# Vectores

September 11, 2024

**Universidad Autónoma del Estado de México**

**Centro Universitario UAEM Zumpango**

**Ingeniería en Computación**

**Graficación Computacional**

**Alumno: Diego Argel Navarrete Godines**

**Profesora: Hazem Álvarez Rodríguez**

**Fecha: 11 de septiembre del 2024**

**Descripcion: Vectores en python**

## 1 ¿Qué es un Vector?

**1.1 En fisica y matematicas, un vector es un segmento de una linea recta, dotado de un sentido, es decir, orientado dentro de un plano euclidiano bidimensional o tridimencional. o bien es un elemento en un espacio vectorial.**

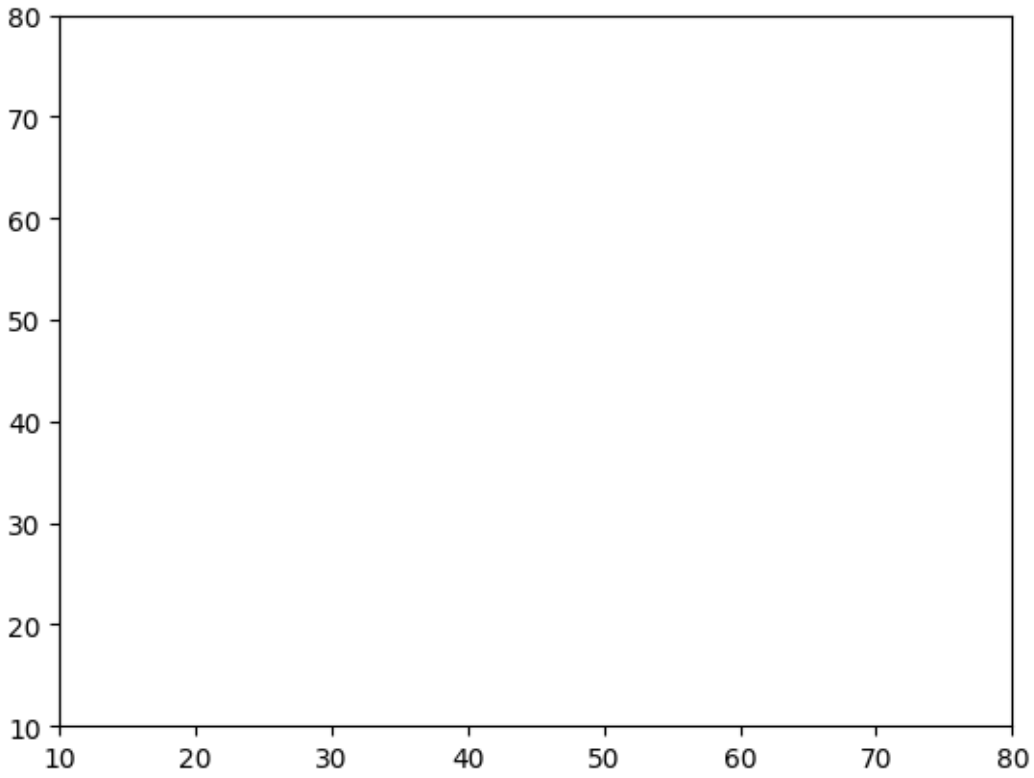
**1.2 Permiten representar magnitudes fisicas dotadas no solo de intensidad, sino de direccion, como es el caso de la fuerza, la velocidad o el desplazamiento. Ese rasgo de contar con dirección es el que distingue a las magnitudes vecotriales de las escalares.**

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

x1=-10
x2=80
y1=80
y2=-10

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

## 2 correcion de los ejes de la grafica

[7]:
```
import numpy as np
import matplotlib.pyplot as plt

x1=10
x2=80
y1=10
y2=80

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

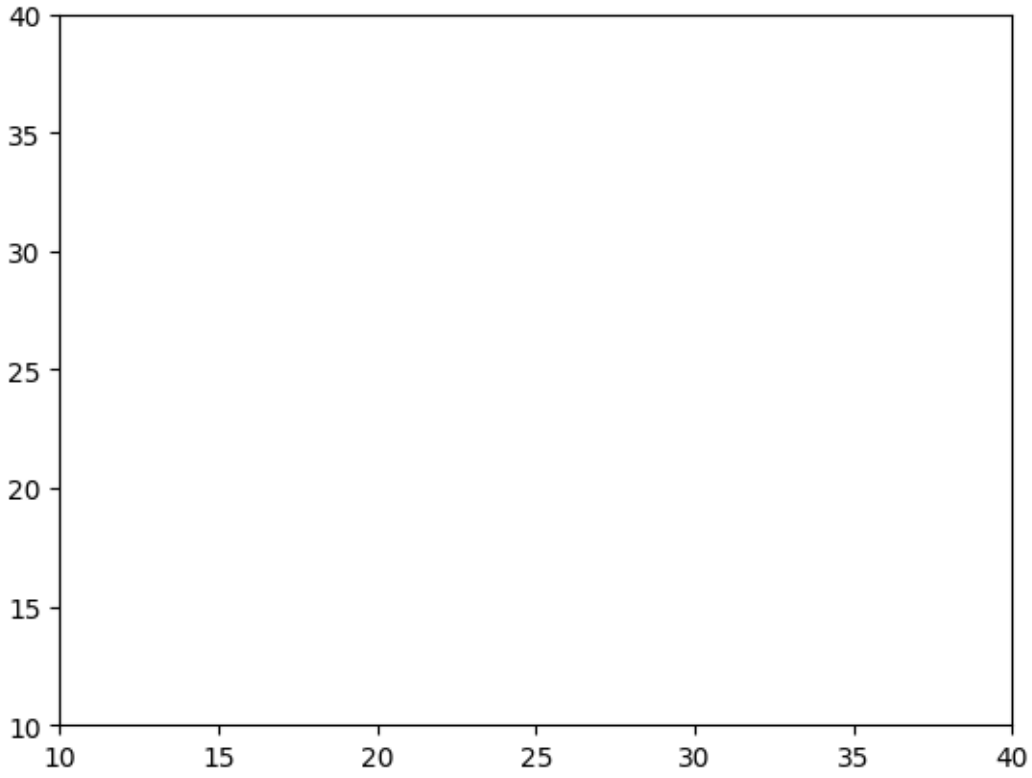
[7]:
```
(10.0, 80.0, 10.0, 80.0)
```




[8]:
```
import numpy as np
import matplotlib.pyplot as plt

x1=10
x2=40
y1=10
y2=40
#Definir ejes
```

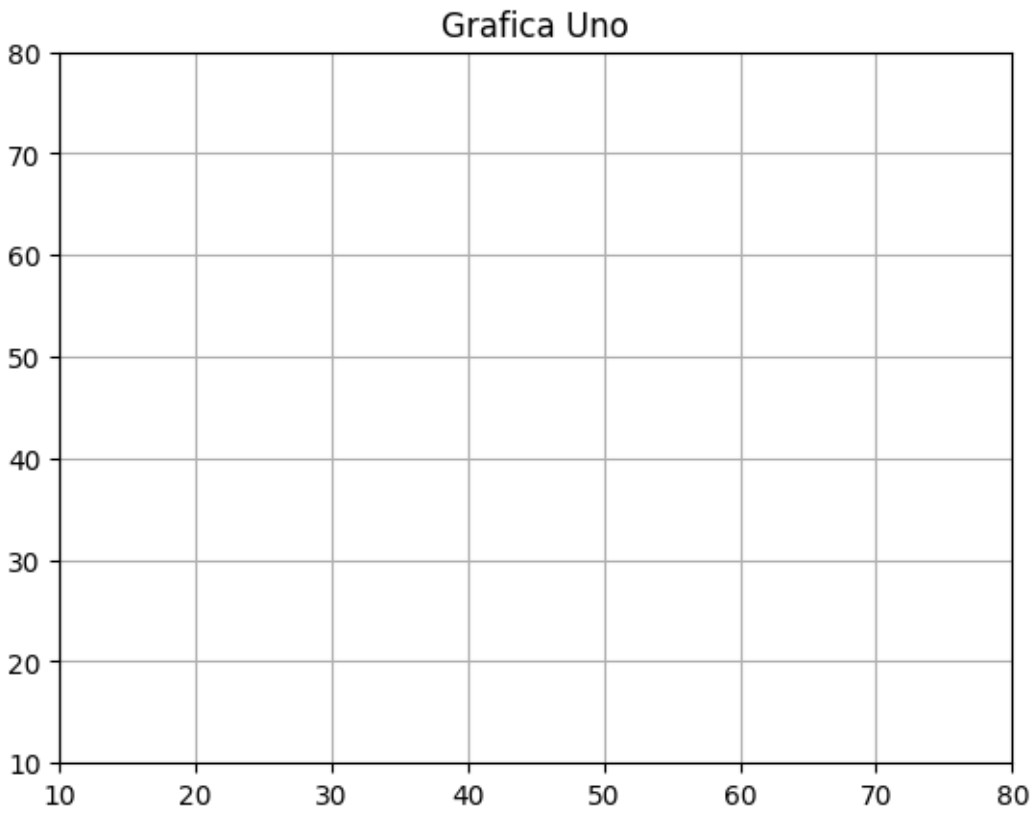
```
plt.axis([x1,x2,y1,y2])
```

[8]: (10.0, 40.0, 10.0, 40.0)



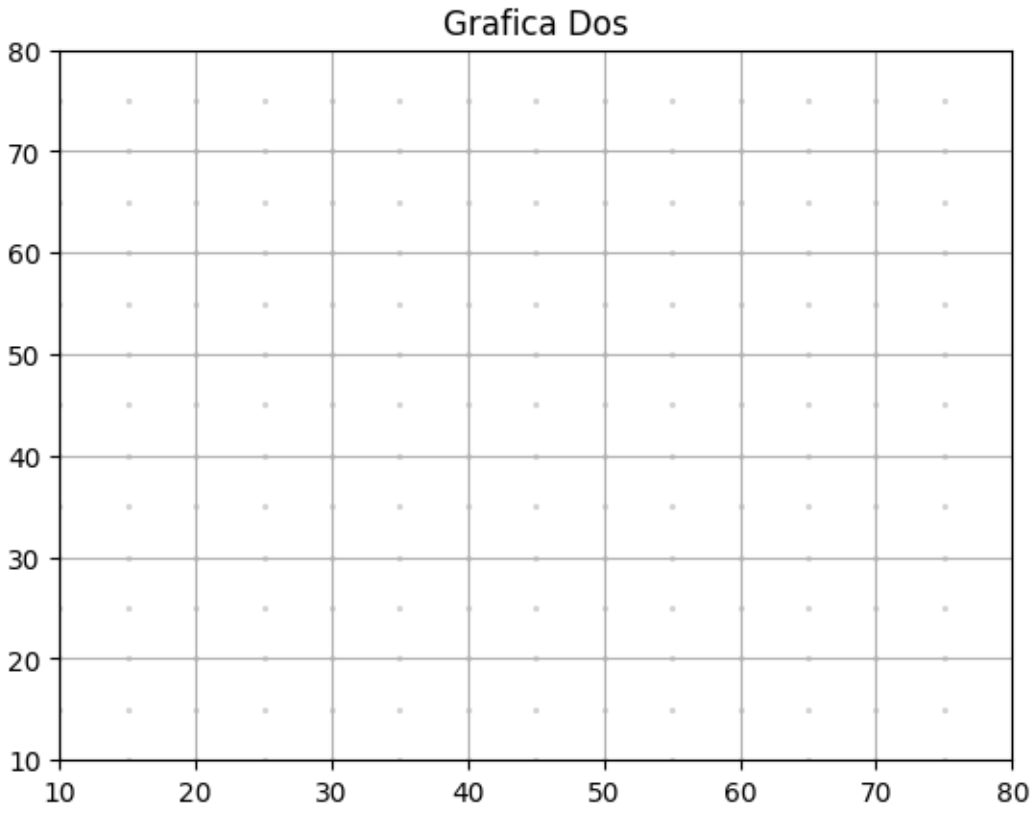

[12]:
```
import numpy as np
import matplotlib.pyplot as plt

x1=10
x2=80
y1=10
y2=80
# Definir ejes
plt.axis([x1,x2,y1,y2])
# Agregamos los grid
plt.axis('on')
# Agregamos los grid
plt.grid(True)
# Agregamos un titulo en la grafica
plt.title('Grafica Uno')
```

[12]: Text(0.5, 1.0, 'Grafica Uno')




[20]:
```
import numpy as np
import matplotlib.pyplot as plt

