

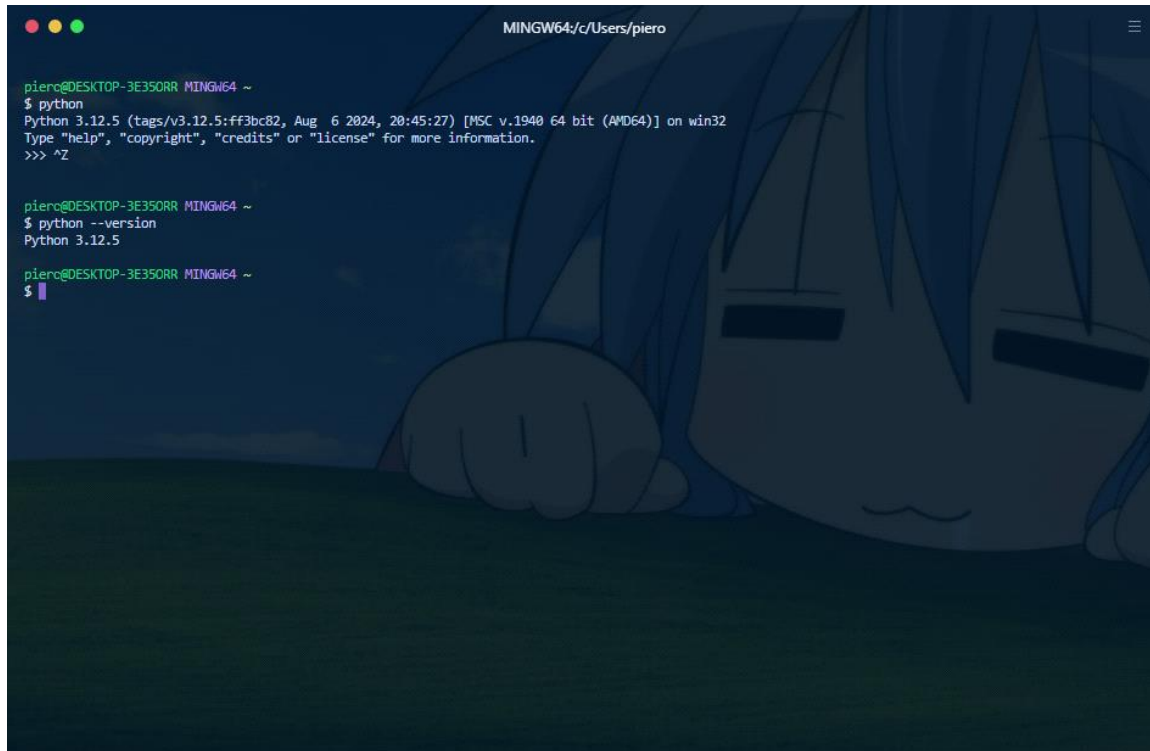
ANEXOS: FUNDAMENTOS DE MANIPULACIÓN DE DATOS CON PANDAS Y NUMPY

SESION 1 / SEMANA 1

Anexo 01: Tarea Aprendizaje 01

Alumno: [Juan Piero Vincha Loza]

1.- Capturar pantalla de la version python (3 ptos)

