



Links compartidos en el chat de la sesión:

- Colab Optimización I: (crear una copia para trabajar)
https://colab.research.google.com/drive/1hiLvCHd3m726LCbUEF0VvhQD0_bt9PFJ?usp=sharing
- Ejemplo Problemas Discretos:
<https://colab.research.google.com/drive/1EFOzkxidSuXpysuSK39kNnELyPatIEjL?usp=sharing>
- Libro: Linear Algebra and Optimization for Machine Learning
https://drive.google.com/file/d/1nklULEPWlu5Pmdk3nCJkZISS7QzqTPuY/view?usp=drive_link





- Código para ver la gráfica:
(aporte del chat)



Comparto el código con la gráfica:

```
import numpy as np
from scipy.optimize import minimize
import matplotlib.pyplot as plt

# Definir la función objetivo
def f(x):
    return x**2 - 4*x + 4

# Inicializar el valor de x
x0 = 0

# Usar el método de optimización para minimizar la función
result = minimize(f, x0)

# Imprimir el resultado
print("El valor de x que minimiza la función es:", result.x[0])

# Generar puntos para graficar la función
x = np.linspace(-2, 6, 100) # Rango de valores de x
y = f(x) # Valores correspondientes de y

# Graficar la función
plt.plot(x, y, label='f(x) = x^2 - 4x + 4')
plt.scatter(result.x[0], f(result.x[0]), color='red', label=f'Valor mínimo en x = {result.x[0]}')
plt.ylabel('f(x)')
plt.legend()
plt.grid(True)
plt.show()

return go(f, seed, [])
}
```





Recuerda que en el [panel del Bootcamp](#) encuentras el plan de estudios, y enlaces a las clases en vivo, grabaciones, recursos, presentaciones, y más.

Te recomendamos que te unas al [canal de anuncios del bootcamp](#) sirve como bookmark dónde encontrarás todos los enlaces, archivos y notas que necesites.