x1 = 10
x2 = 80
y1 = 10
y2 = 80
# Definir ejes
plt.axis([x1,x2,y1,y2])
# Agregamos los grid
plt.axis('on')
# Agregamos los grid
plt.grid(True)
# Agregamos un titulo en la grafica
plt.title('Grafica Dos')

dx = 5
dy = 5
# Graficar puntos a mitad de las lineas (s = 1.5)
for x in np.arange(x1, x2, dx):
    for y in np.arange(y1, y2, dy):
```

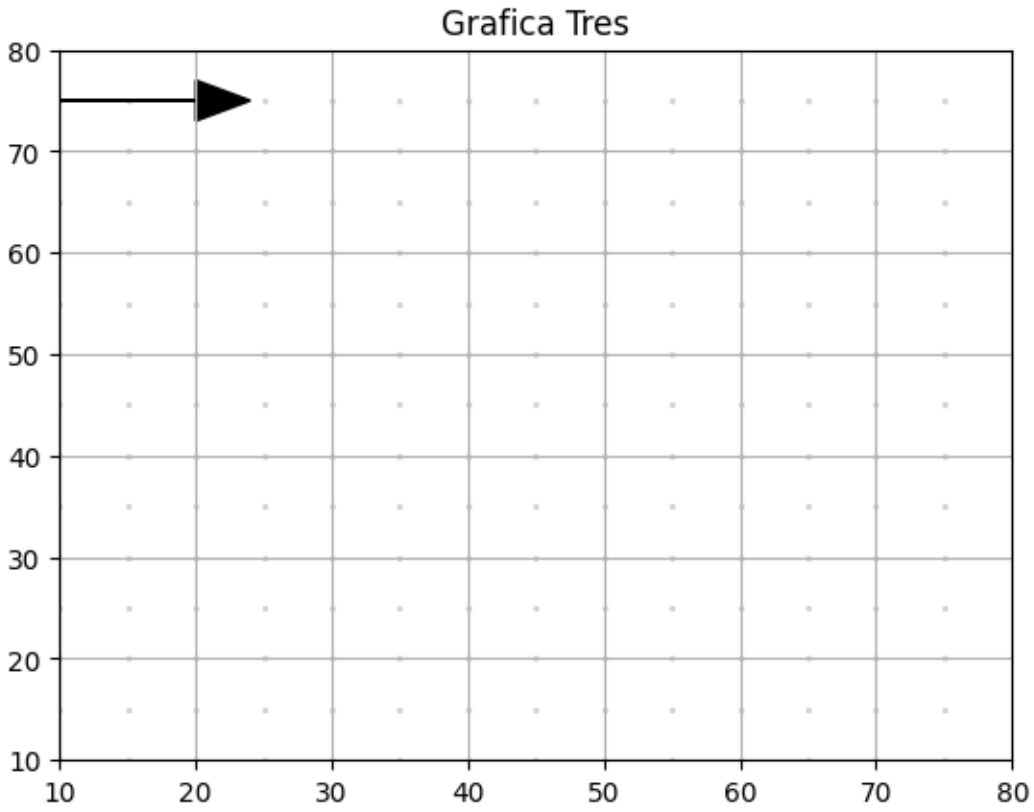
```
        #plt.scatter(x_array, y_array, s_tamaño, color, etc)
        plt.scatter(x, y, s = 1.5, color = 'lightgray')
```




[27]:
```
import numpy as np
import matplotlib.pyplot as plt

x1 = 10
x2 = 80
y1 = 10
y2 = 80
# Definir ejes
plt.axis([x1,x2,y1,y2])
# Agregamos los grid
plt.axis('on')
# Agregamos los grid
plt.grid(True)
# Agregamos un titulo en la grafica
plt.title('Grafica Tres')

dx = 5
```

```
dy = 5
# Graficar puntos a mitad de las lineas (s = 1.5)
for x in np.arange(x1, x2, dx):
    for y in np.arange(y1, y2, dy):
        #plt.scatter(x_array, y_array, s_tamaño, color, etc)
        plt.scatter(x, y, s = 1.5, color = 'lightgray')

# Graficando el vector
        # x,  y, incremento, abscisa, longitud, ancho, color
plt.arrow(0, 75, 20, 0, head_length = 4, head_width = 4, color =  "k") # k es␣
 ↪negro y b es azul
```

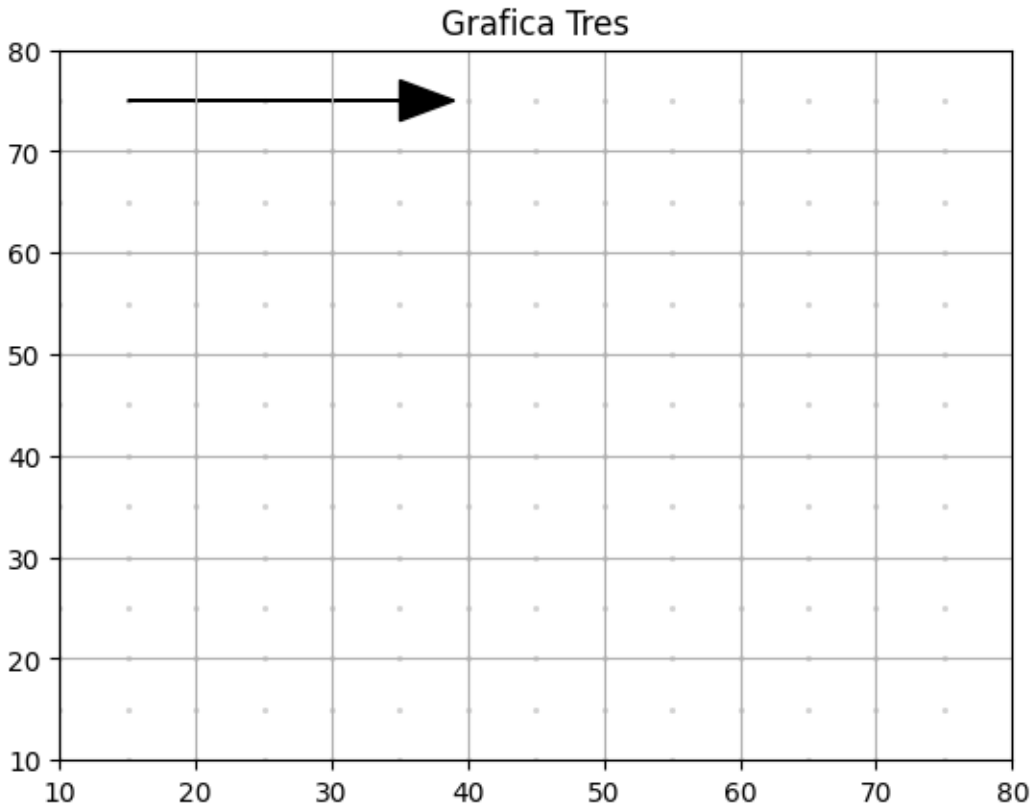
[27]: <matplotlib.patches.FancyArrow at 0x1eee75f9940>




[33]:
```
import numpy as np
import matplotlib.pyplot as plt

x1 = 10
x2 = 80
y1 = 10
y2 = 80
```

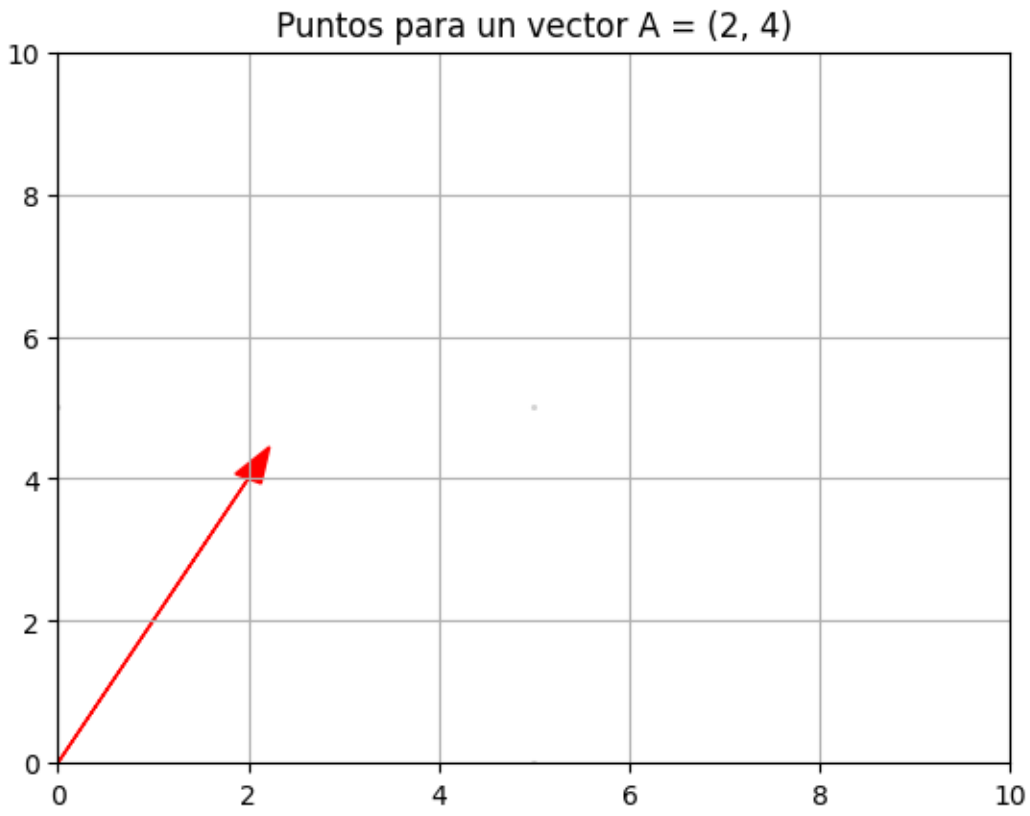
```
# Definir ejes
plt.axis([x1,x2,y1,y2])
# Agregamos los grid
plt.axis('on')
# Agregamos los grid
plt.grid(True)
# Agregamos un titulo en la grafica
plt.title('Grafica Tres')

dx = 5
dy = 5
# Graficar puntos a mitad de las lineas (s = 1.5)
for x in np.arange(x1, x2, dx):
    for y in np.arange(y1, y2, dy):
        #plt.scatter(x_array, y_array, s_tamaño, color, etc)
        plt.scatter(x, y, s = 1.5, color = 'lightgray')

# Graficando el vector
        # x,  y, incremento, abscisa, longitud, ancho, color
plt.arrow(15, 75, 20, 0, head_length = 4, head_width = 4, color =  "k") # k es negro y b es azul
```

[33]: <matplotlib.patches.FancyArrow at 0x1eee9d30050>



[29]:
```
import numpy as np
import matplotlib.pyplot as plt

x1 = 0
x2 = 10
y1 = 0
y2 = 10
# Definir ejes
plt.axis([x1,x2,y1,y2])
# Agregamos los grid
plt.axis('on')
# Agregamos los grid
plt.grid(True)
# Agregamos un titulo en la grafica
plt.title('Puntos para un vector A = (2, 4)')

dx = 5
dy = 5
# Graficar puntos a mitad de las lineas (s = 1.5)
for x in np.arange(x1, x2, dx):
    for y in np.arange(y1, y2, dy):
        #plt.scatter(x_array, y_array, s_tamaño, color, etc)
        plt.scatter(x, y, s = 1.5, color = 'lightgray')