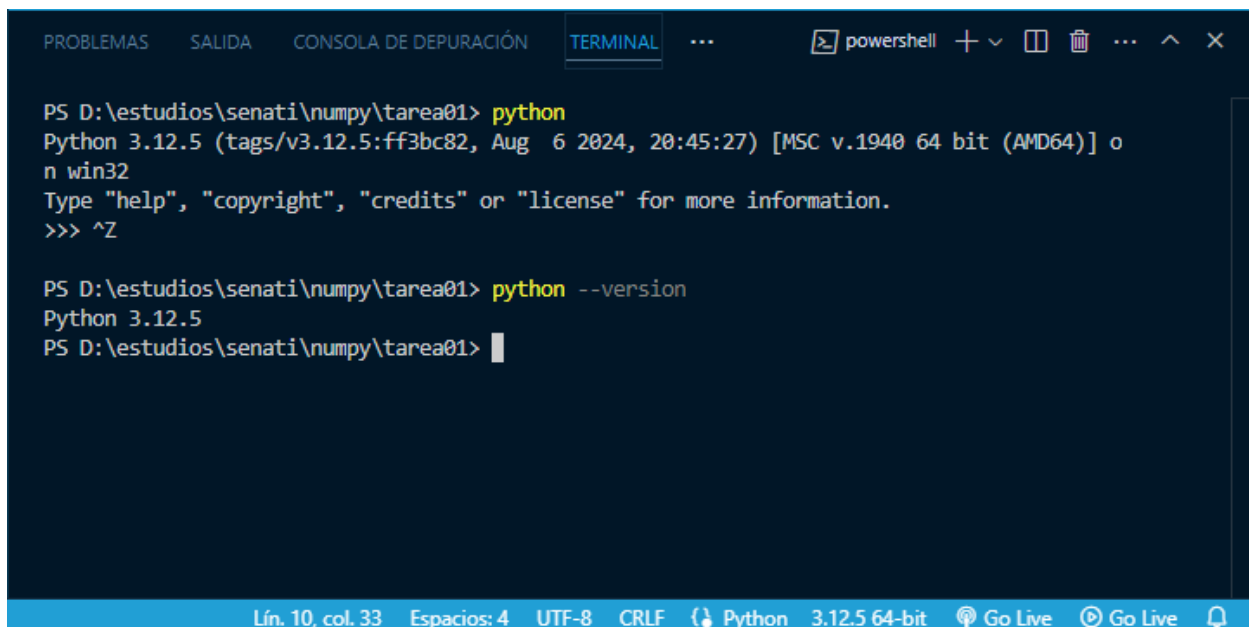


```
MINGW64/c/Users/piero

piero@DESKTOP-3E35ORR MINGW64 ~
$ python
Python 3.12.5 (tags/v3.12.5:ff3bc82, Aug 6 2024, 20:45:27) [MSC v.1940 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license" for more information.
>>> ^Z

piero@DESKTOP-3E35ORR MINGW64 ~
$ python --version
Python 3.12.5

piero@DESKTOP-3E35ORR MINGW64 ~
$
```



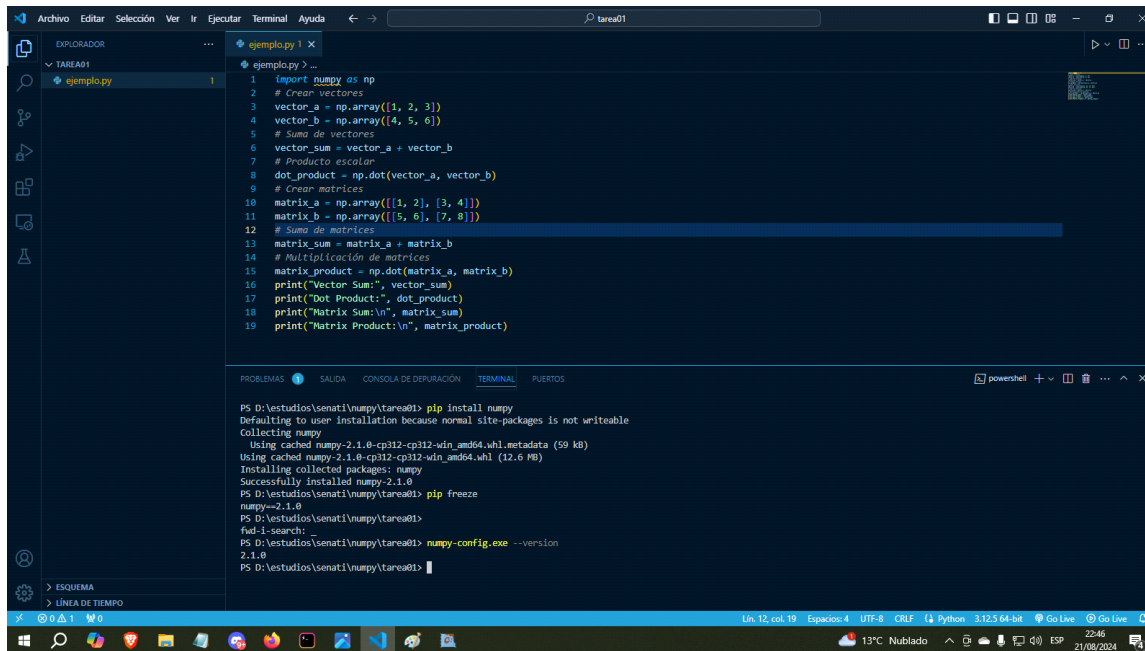
```
PROBLEMAS  SALIDA  CONSOLA DE DEPURACIÓN  TERMINAL  ...  powershell + v  [ ] [ ] ... ^ X

PS D:\estudios\senati\numpy\tarea01> python
Python 3.12.5 (tags/v3.12.5:ff3bc82, Aug 6 2024, 20:45:27) [MSC v.1940 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license" for more information.
>>> ^Z

PS D:\estudios\senati\numpy\tarea01> python --version
Python 3.12.5
PS D:\estudios\senati\numpy\tarea01>

Lín. 10, col. 33  Espacios: 4  UTF-8  CRLF  Python 3.12.5 64-bit  Go Live  Go Live
```

