

# Vectores

September 28, 2024

## 1 Graficar Vectores

1.0.1 Trabajo realizado por: Jessica Naomi Millan Sánchez

1.0.2 Graficación Computacional

1.0.3 Profesora: Hazem Álvarez Rodríguez

1.0.4 Clase del 09 de septiembre de 2024

Graficar los vectores realizados en clase

1.0.5 Ejercicio 1:

Cuadrante 1

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

```
[44]: # Coordenadas que abarca el mapa cartesiano
x1=-0.8
x2=8
y1=-0.8
y2=8

# Crear el eje
plt.axis([x1,x2,y1,y2])

# Habilitar el grid
plt.grid(True)
plt.axis('on')
plt.title('Cuadrante 1')

# Generar puntos
dx=2
dy=2
for x in np.arange(x1,x2,dx):
    for y in np.arange(y1,y2,dy):
        plt.scatter(x,y,s=1.5, color='lightgray')
```

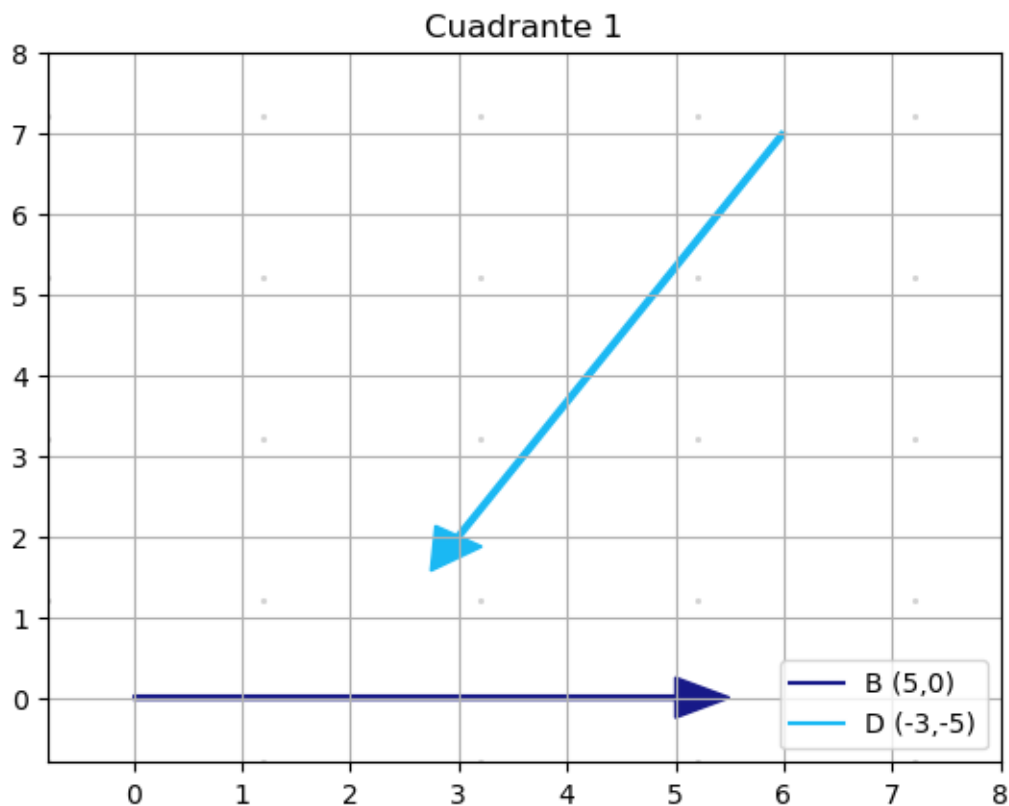
```

# x, y, incremento, abcisa, longitud, ancho, color
plt.arrow(6,7,-3,-5,head_length=0.5, head_width=0.5, color=(26/255,184/255,243/
↪255), width=0.05)
plt.arrow(0,0,5,0,head_length=0.5, head_width=0.5, color=(23/255,25/255,136/
↪255), width=0.05)

# Crear una línea 'invisible' para mostrar la simbología o leyenda
plt.plot([], [], color=(23/255, 25/255, 136/255), label='B (5,0)')
plt.plot([], [], color=(26/255, 184/255, 243/255), label='D (-3,-5)')
plt.legend(loc='lower right')