# Graficando el vector
        # x,  y, incremento, abscisa, longitud, ancho, color
plt.arrow(0, 0, 2, 4, head_length = 0.5, head_width = 0.3, color = "r") #␣
 ↪Vector azul
```

[29]:
```
<matplotlib.patches.FancyArrow at 0x2213a0730b0>
```

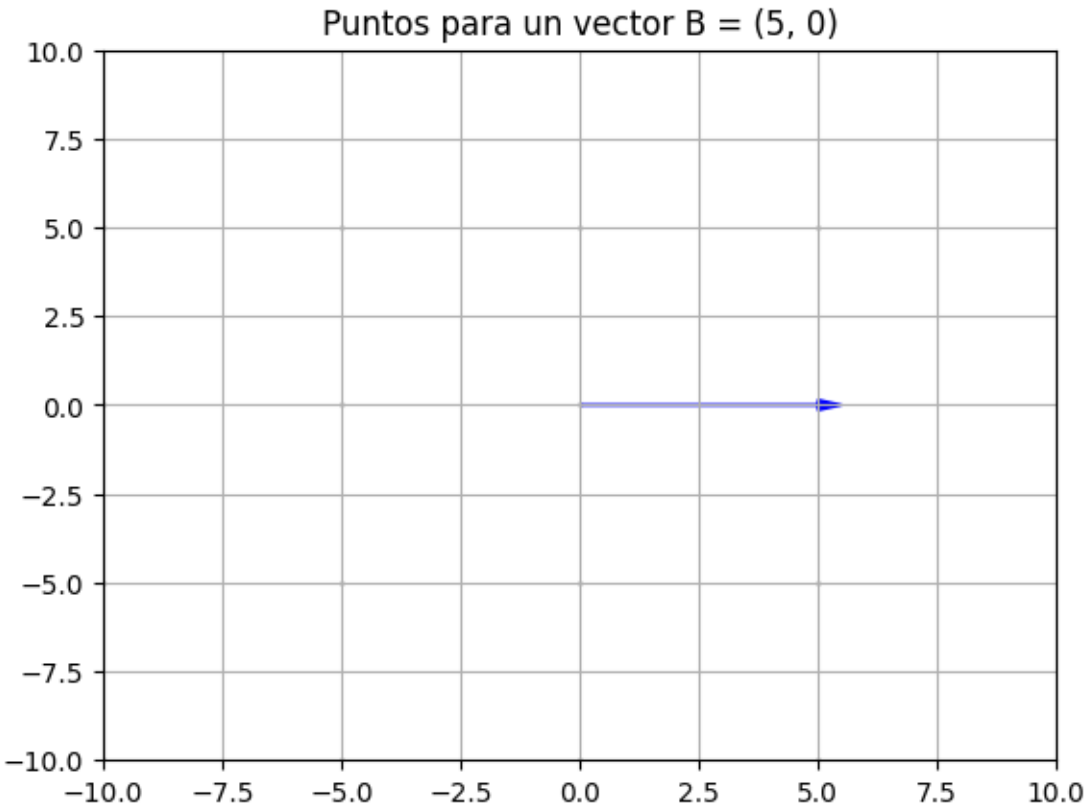



**Este código utiliza numpy y matplotlib para crear una gráfica 2D con una cuadrícula de puntos y un vector. Se definen los ejes de 0 a 10 para x e y, se activa una cuadrícula visible, y se dibujan puntos grises en intervalos de 5 unidades en ambos ejes. Además, se grafica un vector rojo que va desde el origen (0, 0) hasta el punto (2, 4), con una flecha que indica su dirección. El gráfico también incluye un título que describe el vector A = (2, 4).**

[1]:
```
import numpy as np
import matplotlib.pyplot as plt

x1 = -10
x2 = 10
y1 = -10
y2 = 10
# Definir ejes
plt.axis([x1,x2,y1,y2])
# Agregamos los grid
plt.axis('on')
# Agregamos los grid
plt.grid(True)
# Agregamos un titulo en la grafica
```

```
plt.title('Puntos para un vector B = (5, 0)')

dx = 5
dy = 5
# Graficar puntos a mitad de las lineas (s = 1.5)
for x in np.arange(x1, x2, dx):
    for y in np.arange(y1, y2, dy):
        #plt.scatter(x_array, y_array, s_tamaño, color, etc)
        plt.scatter(x, y, s = 1.5, color = 'lightgray')

# Graficando el vector
        # x,  y, incremento, abscisa, longitud, ancho, color
plt.arrow(0, 0, 5, 0, head_length = 0.5, head_width = 0.3, color = "b") #␣
 ↪Vector azul
```

[1]: <matplotlib.patches.FancyArrow at 0x22f58b95be0>

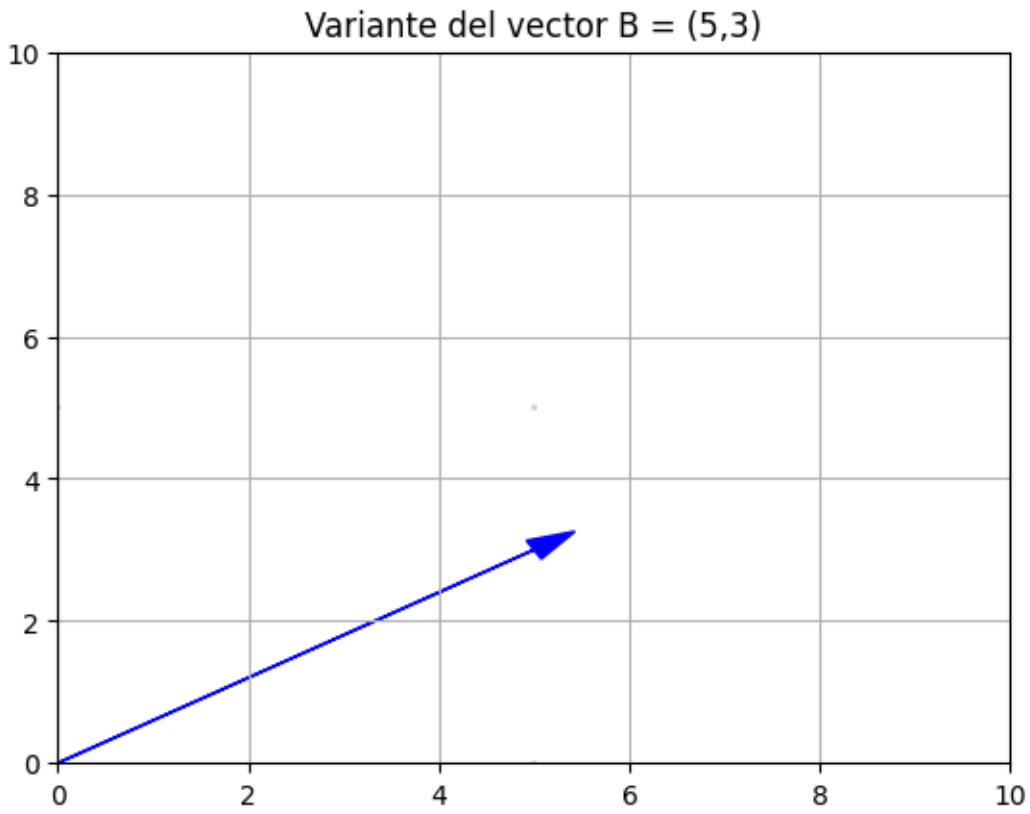



**Este código genera una gráfica en 2D usando numpy y matplotlib, mostrando una cuadrícula de puntos y un vector. Se definen los límites de los ejes de -10 a 10 en ambas direcciones (x e y), y se activa una cuadrícula visible. Luego, se añaden puntos grises claros cada 5 unidades en ambos ejes. Además, se grafica un vector azul que**

**parte desde el origen (0, 0) y se extiende hacia el punto (5, 0), lo que representa el vector B = (5, 0). Finalmente, la gráfica tiene un título que describe este vector.**

[19]:
```
import numpy as np
import matplotlib.pyplot as plt

x1 = 0
x2 = 10
y1 = 0
y2 = 10
# Definir ejes
plt.axis([x1,x2,y1,y2])
# Agregamos los grid
plt.axis('on')
# Agregamos los grid
plt.grid(True)
# Agregamos un titulo en la grafica
plt.title('Variante del vector B = (5,3)')

dx = 5
dy = 5
# Graficar puntos a mitad de las lineas (s = 1.5)
for x in np.arange(x1, x2, dx):
    for y in np.arange(y1, y2, dy):
        #plt.scatter(x_array, y_array, s_tamaño, color, etc)
        plt.scatter(x, y, s = 1.5, color = 'lightgray')

# Graficando el vector
        # x,  y, incremento, abscisa, longitud, ancho, color
plt.arrow(0, 0, 5, 3, head_length = 0.5, head_width = 0.3, color = "b") #␣
 ↪Vector azul
```

[19]:
```
<matplotlib.patches.FancyArrow at 0x221385631d0>
```

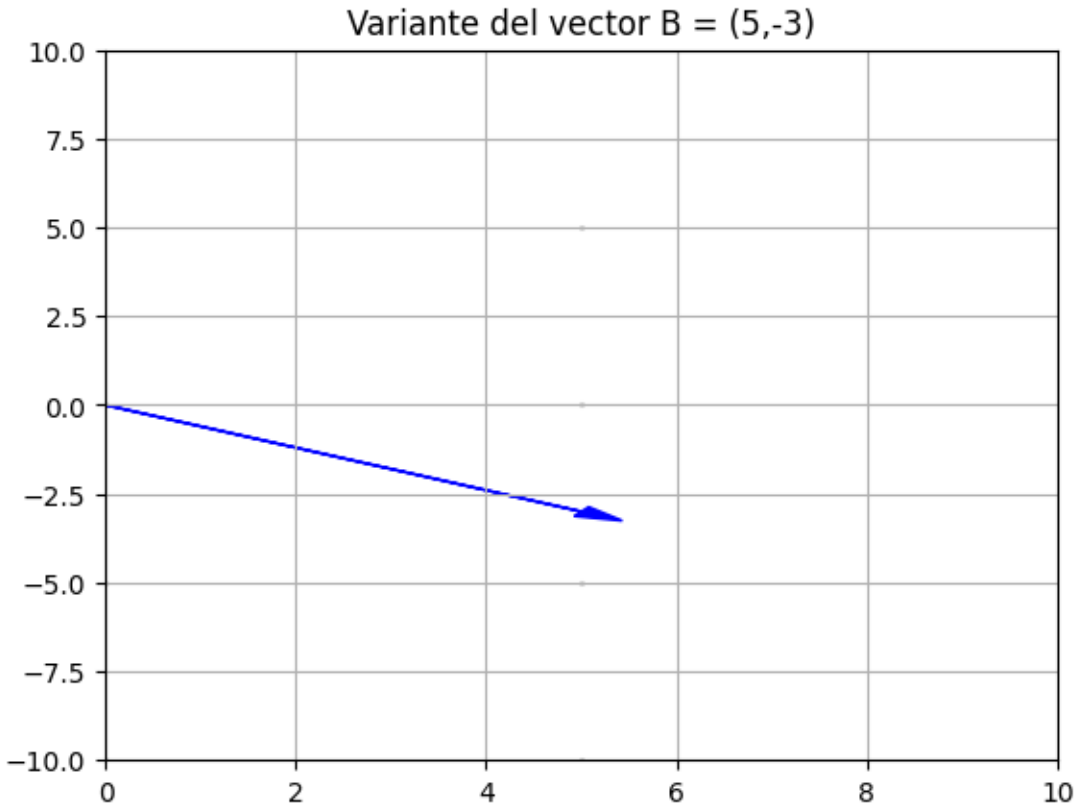



**Este código genera una gráfica en 2D usando numpy y matplotlib, mostrando una cuadrícula de puntos y un vector. Se definen los límites de los ejes de 0 a 10 en x e y, y se activa una cuadrícula visible. Luego, se añaden puntos grises cada 5 unidades en ambos ejes. El vector representado es una variante de B = (5, 3), que parte desde el origen (0, 0) y se extiende hasta el punto (5, 3), dibujado como una flecha azul. El título de la gráfica describe esta variante del vector.**

[20]:
```
import numpy as np
import matplotlib.pyplot as plt

x1 = 0
x2 = 10
y1 = -10
y2 = 10
# Definir ejes
plt.axis([x1,x2,y1,y2])
# Agregamos los grid
plt.axis('on')
# Agregamos los grid
plt.grid(True)
# Agregamos un titulo en la grafica
```

```
plt.title('Variante del vector B = (5,-3)')

dx = 5
dy = 5
# Graficar puntos a mitad de las lineas (s = 1.5)
for x in np.arange(x1, x2, dx):
    for y in np.arange(y1, y2, dy):
        #plt.scatter(x_array, y_array, s_tamaño, color, etc)
        plt.scatter(x, y, s = 1.5, color = 'lightgray')