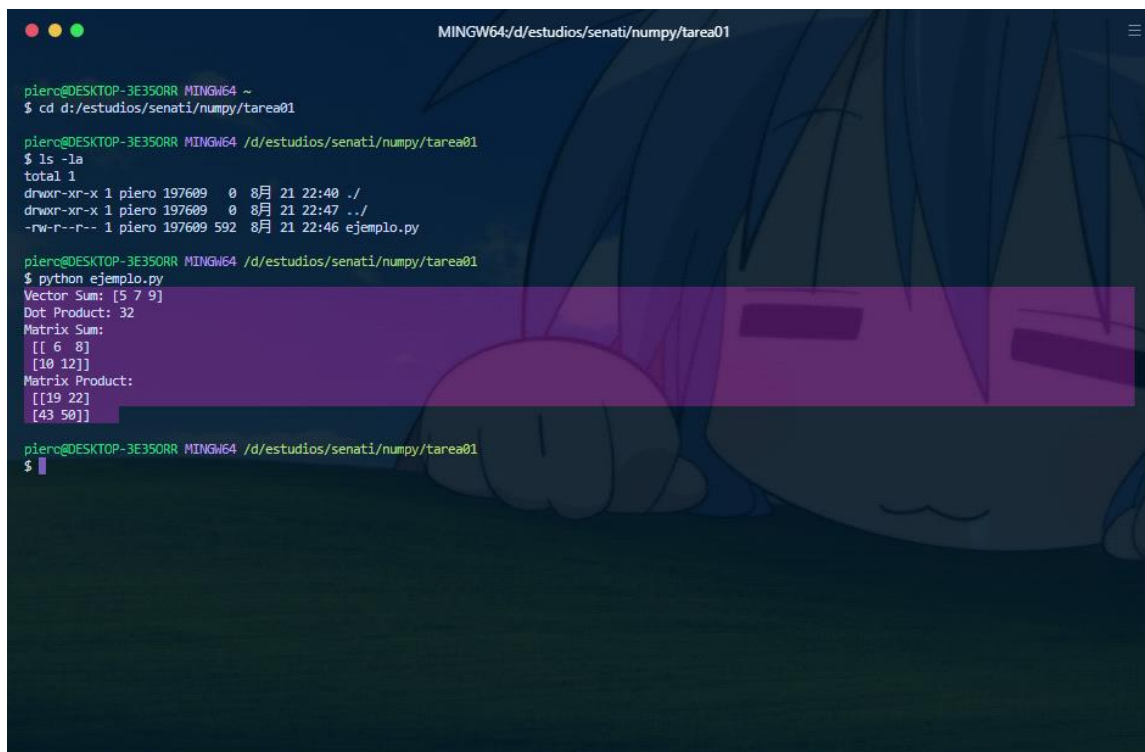
2.- Instalar el paquete numpy con pip install numpy y ejecutar en el terminal pip freeze para ver que version de numpy tiene (4 pts)



```
ejemplo.py
1 import numpy as np
2 # Crear vectores
3 vector_a = np.array([1, 2, 3])
4 vector_b = np.array([4, 5, 6])
5 # Suma de vectores
6 vector_sum = vector_a + vector_b
7 # Producto escalar
8 dot_product = np.dot(vector_a, vector_b)
9 # Crear matrices
10 matrix_a = np.array([[1, 2], [3, 4]])
11 matrix_b = np.array([[5, 6], [7, 8]])
12 # Suma de matrices
13 matrix_sum = matrix_a + matrix_b
14 # Multiplicación de matrices
15 matrix_product = np.dot(matrix_a, matrix_b)
16 print("Vector Sum:", vector_sum)
17 print("Dot Product:", dot_product)
18 print("Matrix Sum:\n", matrix_sum)
19 print("Matrix Product:\n", matrix_product)

TERMINAL
PS D:\estudios\senati\numpy\tarea01> pip install numpy
Defaulting to user installation because normal site-packages is not writeable
Collecting numpy
  Using cached numpy-2.1.0-cp312-cp312-win_amd64.whl.metadata (59 kB)
Using cached numpy-2.1.0-cp312-cp312-win_amd64.whl (12.6 MB)
Installing collected packages: numpy
Successfully installed numpy-2.1.0
PS D:\estudios\senati\numpy\tarea01> pip freeze
numpy==2.1.0
PS D:\estudios\senati\numpy\tarea01>
PS D:\estudios\senati\numpy\tarea01>
PS D:\estudios\senati\numpy\tarea01> numpy-config.exe --version
2.1.0
PS D:\estudios\senati\numpy\tarea01>
```