```

[44]: <matplotlib.legend.Legend at 0x7da7d8799a30>



## Cuadrante 2

[45]: *# Coordenadas que abarca el mapa cartesiano*

```

x2=0
x1=-8
y1=0
y2=8

```

```

# Crear el eje
plt.axis([x1,x2,y1,y2])

# Habilitar el grid
plt.grid(True)
plt.axis('on')
plt.title('Cuadrante 2')

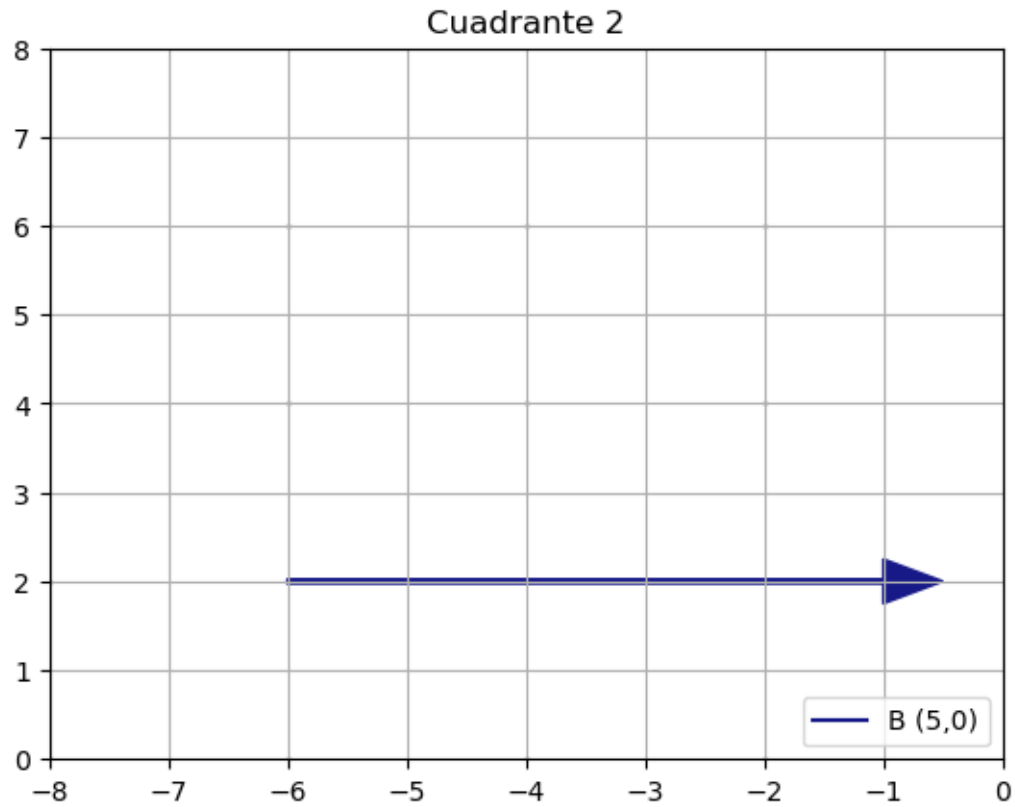
# Generar puntos
dx=2
dy=2
for x in np.arange(x1,x2,dx):
    for y in np.arange(y1,y2,dy):
        plt.scatter(x,y,s=1.5, color='lightgray')

        # x, y, incremento, abcisa, longitud, ancho, color
plt.arrow(-6,2,5,0,head_length=0.5, head_width=0.5, color=(23/255,25/255,136/
↪255), width=0.05)

# Crear una línea 'invisible' para mostrar la simbología o leyenda
plt.plot([], [], color=(23/255, 25/255, 136/255), label='B (5,0)')
plt.legend(loc='lower right')