# Graficando el vector
        # x,  y, incremento, abscisa, longitud, ancho, color
plt.arrow(0, 0, 5, -3, head_length = 0.5, head_width = 0.3, color = "b") #␣
 ↪Vector azul
```

[20]: <matplotlib.patches.FancyArrow at 0x2213880cf50>

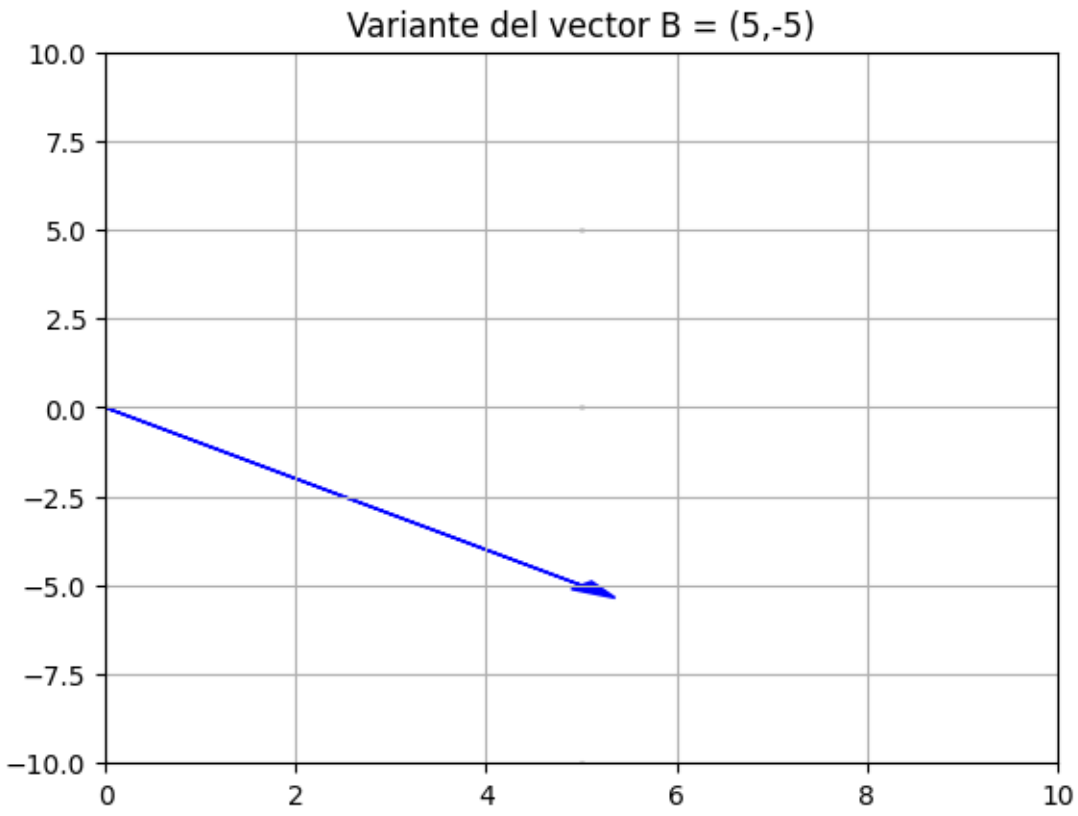



**Este código crea una gráfica en 2D usando numpy y matplotlib, mostrando una cuadrícula de puntos y un vector. Los límites de los ejes se establecen de 0 a 10 en el eje x y de -10 a 10 en el eje y, con una cuadrícula visible. Se añaden puntos grises claros cada 5 unidades tanto en x como en y. El vector representado es B = (5,**

**-3), partiendo del origen (0, 0) hacia el punto (5, -3) y dibujado como una flecha azul. La gráfica incluye un título que describe esta variante del vector.**

[21]:
```
import numpy as np
import matplotlib.pyplot as plt

x1 = 0
x2 = 10
y1 = -10
y2 = 10
# Definir ejes
plt.axis([x1,x2,y1,y2])
# Agregamos los grid
plt.axis('on')
# Agregamos los grid
plt.grid(True)
# Agregamos un titulo en la grafica
plt.title('Variante del vector B = (5,-5)')

dx = 5
dy = 5
# Graficar puntos a mitad de las lineas (s = 1.5)
for x in np.arange(x1, x2, dx):
    for y in np.arange(y1, y2, dy):
        #plt.scatter(x_array, y_array, s_tamaño, color, etc)
        plt.scatter(x, y, s = 1.5, color = 'lightgray')

# Graficando el vector
        # x,  y, incremento, abscisa, longitud, ancho, color
plt.arrow(0, 0, 5, -5, head_length = 0.5, head_width = 0.3, color = "b") #␣
 ↪Vector azul
```

[21]:
```
<matplotlib.patches.FancyArrow at 0x22139aebb90>
```

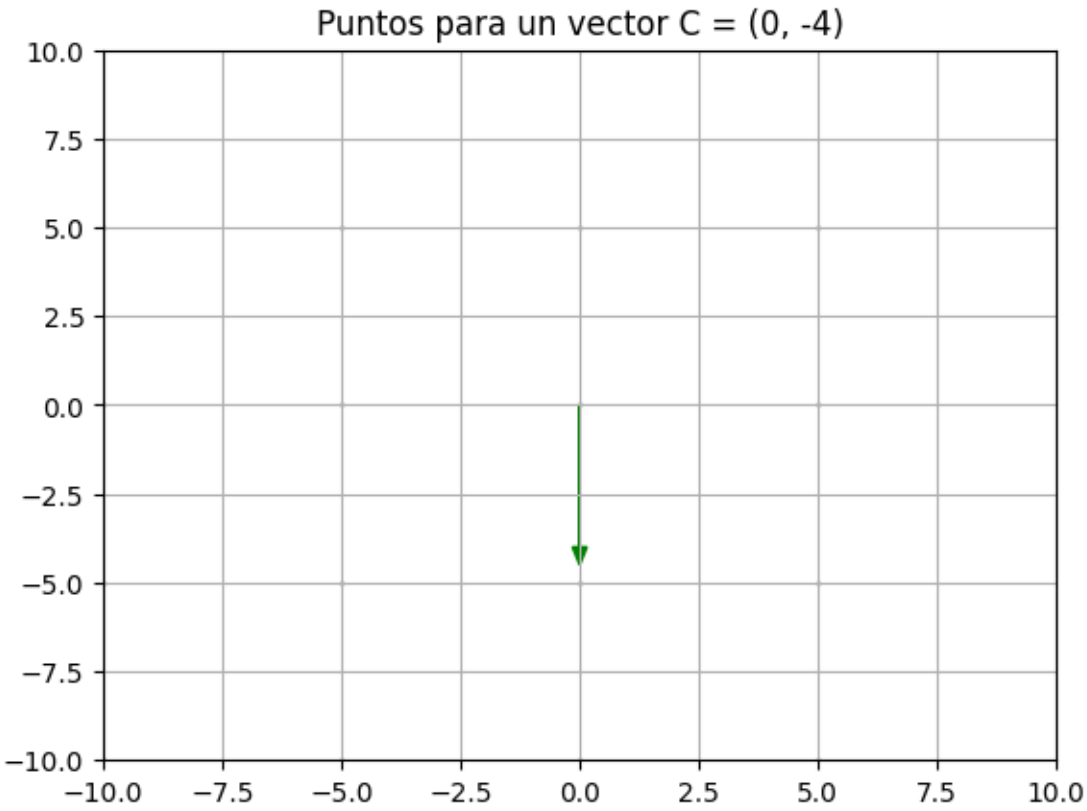



**Este código genera una gráfica 2D usando numpy y matplotlib, mostrando una cuadrícula de puntos y un vector. Los ejes están limitados entre 0 y 10 en el eje x, y entre -10 y 10 en el eje y, con la cuadrícula activada. Se añaden puntos grises cada 5 unidades en ambos ejes. El vector representado es B = (5, -5), que se grafica desde el origen (0, 0) hacia el punto (5, -5) con una flecha azul. El título de la gráfica describe esta variante del vector.**

[137]:
```
import numpy as np
import matplotlib.pyplot as plt

x1 = -10
x2 = 10
y1 = -10
y2 = 10
# Definir ejes
plt.axis([x1,x2,y1,y2])
# Agregamos los grid
plt.axis('on')
# Agregamos los grid
plt.grid(True)
# Agregamos un titulo en la grafica
```

```
plt.title('Puntos para un vector C = (0, -4)')

dx = 5
dy = 5
# Graficar puntos a mitad de las lineas (s = 1.5)
for x in np.arange(x1, x2, dx):
    for y in np.arange(y1, y2, dy):
        #plt.scatter(x_array, y_array, s_tamaño, color, etc)
        plt.scatter(x, y, s = 1.5, color = 'lightgray')

# Graficando el vector
        # x,  y, incremento, abscisa, longitud, ancho, color
plt.arrow(0, 0, 0, -4, head_length = 0.5, head_width = 0.3, color = "g")  #␣
 ↪Vector verde
```

[137]: <matplotlib.patches.FancyArrow at 0x2214ad030b0>

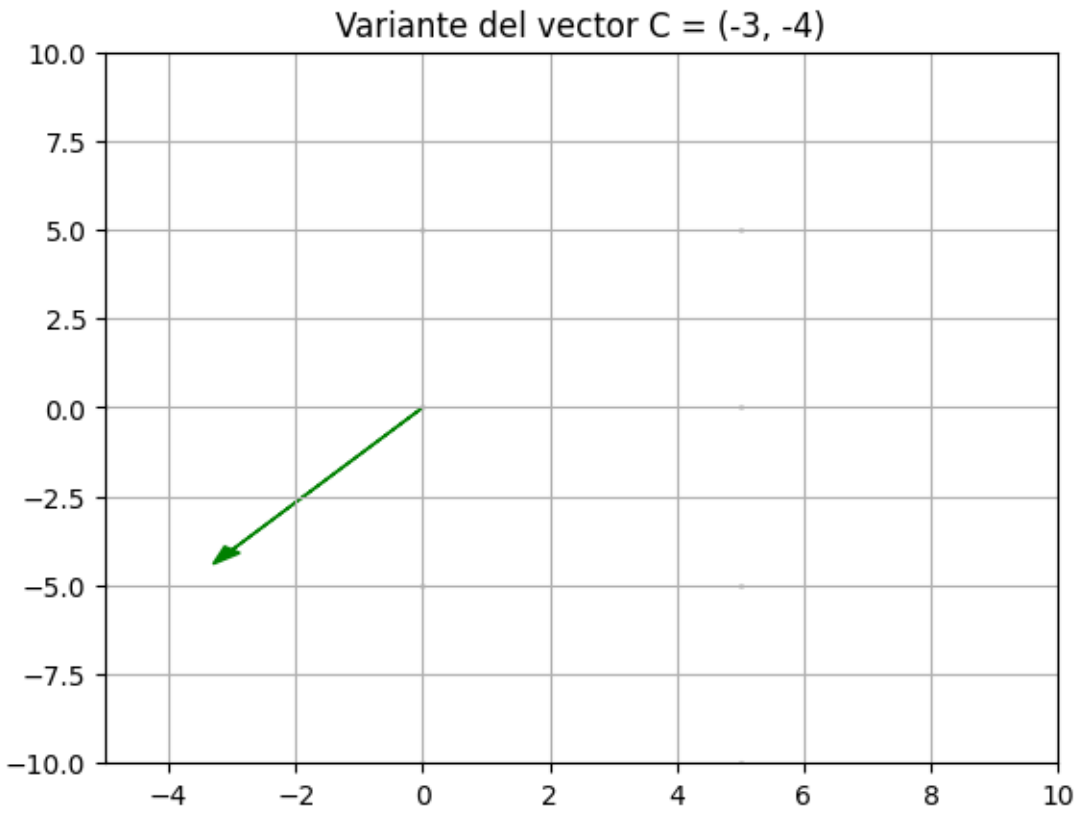



**Este código genera una gráfica en 2D utilizando numpy y matplotlib, mostrando una cuadrícula de puntos y un vector. Los límites de los ejes van de -10 a 10 tanto en x como en y, con la cuadrícula activada. Se dibujan puntos grises cada 5 unidades en ambos ejes. El vector representado es C = (0, -4), graficado como una flecha verde**

**que parte desde el origen (0, 0) y se extiende hacia abajo en la dirección del eje y negativo. El título de la gráfica indica que el vector C es (0, -4).**

[27]:
```
import numpy as np
import matplotlib.pyplot as plt

x1 = -5
x2 = 10
y1 = -10
y2 = 10
# Definir ejes
plt.axis([x1,x2,y1,y2])
# Agregamos los grid
plt.axis('on')
# Agregamos los grid
plt.grid(True)
# Agregamos un titulo en la grafica
plt.title('Variante del vector C = (-3, -4)')

dx = 5
dy = 5
# Graficar puntos a mitad de las lineas (s = 1.5)
for x in np.arange(x1, x2, dx):
    for y in np.arange(y1, y2, dy):
        #plt.scatter(x_array, y_array, s_tamaño, color, etc)
        plt.scatter(x, y, s = 1.5, color = 'lightgray')

