

Vectores3

September 18, 2024

Universidad Autónoma del Estado de México

Centro Universitario UAEM Zumpango

Ingeniería en Computación

Graficacion Computacional

Alumno: Diego Argel Navarrete Godines

Profesora: Hazem Álvarez Rodríguez

Fecha: 18 de septiembre del 2024

Descripcion: Vectores con arrays

1 Se importa la libreria numpy (soporte para vectores), matplotlib (graficador)

```
[ ]: import numpy as np
import matplotlib.pyplot as plt
```

```
[46]: # Crea un campo vectorial con un solo vector
plt.quiver(1, 2, 3, 4, scale_units='xy', angles='xy', scale=1)

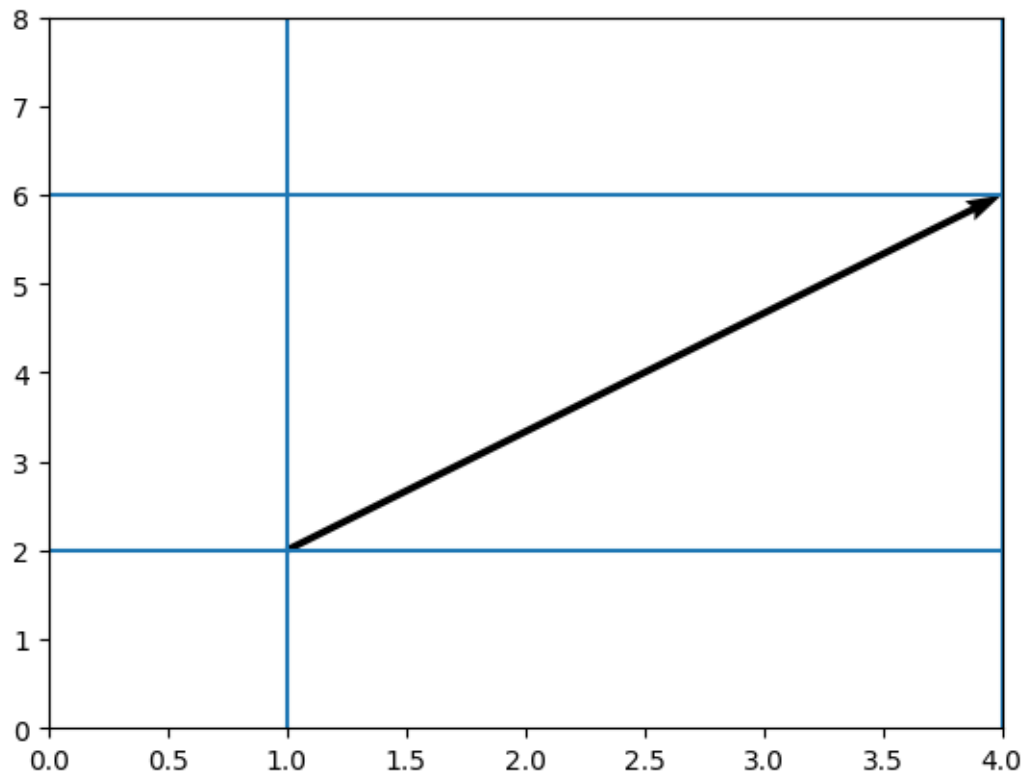
# Establece los límites del eje x
plt.xlim(0, 4)

# Establece los límites del eje y
plt.ylim(0, 8)

# Dibuja líneas verticales
plt.axvline(x=1) # Línea vertical en x=1
plt.axvline(x=1+3) # Línea vertical en x=4

# Dibuja líneas horizontales
plt.axhline(y=2) # Línea horizontal en y=2
plt.axhline(y=2+4) # Línea horizontal en y=6
```

```
[46]: <matplotlib.lines.Line2D at 0x25e11ad4d40>
```