```

[45]: <matplotlib.legend.Legend at 0x7da7d8fd8920>



### Cuadrante 3

[43]: *# Coordenadas que abarca el mapa cartesiano*

```
x1=-8
```

```
x2=0
```

```
y1=-8
```

```
y2=0
```

```
# Crear el eje
```

```
plt.axis([x1,x2,y1,y2])
```

```
# Habilitar el grid
```

```
plt.grid(True)
```

```
plt.axis('on')
```

```
plt.title('Cuadrante 3')
```

```
# Generar puntos
```

```
dx=2
```

```
dy=2
```

```
for x in np.arange(x1,x2,dx):
```

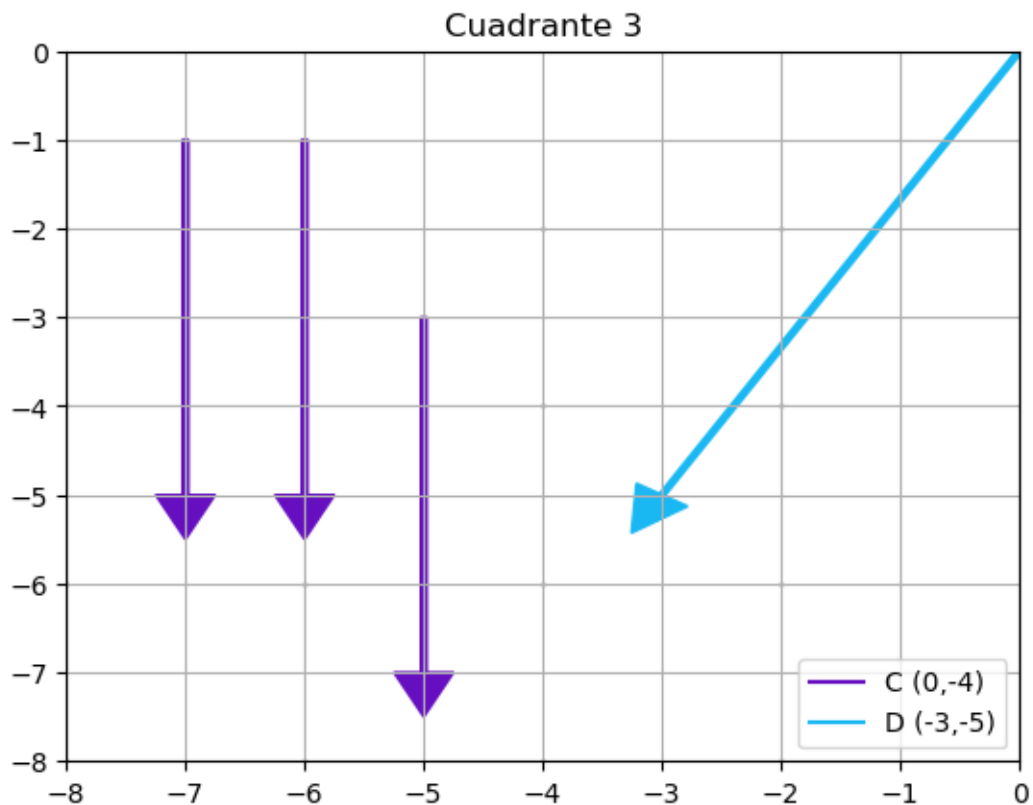
```
    for y in np.arange(y1,y2,dy):
```

```
plt.scatter(x,y,s=1.5, color='lightgray')

# x, y, incremento, abcisa, longitud, ancho, color
plt.arrow(0,0,-3,-5,head_length=0.5, head_width=0.5, color=(26/255,184/255,243/
↪255), width=0.05)
plt.arrow(-7,-1,0,-4,head_length=0.5, head_width=0.5, color=(101/255,15/255,193/
↪255), width=0.05)
plt.arrow(-6,-1,0,-4,head_length=0.5, head_width=0.5, color=(101/255,15/255,193/
↪255), width=0.05)
plt.arrow(-5,-3,0,-4,head_length=0.5, head_width=0.5, color=(101/255,15/255,193/
↪255), width=0.05)

# Crear una línea 'invisible' para mostrar la simbología o leyenda
plt.plot([], [], color=(101/255,15/255,193/255), label='C (0,-4)')
plt.plot([], [], color=(26/255, 184/255, 243/255), label='D (-3,-5)')
plt.legend(loc='lower right')
```