# Graficando el vector
        # x,  y, incremento, abscisa, longitud, ancho, color
plt.arrow(0, 0, -3, -4, head_length = 0.5, head_width = 0.3, color = "g")  #␣
 ↪Vector verde
```

[27]:
```
<matplotlib.patches.FancyArrow at 0x22139d5b0b0>
```

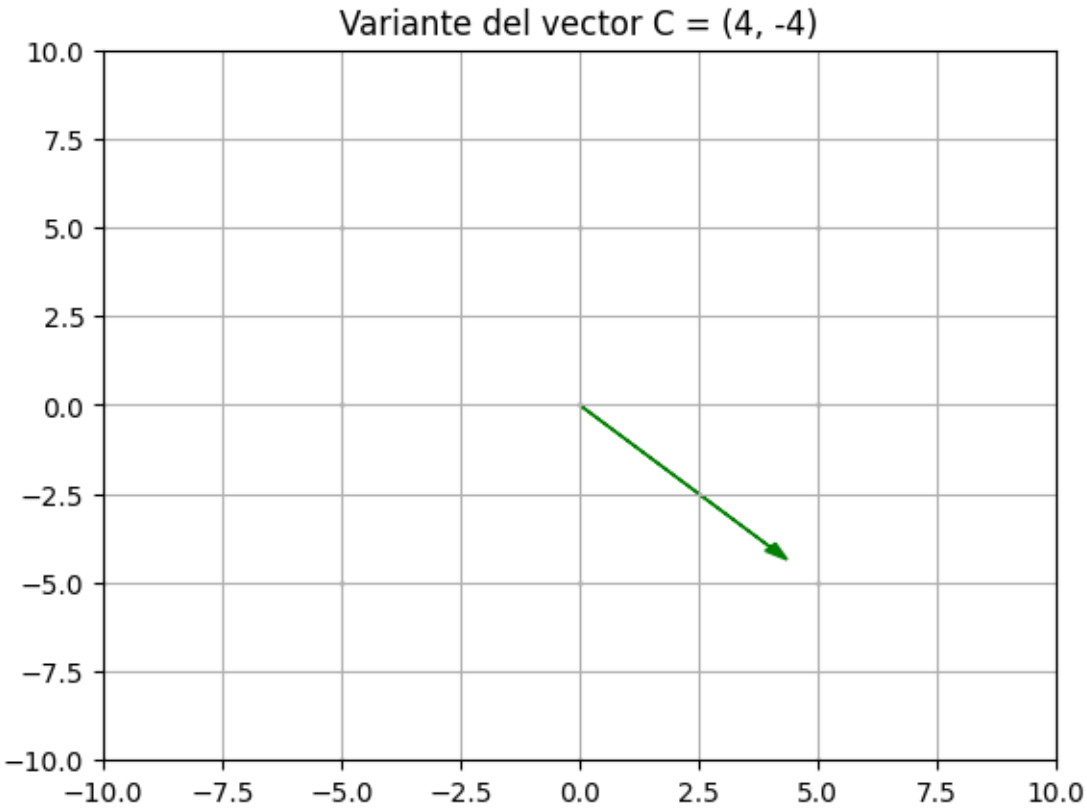



**Este código genera una gráfica 2D utilizando numpy y matplotlib, mostrando una cuadrícula de puntos y un vector. Los límites de los ejes se establecen entre -5 y 10 para x, y entre -10 y 10 para y, con la cuadrícula activada. Se dibujan puntos grises cada 5 unidades en ambos ejes. El vector representado es C = (-3, -4), partiendo del origen (0, 0) hacia el punto (-3, -4), y está graficado como una flecha verde. El título de la gráfica describe esta variante del vector C.**

[138]:
```
import numpy as np
import matplotlib.pyplot as plt

x1 = -10
x2 = 10
y1 = -10
y2 = 10
# Definir ejes
plt.axis([x1,x2,y1,y2])
# Agregamos los grid
plt.axis('on')
# Agregamos los grid
plt.grid(True)
# Agregamos un titulo en la grafica
```

```
plt.title('Variante del vector C = (4, -4)')

dx = 5
dy = 5
# Graficar puntos a mitad de las lineas (s = 1.5)
for x in np.arange(x1, x2, dx):
    for y in np.arange(y1, y2, dy):
        #plt.scatter(x_array, y_array, s_tamaño, color, etc)
        plt.scatter(x, y, s = 1.5, color = 'lightgray')

# Graficando el vector
        # x,  y, incremento, abscisa, longitud, ancho, color
plt.arrow(0, 0, 4, -4, head_length = 0.5, head_width = 0.3, color = "g")  #␣
 ↪Vector verde
```

[138]: <matplotlib.patches.FancyArrow at 0x2214ace6b70>

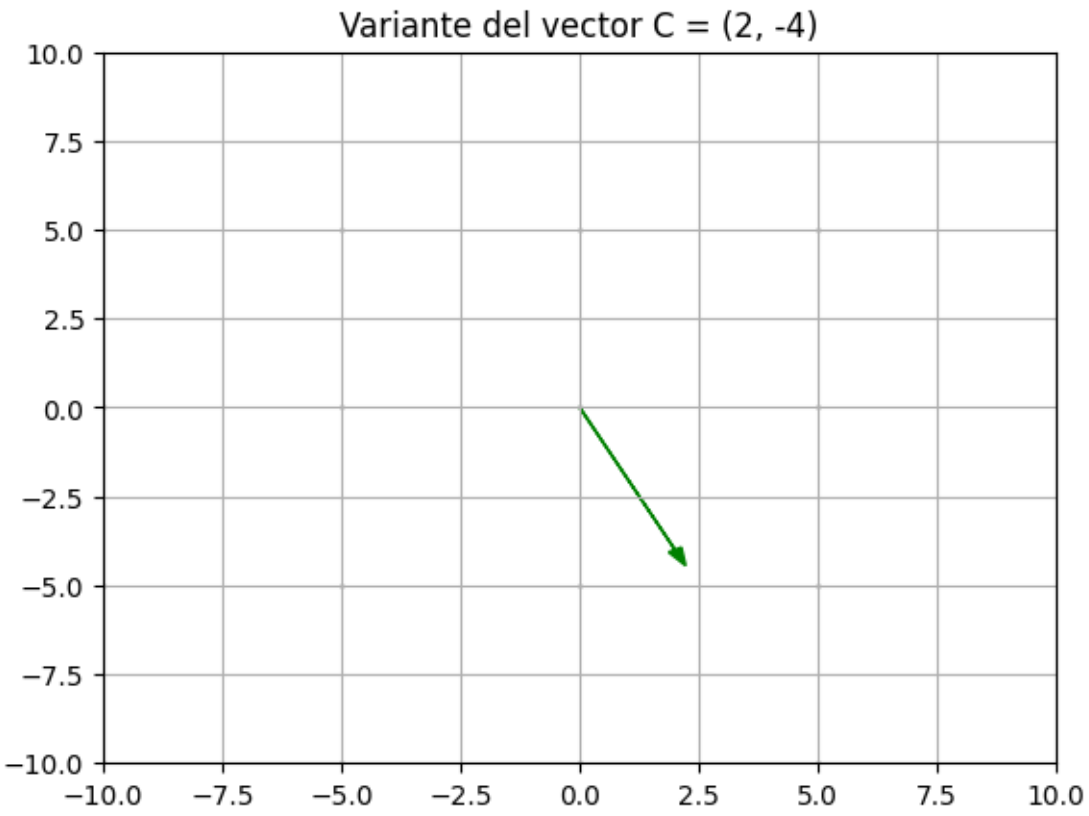



**Este código genera una gráfica 2D usando numpy y matplotlib, mostrando una cuadrícula de puntos y un vector. Los límites de los ejes se establecen de -10 a 10 tanto en x como en y, y se activa la cuadrícula. Se dibujan puntos grises cada 5 unidades en ambos ejes. El vector representado es C = (4, -4), partiendo del origen**

**(0, 0) hacia el punto (4, -4), y se dibuja como una flecha verde. El título de la gráfica indica que esta es una variante del vector C = (4, -4).**

[139]:
```
import numpy as np
import matplotlib.pyplot as plt

x1 = -10
x2 = 10
y1 = -10
y2 = 10
# Definir ejes
plt.axis([x1,x2,y1,y2])
# Agregamos los grid
plt.axis('on')
# Agregamos los grid
plt.grid(True)
# Agregamos un titulo en la grafica
plt.title('Variante del vector C = (2, -4)')

dx = 5
dy = 5
# Graficar puntos a mitad de las lineas (s = 1.5)
for x in np.arange(x1, x2, dx):
    for y in np.arange(y1, y2, dy):
        #plt.scatter(x_array, y_array, s_tamaño, color, etc)
        plt.scatter(x, y, s = 1.5, color = 'lightgray')

# Graficando el vector
        # x,  y, incremento, abscisa, longitud, ancho, color
plt.arrow(0, 0, 2, -4, head_length = 0.5, head_width = 0.3, color = "g")  #␣
 ↪Vector verde
```

[139]:
```
<matplotlib.patches.FancyArrow at 0x2214abaffe0>
```

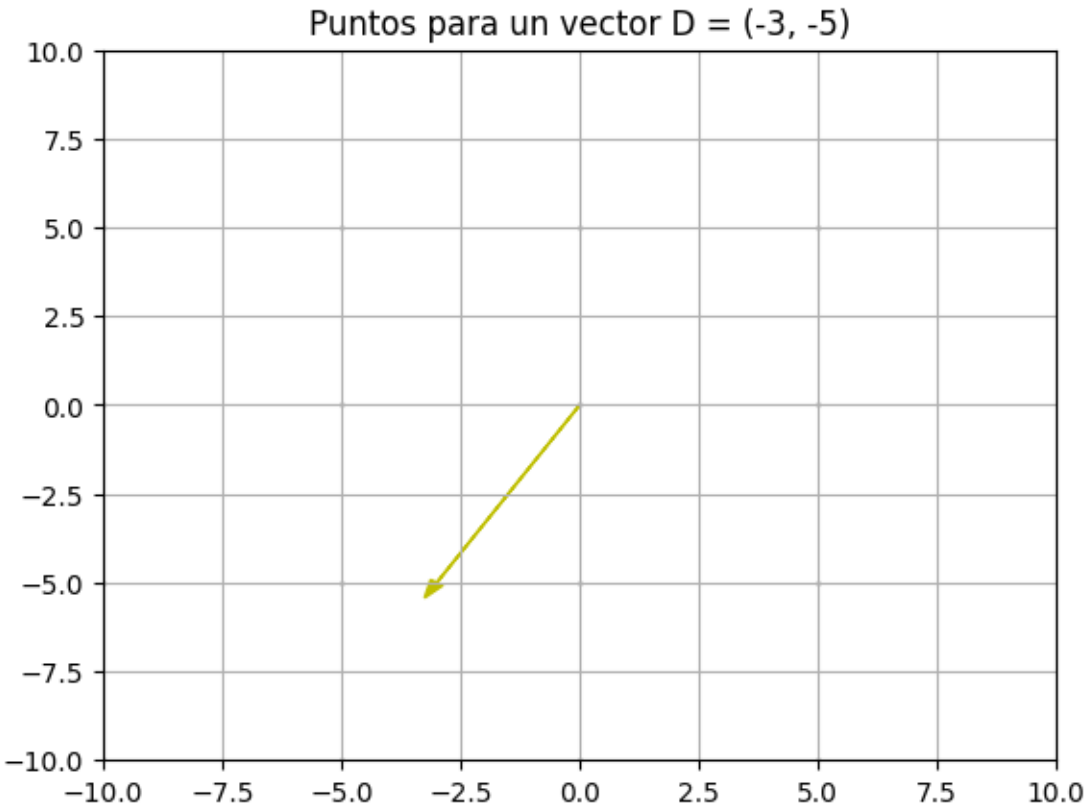



**Este código genera una gráfica 2D usando numpy y matplotlib, mostrando una cuadrícula de puntos y un vector. Los límites de los ejes van de -10 a 10 en ambas direcciones, con una cuadrícula activada. Se dibujan puntos grises cada 5 unidades en los ejes x e y. El vector representado es C = (2, -4), partiendo desde el origen (0, 0) y extendiéndose hacia el punto (2, -4), dibujado como una flecha verde. El título de la gráfica describe esta variante del vector C = (2, -4).**

[37]:
```
import numpy as np
import matplotlib.pyplot as plt

x1 = -10
x2 = 10
y1 = -10
y2 = 10
# Definir ejes
plt.axis([x1,x2,y1,y2])
# Agregamos los grid
plt.axis('on')
# Agregamos los grid
plt.grid(True)
# Agregamos un titulo en la grafica
```