3.-Ejecutar el código y mostrar el resultado (7 pts)



```
MINGW64/d/estudios/senati/numpy/tarea01

piero@DESKTOP-3E3S0RR MINGW64 ~
$ cd d:/estudios/senati/numpy/tarea01

piero@DESKTOP-3E3S0RR MINGW64 /d/estudios/senati/numpy/tarea01
$ ls -la
total 1
drwxr-xr-x 1 piero 197609  0  8月 21 22:40 ./
drwxr-xr-x 1 piero 197609  0  8月 21 22:47 ../
-rw-r--r-- 1 piero 197609 592  8月 21 22:46 ejemplo.py

piero@DESKTOP-3E3S0RR MINGW64 /d/estudios/senati/numpy/tarea01
$ python ejemplo.py
Vector Sum: [5 7 9]
Dot Product: 32
Matrix Sum:
[[ 6  8]
 [10 12]]
Matrix Product:
[[19 22]
 [43 50]]

piero@DESKTOP-3E3S0RR MINGW64 /d/estudios/senati/numpy/tarea01
$
```

4.- Cual es la diferencia entre vector y un matriz ? (2 PTOS)

Diferencia entre Vector y Matriz

1. Definición:

- Vector: Un vector es una estructura unidimensional que representa una lista ordenada de elementos. En matemáticas y programación, un vector puede ser visto como una secuencia de números en una sola dimensión. Ejemplo: `[1, 2, 3]`.

- Matriz: Una matriz es una estructura bidimensional que organiza los datos en filas y columnas. Cada elemento se encuentra en una posición específica identificada por dos índices (fila y columna). Ejemplo: `[[1, 2], [3, 4]]`.

2. Dimensionalidad:

- Vector: Tiene una sola dimensión (número de elementos en una fila o columna).

- Matriz: Tiene dos dimensiones (número de filas y número de columnas).

5.- Cual menciona 5 metodos que tenga pandas y 5 metodos que tenga numpy que hacen cada uno(4 PTOS)

Métodos en Pandas

1. `DataFrame.head(n)`: Devuelve las primeras `n` filas del DataFrame. Útil para inspeccionar los primeros registros del DataFrame.

2. `DataFrame.describe()`: Proporciona estadísticas descriptivas sobre las columnas numéricas del DataFrame, como la media, desviación estándar, mínimo y máximo.

3. `DataFrame.groupby(by)`: Agrupa el DataFrame por una o más columnas, permitiendo aplicar funciones agregadas a cada grupo.

4. `DataFrame.merge(right, how='inner', on=None)` : Realiza una fusión de DataFrames similar a una operación JOIN en SQL, basándose en una o más columnas.
5. `DataFrame.drop(labels, axis=0 or 1)` : Elimina filas o columnas del DataFrame basándose en las etiquetas especificadas.

Métodos en NumPy

1. `numpy.array(object)` : Crea un array de NumPy a partir de un objeto iterable, como una lista o tupla.
2. `numpy.dot(a, b)` : Realiza el producto punto entre dos arrays. Si ambos arrays son bidimensionales, realiza la multiplicación de matrices.
3. `numpy.mean(a, axis=None)` : Calcula la media aritmética de los elementos a lo largo del eje especificado o de todos los elementos si no se especifica un eje.
4. `numpy.reshape(a, newshape)` : Cambia la forma del array sin modificar sus datos. Por ejemplo, convierte un array de 1D en una matriz de 2D.
5. `numpy.linspace(start, stop, num)` : Genera un array con `num` valores igualmente espaciados entre `start` y `stop`, útil para crear secuencias numéricas.