

```

import numpy as np
import matplotlib.pyplot as plt
from scipy.optimize import curve_fit

datos_next_fit = [
    (100_000, 0.003993),
    (1_000_000, 0.040563),
    (10_000_000, 0.398887),
    (20_000_000, 0.788990),
    (30_000_000, 1.576250),
    (40_000_000, 2.067115),
    (50_000_000, 2.587466),
    (60_000_000, 3.349409),
    (70_000_000, 3.729392),
    (80_000_000, 4.286489),
    (90_000_000, 5.001271),
    (100_000_000, 5.362280)
]

steps = np.array([d[0] for d in datos_next_fit])
times = np.array([d[1] for d in datos_next_fit])

plt.figure(figsize=(10, 6))

# Usamos plot con línea y marcadores para ver la tendencia claramente
plt.plot(steps, times, marker='o', linestyle='--', color='navy',
        linewidth=2, markersize=8, label='Datos Experimentales')

plt.title('Tiempo de Ejecución Empírica (Next Fit)', fontsize=16)
plt.xlabel('Tamaño de entrada (N)', fontsize=12)
plt.ylabel('Tiempo (Segundos)', fontsize=12)
plt.grid(True, which='both', linestyle='--', linewidth=0.5)
plt.ticklabel_format(style='plain', axis='x')
plt.xticks(rotation=45)
plt.tight_layout()
plt.show()

# Definimos la función teórica LINEAL: f(N) = C * N
def func_lineal(N, C):
    return C * N

# Ajuste de curva (Encontrar C)
popt, pcov = curve_fit(func_lineal, steps, times)
C_optimo = popt[0]

print(f"El valor óptimo para la constante C es: {C_optimo:.10g}")

# Generar datos para la línea suave (Teórica)
N_teorico = np.linspace(min(steps), max(steps), 400)
y_teorico = func_lineal(N_teorico, C_optimo)

```

```

plt.figure(figsize=(10, 6))

# Puntos experimentales (Scatter)
plt.scatter(steps, times, color='blue', s=100, zorder=5, label='Datos Experimentales')

# Línea teórica
plt.plot(N_teorico, y_teorico, color='red', linewidth=2,
         label=f'Ajuste Teórico ($C \cdot N$)\n$C \approx {C_optimo:.2e}$')

plt.title('Next Fit: Comparación Empírica vs Teórica Lineal',
          fontsize=16)
plt.xlabel('Tamaño de entrada (N)', fontsize=12)
plt.ylabel('Tiempo (Segundos)', fontsize=12)

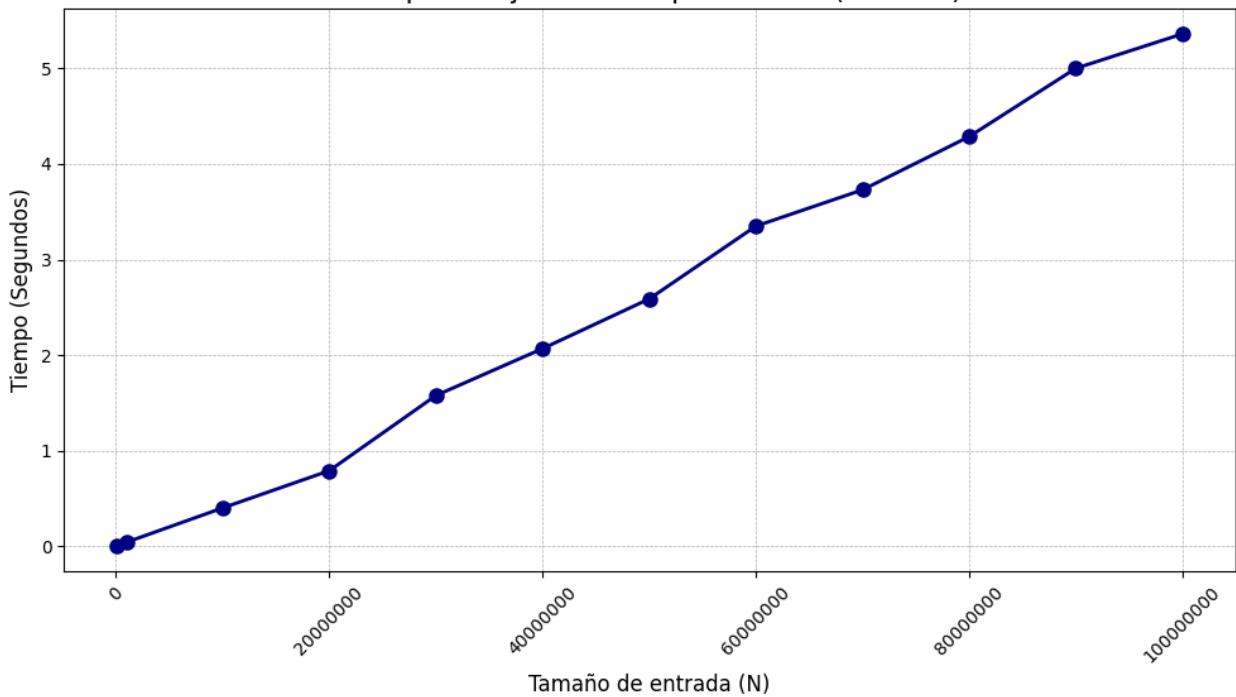
plt.legend()
plt.grid(True, which='both', linestyle='--', linewidth=0.5)
plt.ylim(bottom=0, top=max(times) * 1.1)
plt.ticklabel_format(style='plain', axis='x')
plt.xticks(rotation=45)

plt.tight_layout()
plt.show()

<>:67: SyntaxWarning: invalid escape sequence '\c'
<>:67: SyntaxWarning: invalid escape sequence '\c'
/tmp/ipython-input-717083826.py:67: SyntaxWarning: invalid escape
sequence '\c'
    label=f'Ajuste Teórico ($C \cdot N$)\n$C \approx {C_optimo:.2e}$')

```

### Tiempo de Ejecución Experimental (Next Fit)



El valor óptimo para la constante C es:  $5.37762132e-08$

### Next Fit: Comparación Empírica vs Teórica Lineal