```
plt.title('Puntos para un vector D = (-3, -5)')

dx = 5
dy = 5
# Graficar puntos a mitad de las lineas (s = 1.5)
for x in np.arange(x1, x2, dx):
    for y in np.arange(y1, y2, dy):
        #plt.scatter(x_array, y_array, s_tamaño, color, etc)
        plt.scatter(x, y, s = 1.5, color = 'lightgray')

# Graficando el vector
        # x,  y, incremento, abscisa, longitud, ancho, color
plt.arrow(0, 0, -3, -5, head_length = 0.5, head_width = 0.3, color = "y")  #␣
 ↪Vector rojo
```

[37]: <matplotlib.patches.FancyArrow at 0x22139f1cf50>

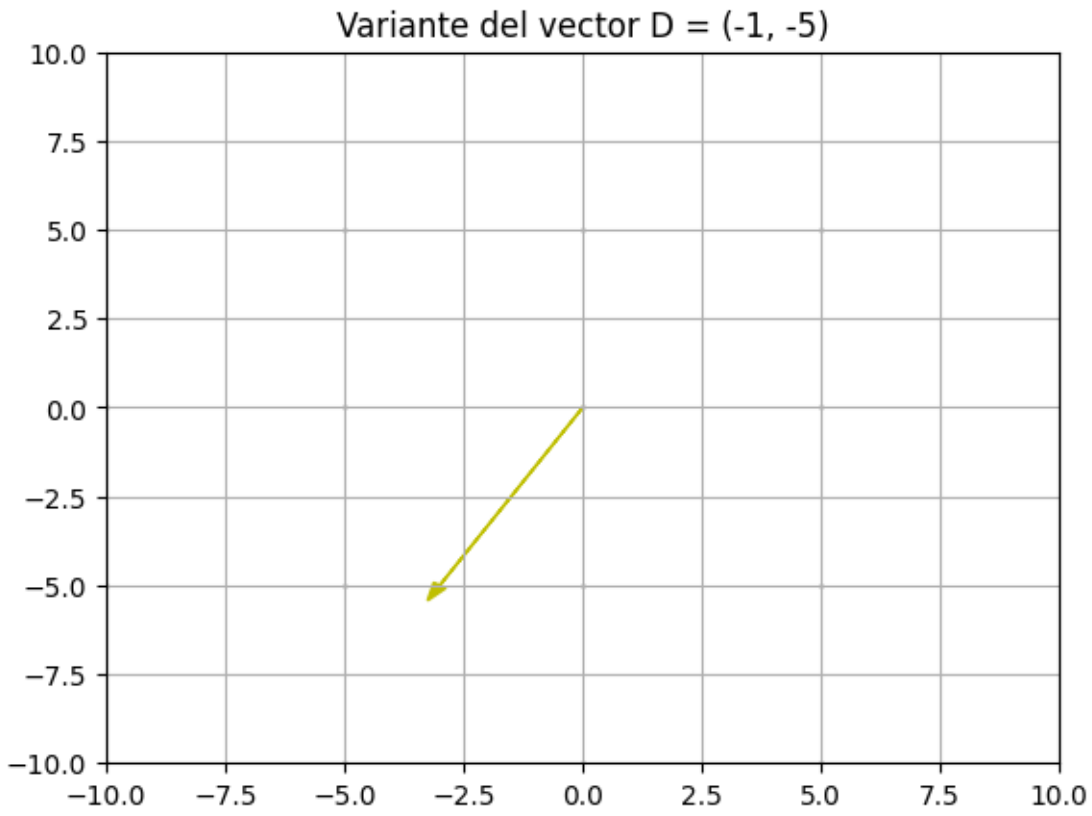



**Este código crea una gráfica 2D utilizando numpy y matplotlib, mostrando una cuadrícula de puntos y un vector. Los ejes están limitados entre -10 y 10 para ambos ejes (x e y), con la cuadrícula activada. Se dibujan puntos grises cada 5 unidades en ambos ejes. El vector representado es D = (-3, -5), que comienza en el origen (0, 0) y**

**se extiende hacia el punto (-3, -5), representado por una flecha amarilla. El título de la gráfica describe el vector D = (-3, -5).**

[38]:
```
import numpy as np
import matplotlib.pyplot as plt

x1 = -10
x2 = 10
y1 = -10
y2 = 10
# Definir ejes
plt.axis([x1,x2,y1,y2])
# Agregamos los grid
plt.axis('on')
# Agregamos los grid
plt.grid(True)
# Agregamos un titulo en la grafica
plt.title('Variante del vector D = (-1, -5)')

dx = 5
dy = 5
# Graficar puntos a mitad de las lineas (s = 1.5)
for x in np.arange(x1, x2, dx):
    for y in np.arange(y1, y2, dy):
        #plt.scatter(x_array, y_array, s_tamaño, color, etc)
        plt.scatter(x, y, s = 1.5, color = 'lightgray')

# Graficando el vector
        # x,  y, incremento, abscisa, longitud, ancho, color
plt.arrow(0, 0, -3, -5, head_length = 0.5, head_width = 0.3, color = "y")  #␣
 ↪Vector rojo
```

[38]:
```
<matplotlib.patches.FancyArrow at 0x2213b534500>
```

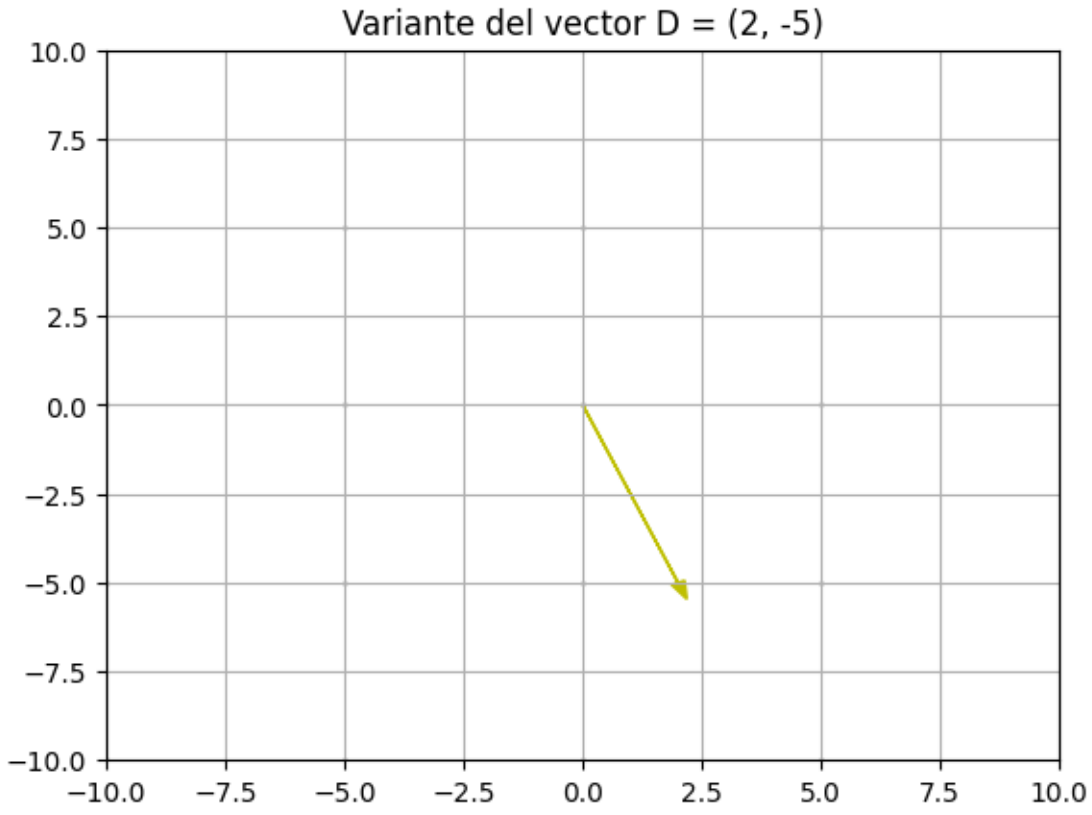



**Este código genera una gráfica 2D utilizando numpy y matplotlib, que muestra una cuadrícula de puntos y un vector. Los ejes están configurados para ir de -10 a 10 en ambas direcciones, con una cuadrícula activada. Se dibujan puntos grises cada 5 unidades en los ejes x e y. El vector representado es D = (-1, -5), comenzando en el origen (0, 0) y extendiéndose hacia el punto (-1, -5), representado por una flecha amarilla. El título de la gráfica indica que se trata de una variante del vector D = (-1, -5).**

[42]:
```
import numpy as np
import matplotlib.pyplot as plt

x1 = -10
x2 = 10
y1 = -10
y2 = 10
# Definir ejes
plt.axis([x1,x2,y1,y2])
# Agregamos los grid
plt.axis('on')
# Agregamos los grid
plt.grid(True)
```

```
# Agregamos un titulo en la grafica
plt.title('Variante del vector D = (2, -5)')

dx = 5
dy = 5
# Graficar puntos a mitad de las lineas (s = 1.5)
for x in np.arange(x1, x2, dx):
    for y in np.arange(y1, y2, dy):
        #plt.scatter(x_array, y_array, s_tamaño, color, etc)
        plt.scatter(x, y, s = 1.5, color = 'lightgray')

# Graficando el vector
        # x,  y, incremento, abscisa, longitud, ancho, color
plt.arrow(0, 0, 2, -5, head_length = 0.5, head_width = 0.3, color = "y")  #␣
 ↪Vector rojo
```

[42]: <matplotlib.patches.FancyArrow at 0x2213b7b49b0>

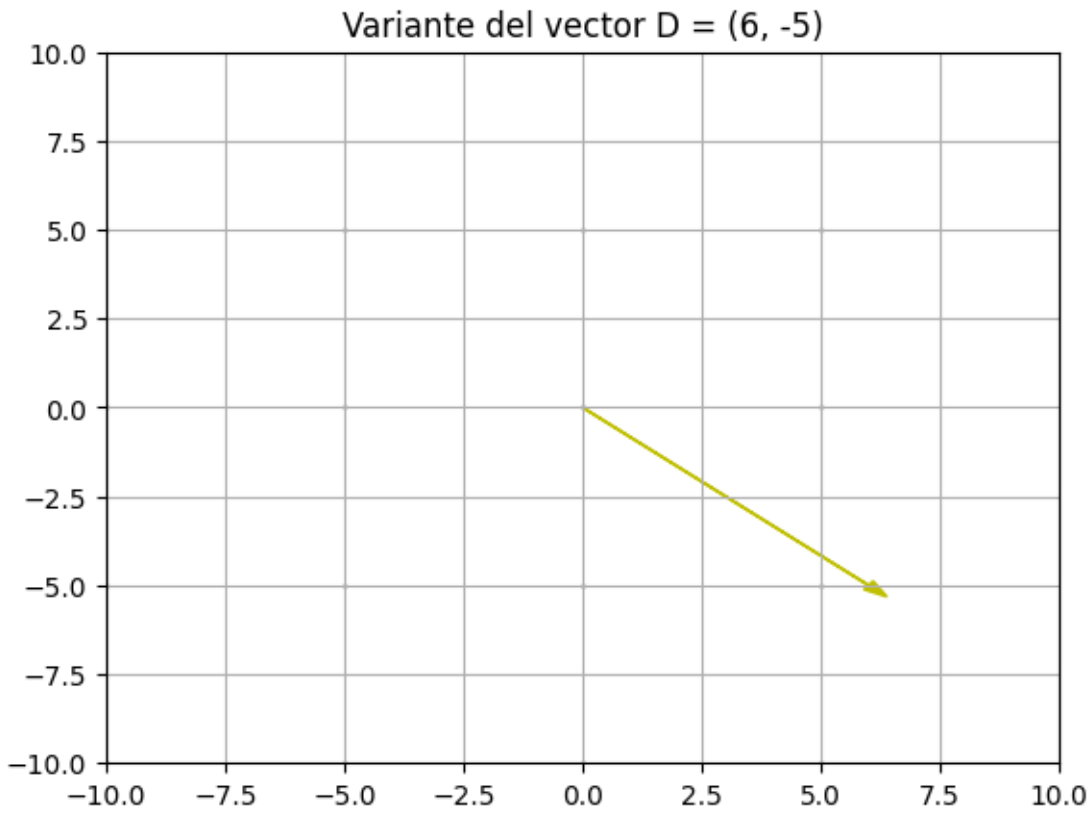



**Este código crea una gráfica 2D usando numpy y matplotlib, con una cuadrícula de puntos y un vector. Los ejes se extienden de -10 a 10 en ambas direcciones, y la cuadrícula está activada. Se dibujan puntos grises cada 5 unidades en los ejes x e y. El**

**vector representado es D = (2, -5), partiendo del origen (0, 0) y extendiéndose hasta el punto (2, -5), y está representado por una flecha amarilla. El título de la gráfica señala que se trata de una variante del vector D = (2, -5).**

[41]:
```
import numpy as np
import matplotlib.pyplot as plt

x1 = -10
x2 = 10
y1 = -10
y2 = 10
# Definir ejes
plt.axis([x1,x2,y1,y2])
# Agregamos los grid
plt.axis('on')
# Agregamos los grid
plt.grid(True)
# Agregamos un titulo en la grafica
plt.title('Variante del vector D = (6, -5)')

dx = 5
dy = 5
# Graficar puntos a mitad de las lineas (s = 1.5)
for x in np.arange(x1, x2, dx):
    for y in np.arange(y1, y2, dy):
        #plt.scatter(x_array, y_array, s_tamaño, color, etc)
        plt.scatter(x, y, s = 1.5, color = 'lightgray')

