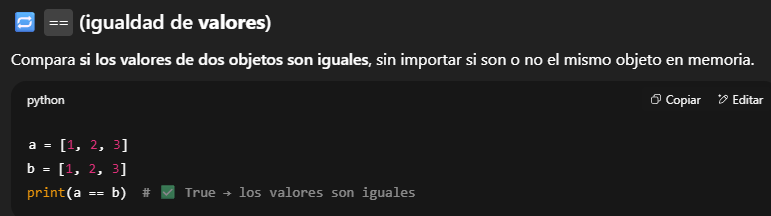


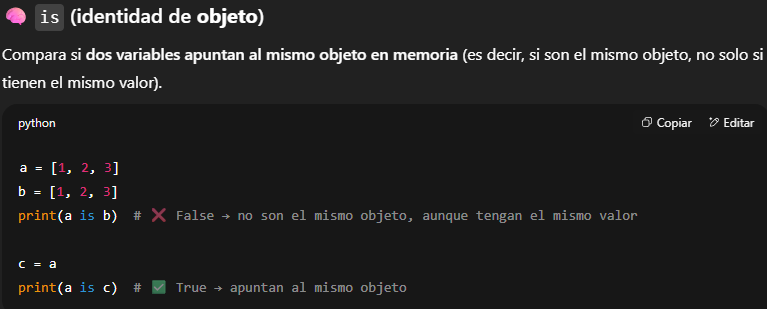
Cuándo **no** usar lambda:

* Cuando la función es compleja o extensa.
* Cuando se requiere reutilizar el código muchas veces (mejor definirla con def).
* ¿Qué implica que una estructura sea mutable?

Significa que puedo cambiar o eliminar los valores de los elementos de la estructura una vez que esta ha sido creada

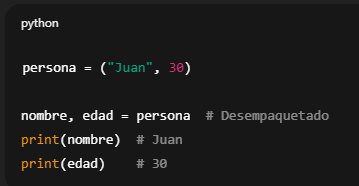
* ¿Qué diferencia hay entre is y ==?





* ¿Qué es el desempaquetado de variables?

El **desempaquetado de variables** en Python es una forma de **asignar múltiples valores a múltiples variables de forma simultánea**, de manera clara y elegante, especialmente cuando trabajas con listas, tuplas u otras secuencias.



# 2. Librerías esenciales

**NumPy**

¿Qué ventaja tiene NumPy frente a listas de Python?

La ventaja es que es una librería especializada y pensada para cálculo numérico y arreglos. Además tiene una flexibilidad en la creación de arreglos

¿Qué significa vectorización?

¿Qué diferencias hay entre listas de Python y arrays de NumPy?

¿Qué es el broadcasting y por qué es importante en operaciones con arrays?

¿Cómo inicializarías una matriz de 5x5 con valores aleatorios entre 0 y 1?

¿Qué hace la función np.where() y cómo se usa?

¿Cuál es la diferencia entre reshape() y ravel()?

¿Cómo verificarías si un array tiene valores NaN o infinitos?

¿Qué sucede si intentas sumar dos arrays con formas incompatibles?

# pandas

¿Qué es una Serie y qué es un DataFrame?

¿Cómo seleccionas filas por condición?

¿Cómo manejas valores nulos?

¿Qué hace groupby() y cuándo se usa?

¿Cómo haces un merge entre dos DataFrames?

¿Cuál es la diferencia entre df.loc[] y df.iloc[]?

¿Cómo manejarías duplicados en un DataFrame y cómo decides si eliminarlos?

¿Qué hace el método .groupby() y en qué situaciones es útil?

¿Cómo puedes combinar o unir dos DataFrames? Menciona diferencias entre merge(), join() y concat().

¿Cómo filtrarías las filas de un DataFrame con múltiples condiciones lógicas?

¿Qué hace el método .apply() y cuándo lo usarías en lugar de un bucle for?

¿Cómo convertirías una columna con fechas en un índice temporal y qué beneficios tiene hacerlo?

# Matplotlib/Seaborn

* ¿Diferencias entre Matplotlib y Seaborn?
* ¿Qué gráfico usarías para:  
  a) distribución,  
  b) correlación,  
  c) comparación de categorías?

 ¿Cómo cambiarías el tamaño de una figura en Matplotlib?

 ¿Qué función se usa para agregar etiquetas a los ejes y un título a la gráfica?

 ¿Cómo graficarías múltiples líneas en un solo gráfico y cómo distinguirlas visualmente?

 ¿Qué ventajas ofrece usar subplot() y cuándo lo usarías?

 ¿Cómo cambiarías el estilo de línea y color en una gráfica de líneas?

 ¿Cuál es la diferencia entre plt.plot(), plt.bar(), y plt.hist()?

 ¿Qué técnicas usarías para hacer una visualización más clara si tienes muchos datos en un mismo gráfico?

# 3. Flujo de trabajo y preprocesamiento

* ¿Cómo cargas un CSV y ves las primeras filas?
* ¿Cómo identificas y tratas valores faltantes?
* ¿Qué es codificación one-hot? ¿Cuándo la usas?
* ¿Qué diferencia hay entre normalización y estandarización?
* ¿Cómo filtras, ordenas y renombrar columnas con pandas?

 ¿Qué pasos considerarías antes de entrenar un modelo con tus datos?

 ¿Cómo manejarías valores faltantes? ¿Qué criterios usarías para imputarlos o eliminarlos?

 ¿Por qué es importante normalizar o escalar los datos antes de usar ciertos modelos?

 ¿Cuál es la diferencia entre codificación ordinal y codificación one-hot? ¿Cuándo usar cada una?

 ¿Cómo detectarías y manejarías outliers?

 ¿Qué es el leakage de datos y cómo lo evitarías en el preprocesamiento?

 ¿Cuál es la diferencia entre transformar y estandarizar datos?

# 4. Lógica y estadística básica

* ¿Cómo calculas media, mediana y std en pandas?
* ¿Qué es la correlación y cómo la obtienes?
* ¿Qué es un outlier y cómo lo detectarías?
* ¿Qué diferencia hay entre apply() y map()?
* ¿Qué hace value\_counts() y para qué sirve?

# 5. Buenas prácticas y razonamiento

* ¿Qué es código reproducible?
* ¿Por qué es mejor vectorizar que usar bucles?
* ¿Qué significa documentar bien tu código?
* ¿Cómo dividirías un proyecto de análisis en etapas claras?
* ¿Qué es un pipeline y cómo ayuda en ciencia de datos?

 ¿Por qué es importante documentar tu código y cómo lo haces normalmente?

 ¿Cuál es el propósito de usar control de versiones como Git en proyectos de ciencia de datos?

 ¿Qué es un pipeline de datos y por qué es útil implementarlo?

 ¿Qué consideraciones tienes al trabajar con datos sensibles o privados?

 ¿Por qué es importante la reproducibilidad en ciencia de datos y cómo la garantizas?

 ¿Qué errores comunes has encontrado al trabajar con notebooks de Jupyter y cómo los evitas?

 ¿Por qué deberías evitar modificar un DataFrame directamente sin hacer una copia explícita?