[43]: <matplotlib.legend.Legend at 0x7da7d8d6f860>



Cuadrante 4

```
[46]: # Coordenadas que abarca el mapa cartesiano
x1=0
x2=9
y2=0
y1=-9

# Crear el eje
plt.axis([x1,x2,y1,y2])

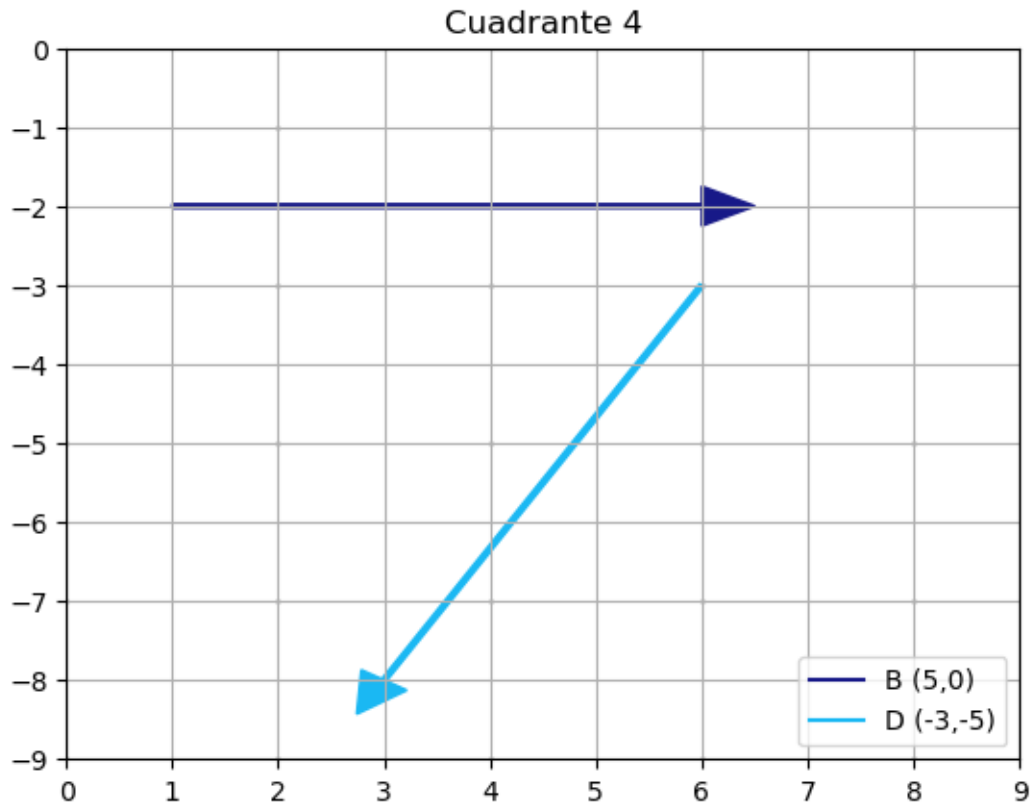
# Habilitar el grid
plt.grid(True)
plt.axis('on')
plt.title('Cuadrante 4')

# Generar puntos
dx=2
dy=2
for x in np.arange(x1,x2,dx):
    for y in np.arange(y1,y2,dy):
        plt.scatter(x,y,s=1.5, color='lightgray')

        # x, y, incremento, abcisa, longitud, ancho, color
plt.arrow(6,-3,-3,-5,head_length=0.5, head_width=0.5, color=(26/255,184/255,243/
↪255), width=0.05)
plt.arrow(1,-2,5,0,head_length=0.5, head_width=0.5, color=(23/255,25/255,136/
↪255), width=0.05)

# Crear una línea 'invisible' para mostrar la simbología o leyenda
plt.plot([], [], color=(23/255, 25/255, 136/255), label='B (5,0)')
plt.plot([], [], color=(26/255, 184/255, 243/255), label='D (-3,-5)')
plt.legend(loc='lower right')
```