# Graficando el vector
        # x,  y, incremento, abscisa, longitud, ancho, color
plt.arrow(0, 0, 6, -5, head_length = 0.5, head_width = 0.3, color = "y")  #␣
 ↪Vector rojo
```

[41]:
```
<matplotlib.patches.FancyArrow at 0x2213b8f8620>
```

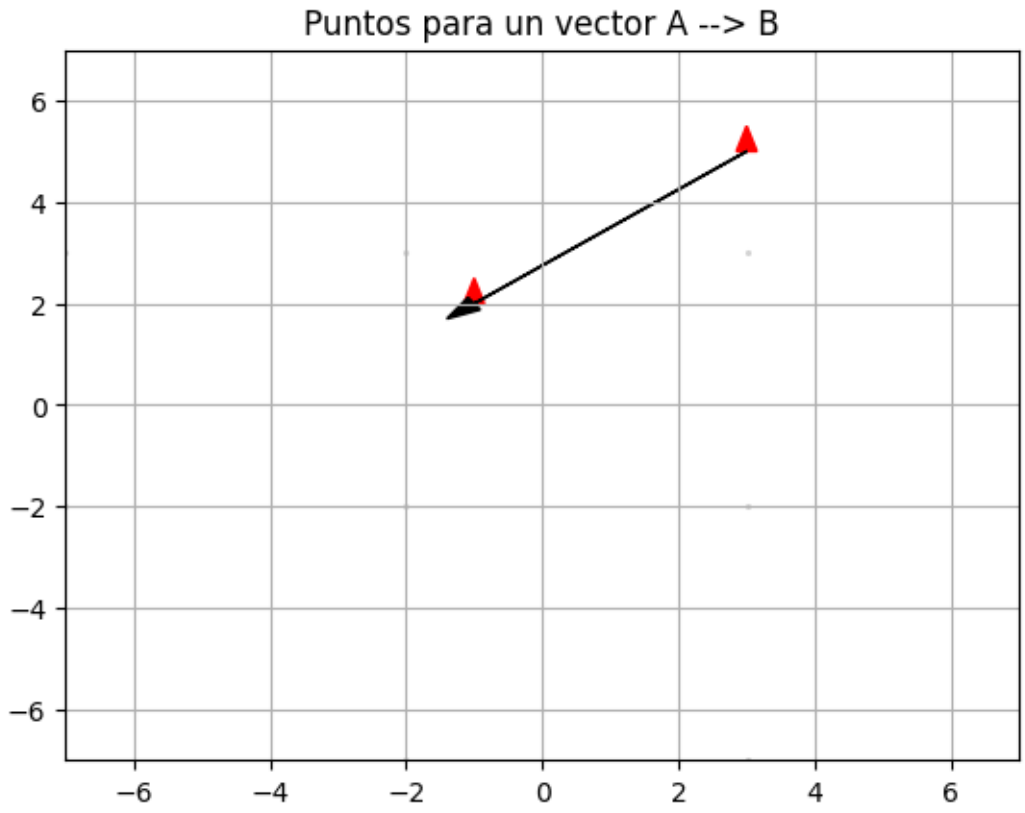



**Este código genera una gráfica 2D con numpy y matplotlib, mostrando una cuadrícula de puntos y un vector. Los ejes van de -10 a 10 en ambas direcciones, y se activa la cuadrícula. Se añaden puntos grises cada 5 unidades en los ejes x e y. El vector representado es D = (6, -5), que se dibuja como una flecha amarilla partiendo del origen (0, 0) y extendiéndose hacia el punto (6, -5). El título de la gráfica indica que es una variante del vector D = (6, -5).**

[69]:
```
import numpy as np
import matplotlib.pyplot as plt

x1 = -7
x2 = 7
y1 = -7
y2 = 7
# Definir ejes
plt.axis([x1,x2,y1,y2])
# Agregamos los grid
plt.axis('on')
# Agregamos los grid
plt.grid(True)
# Agregamos un titulo en la grafica
```

```
plt.title('Puntos para un vector A --> B')

dx = 5
dy = 5
# Graficar puntos a mitad de las lineas (s = 1.5)
for x in np.arange(x1, x2, dx):
    for y in np.arange(y1, y2, dy):
        #plt.scatter(x_array, y_array, s_tamaño, color, etc)
        plt.scatter(x, y, s = 1.5, color = 'lightgray')

# Graficando el vector
plt.arrow(3, 5, 0, 0, head_length = 0.5, head_width = 0.3, color = "r")
# Graficando
plt.arrow(-1,2, 0, 0, head_length = 0.5, head_width = 0.3, color = "r")
# Trazando el vector
plt.arrow(3, 5, -4, -3, head_length=0.5, head_width=0.3, color='black')
```

[69]: <matplotlib.patches.FancyArrow at 0x2213cdf1a60>

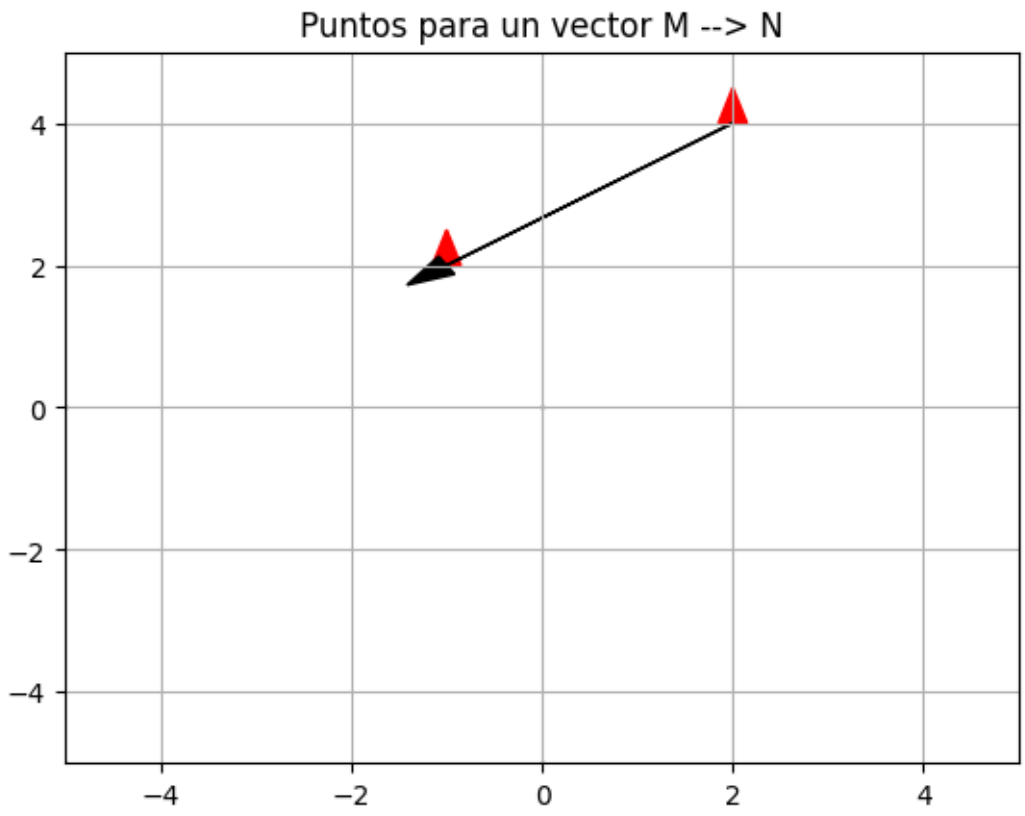



**Este código crea una gráfica 2D usando numpy y matplotlib, mostrando una cuadrícula de puntos y varios vectores. Los ejes están configurados entre -7 y 7 en ambas di-**

**recciones, con una cuadrícula visible. Se añaden puntos grises cada 5 unidades en los ejes x e y. En la gráfica se dibujan tres vectores: dos flechas rojas en las posiciones (3, 5) y (-1, 2) que están dibujadas sin extensión (representan los puntos finales de los vectores), y un vector negro que parte del punto (3, 5) y se extiende hacia el punto (-1, 2), con una dirección de (-4, -3). El título de la gráfica indica que se está mostrando la transición entre los puntos A y B.**

[92]:
```
import numpy as np
import matplotlib.pyplot as plt

x1 = -5
x2 = 5
y1 = -5
y2 = 5
# Definir ejes
plt.axis([x1,x2,y1,y2])
# Agregamos los grid
plt.axis('on')
# Agregamos los grid
plt.grid(True)
# Agregamos un titulo en la grafica
plt.title('Puntos para un vector M --> N')

dx = 5
dy = 5
# Graficar puntos a mitad de las lineas (s = 1.5)
for x in np.arange(x1, x2, dx):
    for y in np.arange(y1, y2, dy):
        #plt.scatter(x_array, y_array, s_tamaño, color, etc)
        plt.scatter(x, y, s = 1.5, color = 'lightgray')

# Graficando el vector
plt.arrow(2, 4, 0, 0, head_length = 0.5, head_width = 0.3, color = "r")
# Graficando
plt.arrow(-1,2, 0, 0, head_length = 0.5, head_width = 0.3, color = "r")
# Trazando el vector
plt.arrow(2, 4, -3, -2, head_length=0.5, head_width=0.3, color='black')
```

[92]:
```
<matplotlib.patches.FancyArrow at 0x221435516a0>
```

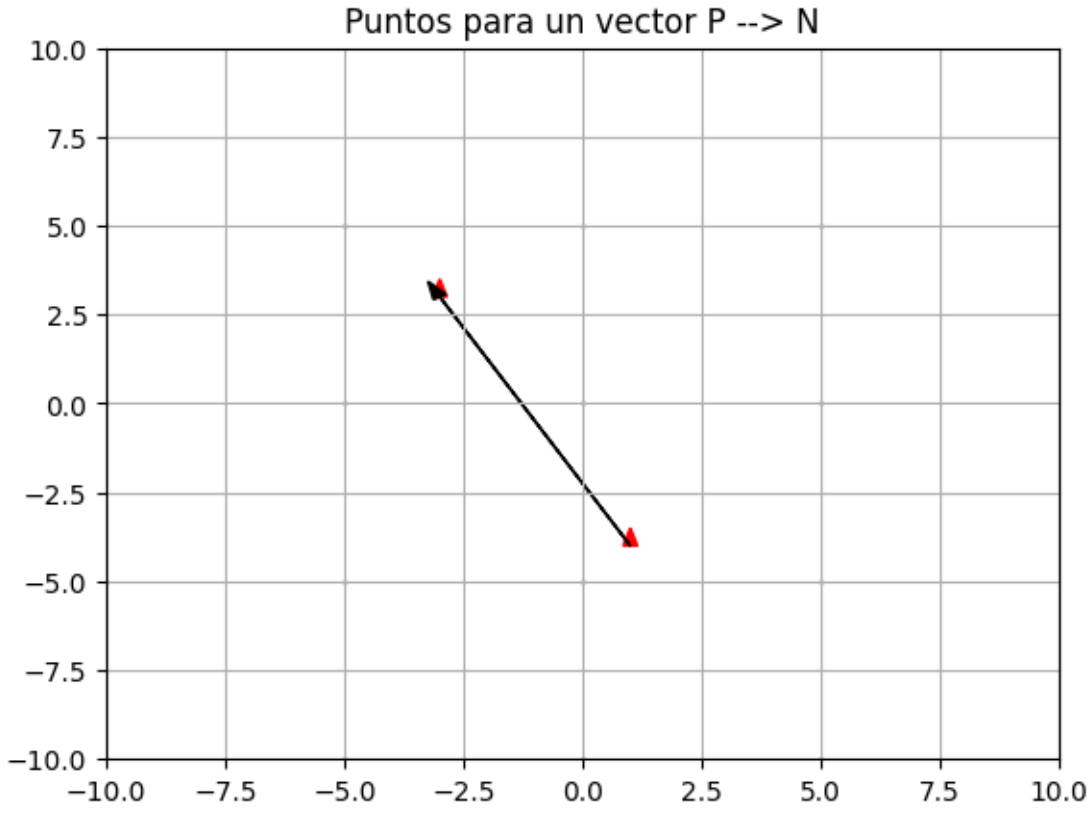



**Este código crea una gráfica 2D utilizando numpy y matplotlib, con una cuadrícula de puntos y vectores. Los ejes están configurados entre -5 y 5 en ambas direcciones, con la cuadrícula activada. Se dibujan puntos grises cada 5 unidades en los ejes x e y. En la gráfica, se muestran dos puntos finales de vectores en las posiciones (2, 4) y (-1, 2) como flechas rojas sin extensión, y un vector negro que parte del punto (2, 4) y se extiende hacia el punto (-1, 2), con una dirección de (-3, -2). El título de la gráfica indica que se está mostrando la transición entre los puntos M y N.**

[125]:
```
import numpy as np
import matplotlib.pyplot as plt

x1 = -10
x2 = 10
y1 = -10
y2 = 10
# Definir ejes
plt.axis([x1,x2,y1,y2])
# Agregamos los grid
plt.axis('on')
# Agregamos los grid
plt.grid(True)
```