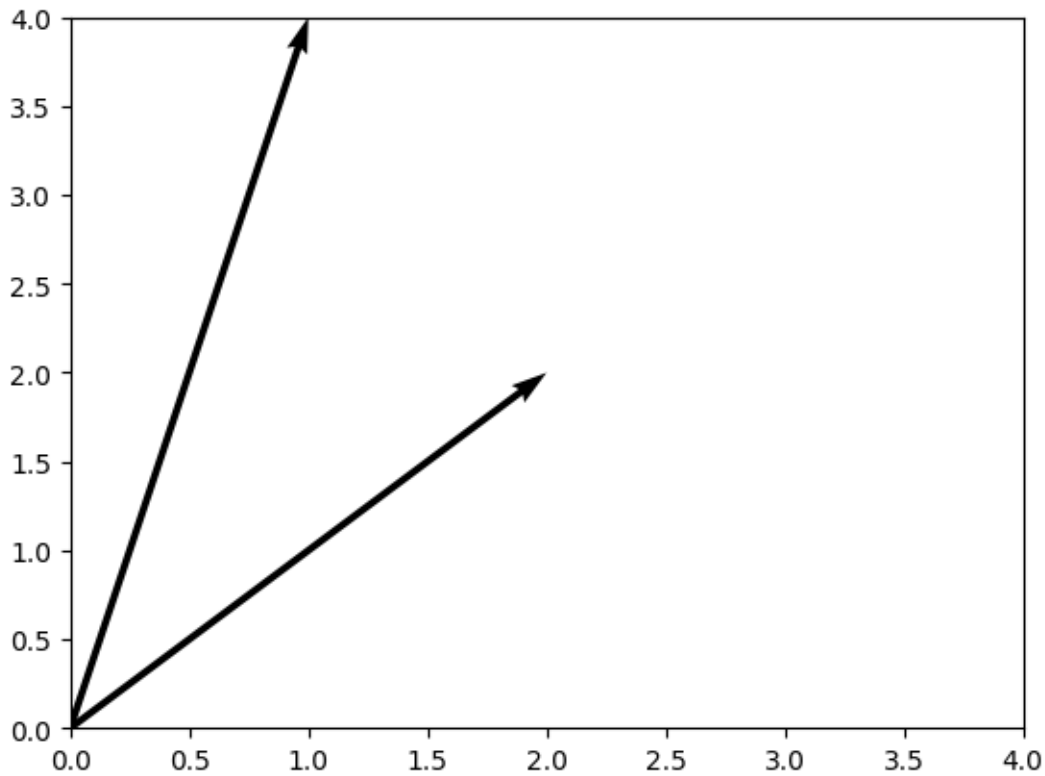


```
[47]: # Crea dos vectores
plt.quiver([0, 0], [0, 0], [1, 2], [4, 2], scale_units='xy', angles='xy',
↪scale=1)

# Establece los límites del eje x
plt.xlim(0, 4)

# Establece los límites del eje y
plt.ylim(0, 4)
```

[47]: (0.0, 4.0)



```
[48]: # Crea dos vectores
plt.quiver([1, 0], [2, 0], [1, 2], [3, 2], scale_units='xy', angles='xy',
           ↪scale=1)

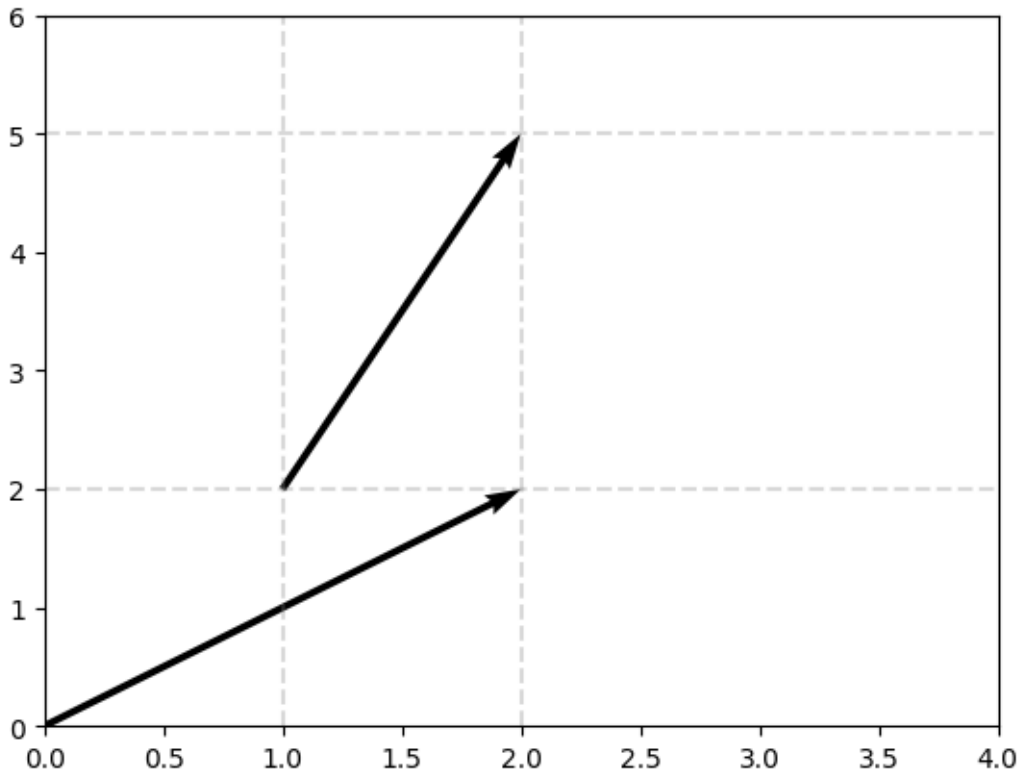
# Establece los límites del eje x
plt.xlim(0, 4)

# Establece los límites del eje y
plt.ylim(0, 6)

# Dibuja líneas verticales
plt.axvline(x=1, color='gray', linestyle='--', alpha=0.3)
plt.axvline(x=2, color='gray', linestyle='--', alpha=0.3)

# Dibuja líneas horizontales
plt.axhline(y=2, color='gray', linestyle='--', alpha=0.3)
plt.axhline(y=5, color='gray', linestyle='--', alpha=0.3)
```

```
[48]: <matplotlib.lines.Line2D at 0x25e1168b2c0>
```