[46]: <matplotlib.legend.Legend at 0x7da7d90f1a30>



### Plano Cartesiano Completo

[47]: *# Coordenadas que abarca el mapa cartesiano*

```
x1=-9
x2=9
y1=-9
y2=9

# Crear el eje
plt.axis([x1,x2,y1,y2])

# Habilitar el grid
plt.grid(True)
plt.axis('on')
plt.title('Plano Cartesiano 1')

# Generar puntos
dx=2
dy=2
for x in np.arange(x1,x2,dx):
    for y in np.arange(y1,y2,dy):
```

```

plt.scatter(x,y,s=1.5, color='lightgray')

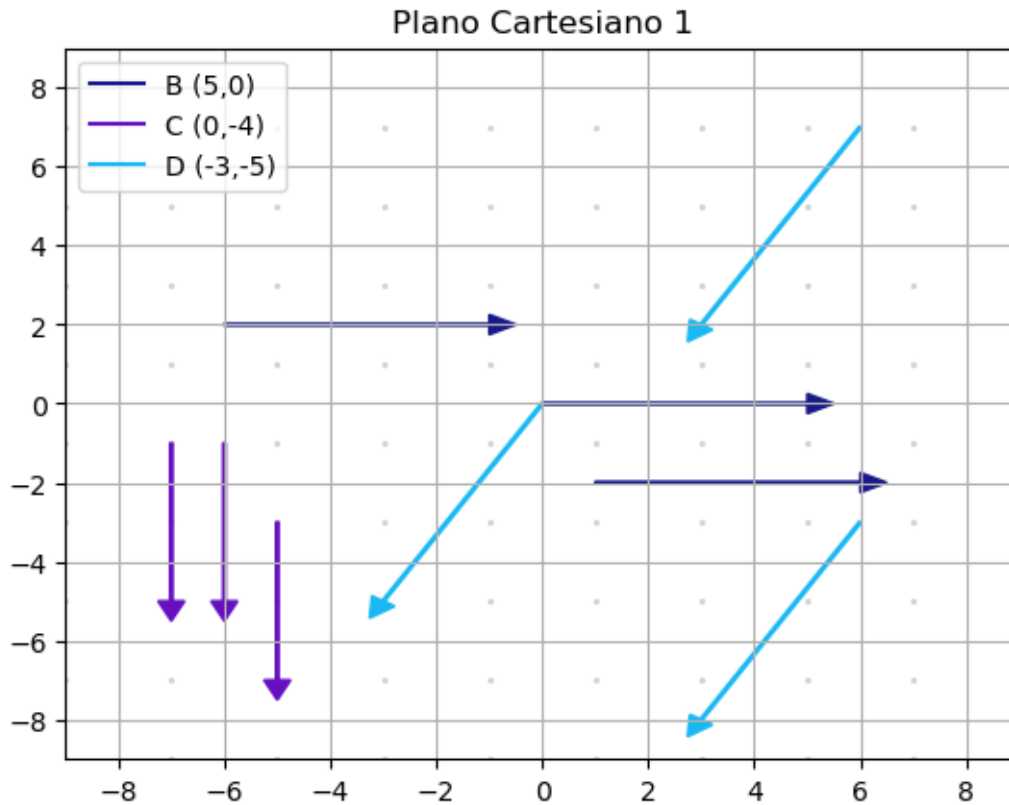
# x, y, incremento, abcisa, longitud, ancho, color
plt.arrow(6,7,-3,-5,head_length=0.5, head_width=0.5, color=(26/255,184/255,243/
↪255), width=0.05)
plt.arrow(0,0,5,0,head_length=0.5, head_width=0.5, color=(23/255,25/255,136/
↪255), width=0.05)
plt.arrow(-6,2,5,0,head_length=0.5, head_width=0.5, color=(23/255,25/255,136/
↪255), width=0.05)
plt.arrow(0,0,-3,-5,head_length=0.5, head_width=0.5, color=(26/255,184/255,243/
↪255), width=0.05)
plt.arrow(-7,-1,0,-4,head_length=0.5, head_width=0.5, color=(101/255,15/255,193/
↪255), width=0.05)
plt.arrow(-6,-1,0,-4,head_length=0.5, head_width=0.5, color=(101/255,15/255,193/
↪255), width=0.05)
plt.arrow(-5,-3,0,-4,head_length=0.5, head_width=0.5, color=(101/255,15/255,193/
↪255), width=0.05)
plt.arrow(6,-3,-3,-5,head_length=0.5, head_width=0.5, color=(26/255,184/255,243/
↪255), width=0.05)
plt.arrow(1,-2,5,0,head_length=0.5, head_width=0.5, color=(23/255,25/255,136/
↪255), width=0.05)

# Crear una línea 'invisible' para mostrar la simbología o leyenda
plt.plot([], [], color=(23/255, 25/255, 136/255), label='B (5,0)')
plt.plot([], [], color=(101/255,15/255,193/255), label='C (0,-4)')
plt.plot([], [], color=(26/255, 184/255, 243/255), label='D (-3,-5)')
plt.legend(loc='upper left')