```
# Agregamos un titulo en la grafica
plt.title('Puntos para un vector P --> N')

dx = 5
dy = 5
# Graficar puntos a mitad de las lineas (s = 1.5)
for x in np.arange(x1, x2, dx):
    for y in np.arange(y1, y2, dy):
        #plt.scatter(x_array, y_array, s_tamaño, color, etc)
        plt.scatter(x, y, s = 1.5, color = 'lightgray')

# Graficando el vector
plt.arrow(1, -4, 0, 0, head_length = 0.5, head_width = 0.3, color = "r")
# Graficando
plt.arrow(-3,3, 0, 0, head_length = 0.5, head_width = 0.3, color = "r")
# Trazando el vector
plt.arrow(1, -4, -4, 7, head_length=0.5, head_width=0.3, color='black')
```

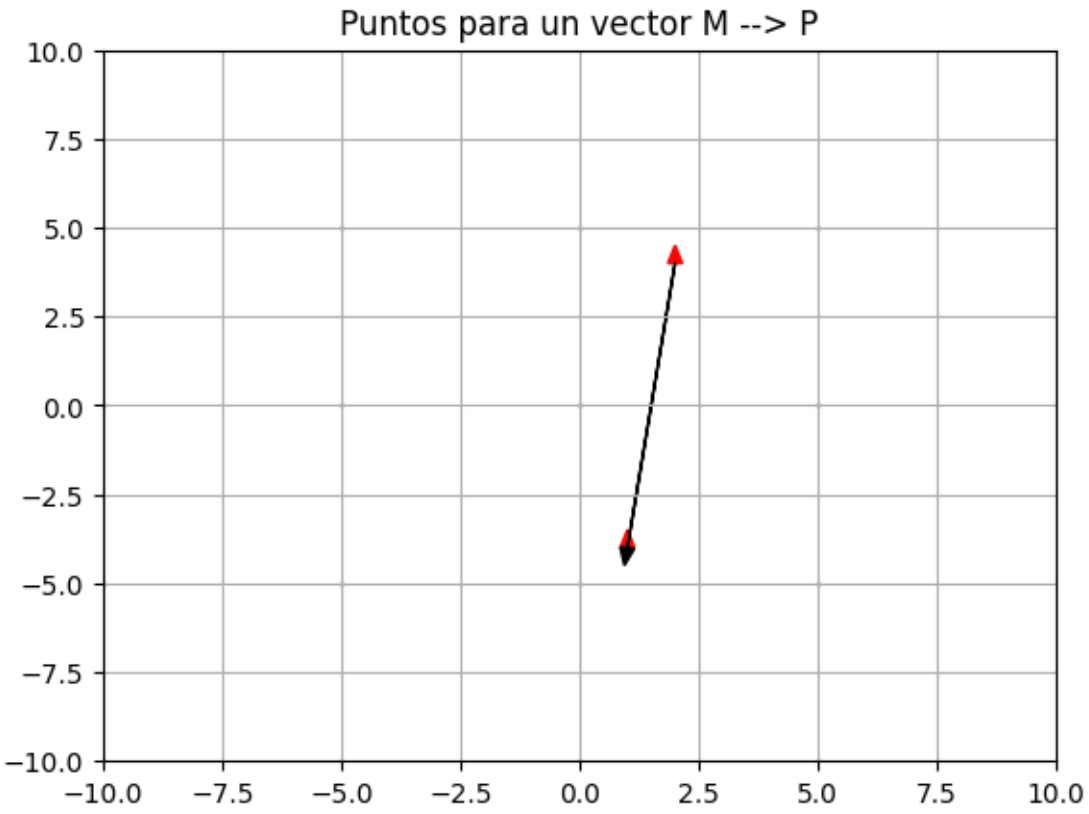
[125]: <matplotlib.patches.FancyArrow at 0x2214931a990>



**Este código crea una gráfica 2D utilizando numpy y matplotlib, mostrando una cuadrícula de puntos y un vector. Los ejes están configurados para ir de -10 a 10 en ambas direcciones, con la cuadrícula activada. Los puntos grises se dibujan cada 5 unidades en ambos ejes. En la gráfica, se incluyen dos puntos finales de vectores en las posiciones (1, -4) y (-3, 3) representados como flechas rojas sin extensión. Además, se dibuja un vector negro que parte del punto (1, -4) y se extiende hacia el punto (-3, 3), con una dirección de (-4, 7). El título de la gráfica indica que se está mostrando la transición entre los puntos P y N.**

[134]:
```
import numpy as np
import matplotlib.pyplot as plt

x1 = -10
x2 = 10
y1 = -10
y2 = 10
# Definir ejes
plt.axis([x1,x2,y1,y2])
# Agregamos los grid
plt.axis('on')
# Agregamos los grid
plt.grid(True)
# Agregamos un titulo en la grafica
plt.title('Puntos para un vector M --> P')

dx = 5
dy = 5
# Graficar puntos a mitad de las lineas (s = 1.5)
for x in np.arange(x1, x2, dx):
    for y in np.arange(y1, y2, dy):
        #plt.scatter(x_array, y_array, s_tamaño, color, etc)
        plt.scatter(x, y, s = 1.5, color = 'lightgray')

# Graficando el vector
plt.arrow(2, 4, 0, 0, head_length = 0.5, head_width = 0.3, color = "r")
# Graficando
plt.arrow(1,-4, 0, 0, head_length = 0.5, head_width = 0.3, color = "r")
# Trazando el vector
plt.arrow(2, 4, -1, -8, head_length=0.5, head_width=0.3, color='black')
```

[134]:
```
<matplotlib.patches.FancyArrow at 0x221492e6720>
```

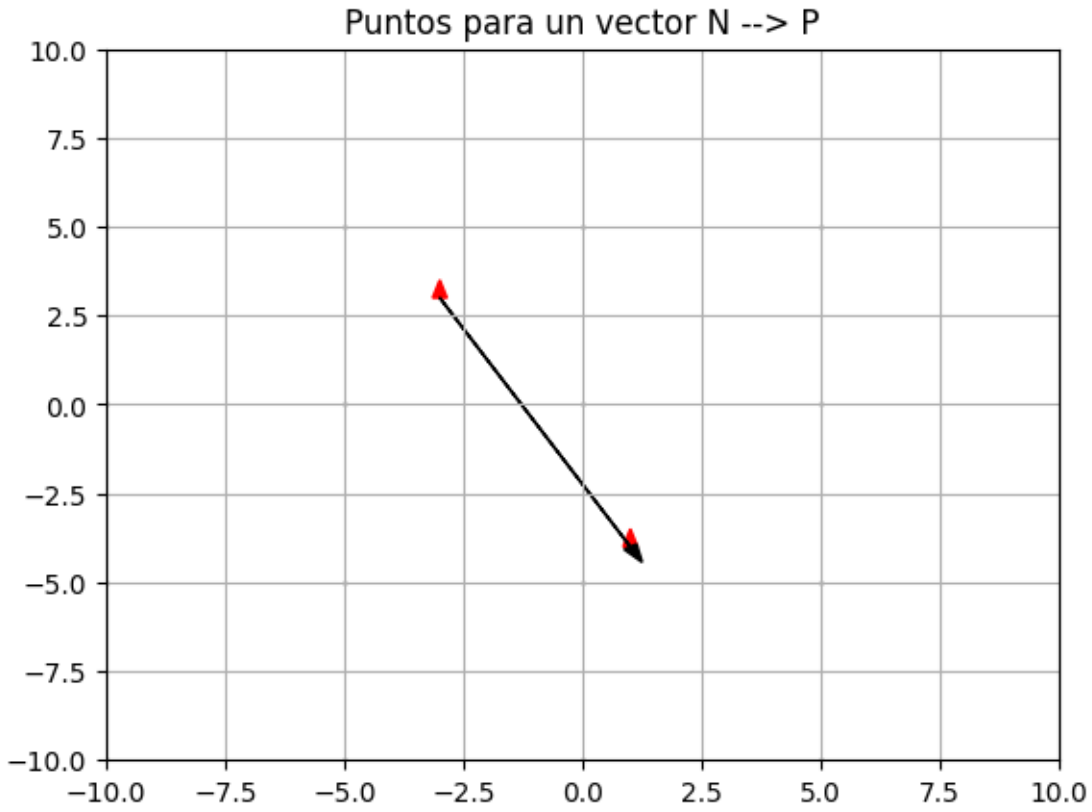



**Este código genera una gráfica 2D usando numpy y matplotlib, con una cuadrícula de puntos y un vector. Los ejes se extienden de -10 a 10 en ambas direcciones, y la cuadrícula está activada. Los puntos grises se dibujan cada 5 unidades en los ejes x e y. En la gráfica, se muestran dos puntos finales de vectores en las posiciones (2, 4) y (1, -4), representados como flechas rojas sin extensión. Además, se dibuja un vector negro que parte del punto (2, 4) y se extiende hacia el punto (1, -4), con una dirección de (-1, -8). El título de la gráfica indica que se está mostrando la transición entre los puntos M y P.**

[136]:
```
import numpy as np
import matplotlib.pyplot as plt

x1 = -10
x2 = 10
y1 = -10
y2 = 10
# Definir ejes
plt.axis([x1,x2,y1,y2])
# Agregamos los grid
plt.axis('on')
# Agregamos los grid
```

```
plt.grid(True)
# Agregamos un titulo en la grafica
plt.title('Puntos para un vector N --> P')

dx = 5
dy = 5
# Graficar puntos a mitad de las lineas (s = 1.5)
for x in np.arange(x1, x2, dx):
    for y in np.arange(y1, y2, dy):
        #plt.scatter(x_array, y_array, s_tamaño, color, etc)
        plt.scatter(x, y, s = 1.5, color = 'lightgray')

# Graficando el vector
plt.arrow(-3, 3, 0, 0, head_length = 0.5, head_width = 0.3, color = "r")
# Graficando
plt.arrow(1,-4, 0, 0, head_length = 0.5, head_width = 0.3, color = "r")
# Trazando el vector
plt.arrow(-3, 3, 4, -7, head_length=0.5, head_width=0.3, color='black')
```

[136]: <matplotlib.patches.FancyArrow at 0x2214ac5eab0>



**Este código produce una gráfica 2D usando numpy y matplotlib, con una cuadrícula de puntos y un vector. Los ejes se extienden de -10 a 10 en ambas direcciones, y la cuadrícula está activada. Los puntos grises se dibujan cada 5 unidades en los ejes x e y. En la gráfica, se muestran dos puntos finales de vectores en las posiciones (-3, 3) y (1, -4) como flechas rojas sin extensión. Además, se dibuja un vector negro que parte del punto (-3, 3) y se extiende hacia el punto (1, -4), con una dirección de (4, -7). El título de la gráfica indica que se está mostrando la transición entre los puntos N y P.**