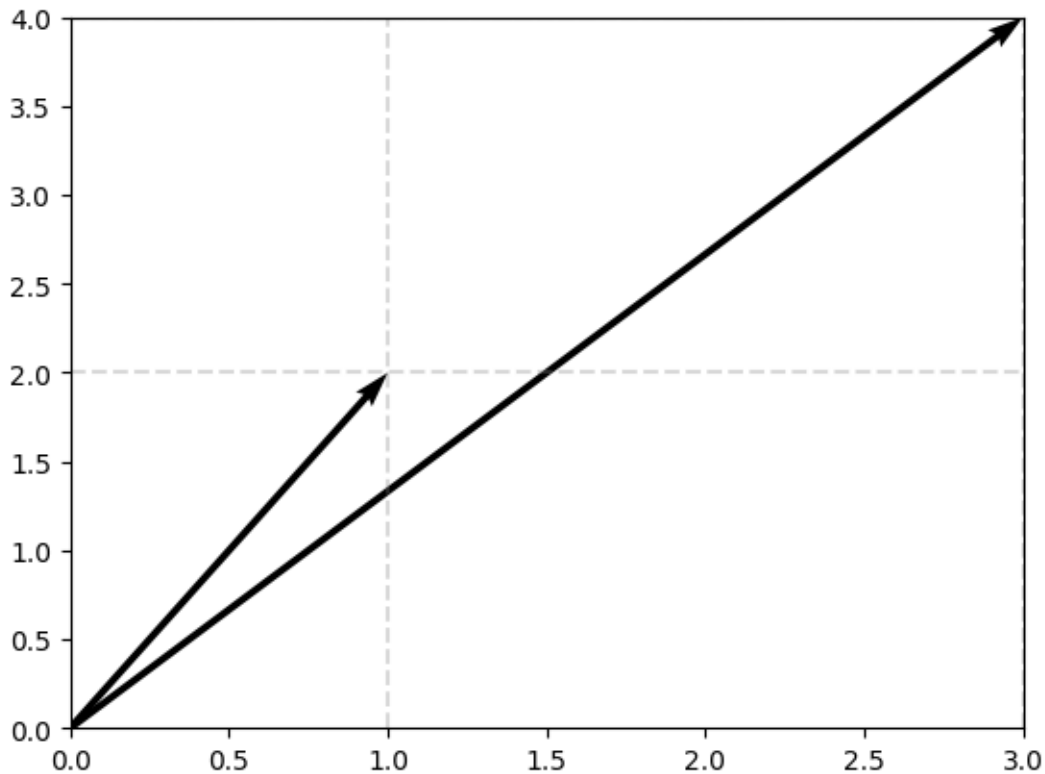
```
[26]: # Definimos dos vectores
vector1 = np.array([1, 2])
vector2 = np.array([3, 4])

# Creamos un gráfico de vectores
plt.quiver([0, 0], [0, 0], [vector1[0], vector2[0]], [vector1[1], vector2[1]],
           angles='xy', scale_units='xy', scale=1)

# Establecemos los límites de los ejes
plt.xlim(0, max(vector1[0], vector2[0]))
plt.ylim(0, max(vector1[1], vector2[1]))

# Dibujamos líneas verticales y horizontales
plt.axvline(x=1, color='gray', linestyle='--', alpha=0.3)
plt.axvline(x=3, color='gray', linestyle='--', alpha=0.3)
plt.axhline(y=2, color='gray', linestyle='--', alpha=0.3)
plt.axhline(y=4, color='gray', linestyle='--', alpha=0.3)
```

```
[26]: <matplotlib.lines.Line2D at 0x25e1307a120>
```



```
[49]: def grafvecs(vecs, cols): # Esta función crea un gráfico de vectores utilizando
    ↪ la biblioteca Matplotlib.
        # vecs: Una lista de vectores, donde cada vector es una lista o array
    ↪ de NumPy con dos elementos (x, y).
        # cols: Una lista de colores, correspondiente a cada vector.

    plt.figure(figsize=(5, 5)) # Crea una figura de 5x5 pulgadas

    # Itera sobre cada vector y su color correspondiente