```

[47]: <matplotlib.legend.Legend at 0x7da7da9cf560>





### 1.0.6 Ejercicio 2:

#### Plano Cartesiano Completo

```
[49]: # Coordenadas que abarca el mapa cartesiano
x1=-8
x2=8
y1=-8
y2=8

# Crear el eje
plt.axis([x1,x2,y1,y2])

# Habilitar el grid
plt.grid(True)
plt.axis('on')
plt.title('Plano Cartesiano 2')

# Generar puntos
dx=2
dy=2
```

```

for x in np.arange(x1,x2,dx):
    for y in np.arange(y1,y2,dy):
        plt.scatter(x,y,s=1.5, color='lightgray')

        # x, y, incremento, abcisa, longitud, ancho, color
plt.arrow(2,4,-5,-1,head_length=0.5, head_width=0.5, color=(51/255,131/255,163/
↪255), width=0.05)
plt.arrow(1,-4,-4,7,head_length=0.5, head_width=0.5, color=(159/255,69/255,195/
↪255), width=0.05)
plt.arrow(6,3,-5,-1,head_length=0.5, head_width=0.5, color=(51/255,131/255,163/
↪255), width=0.05)
plt.arrow(5,-5,-4,7,head_length=0.5, head_width=0.5, color=(159/255,69/255,195/
↪255), width=0.05)

# Crear una línea 'invisible' para mostrar la simbología o leyenda
plt.plot([], [], color=(51/255,131/255,163/255), label='MN (-5,-1)')
plt.plot([], [], color=(159/255,69/255,195/255), label='PN (-4,7)')
plt.legend(loc='upper left')

```

[49]: <matplotlib.legend.Legend at 0x7da7d84df7d0>