    for i in range(len(vecs)):
        vec = vecs[i]
        color = cols[i]

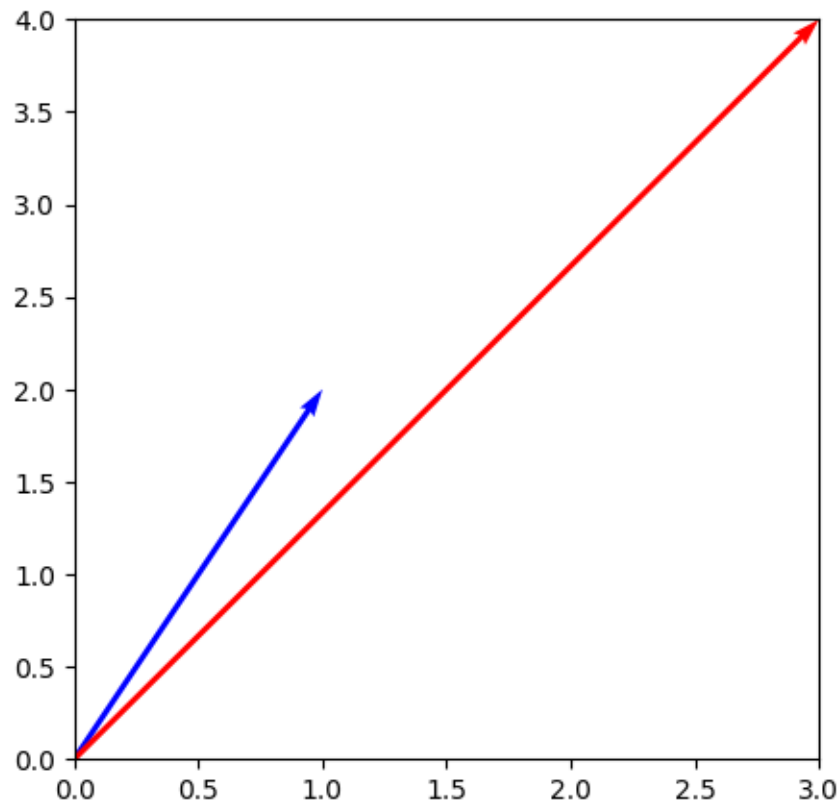
        # Dibuja un vector desde el origen (0, 0) hasta el punto definido por
    ↪ el vector
        plt.quiver(0, 0, vec[0], vec[1], color=color, angles='xy',
    ↪ scale_units='xy', scale=1)
```

```
[50]: # Definimos dos vectores
vector1 = np.array([1, 2])
vector2 = np.array([3, 4])
```

```
# Llamamos a la función grafvecs para crear un gráfico de vectores
grafvecs([vector1, vector2], ['blue', 'red'])

# Establecemos los límites de los ejes para una mejor visualización
plt.xlim(0, max(vector1[0], vector2[0]))
plt.ylim(0, max(vector1[1], vector2[1]))
```

[50]: (0.0, 4.0)



```
[51]: # Definimos tres vectores
vector1 = np.array([1, 2])
vector2 = np.array([3, 4])
vector3 = np.array([8, 4])

# Llamamos a la función grafvecs para crear un gráfico de vectores
grafvecs([vector1, vector2, vector3], ['blue', 'red', 'green'])

# Establecemos los límites de los ejes para una mejor visualización
plt.xlim(0, max(vector1[0], vector2[0], vector3[0]))
plt.ylim(0, max(vector1[1], vector2[1], vector3[1]))
```

[51]: (0.0, 4.0)

