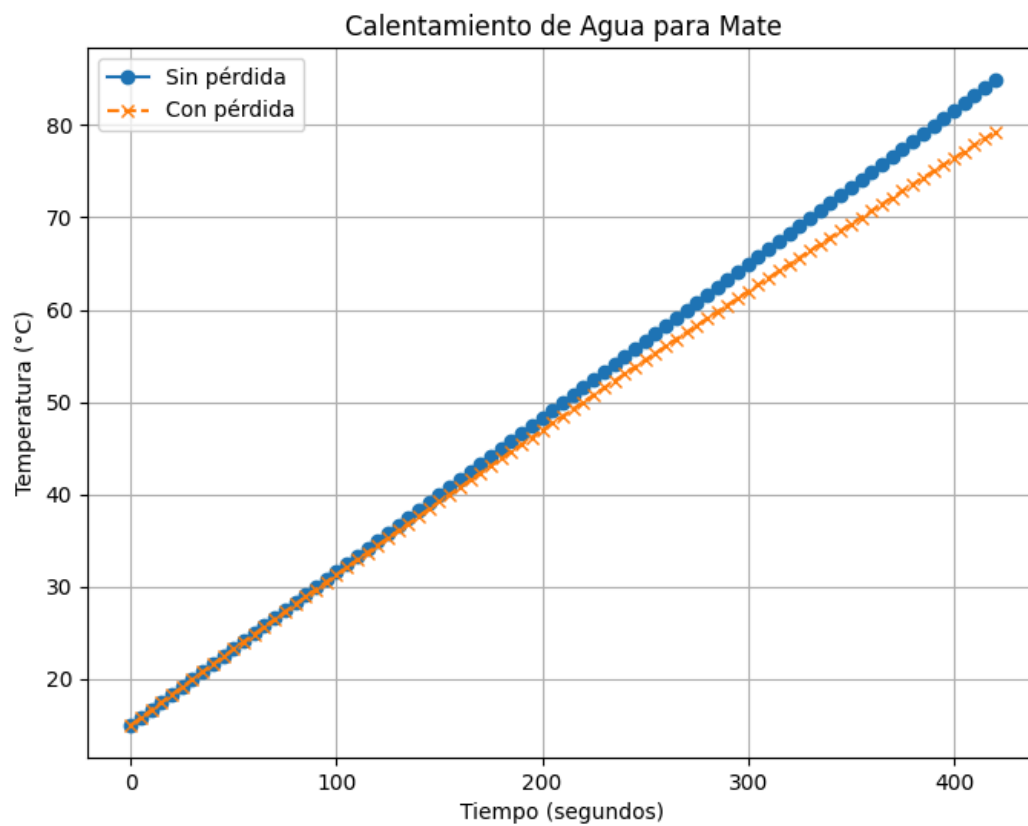


TP4: Graficar la temperatura del fluido dentro del calentador sin pérdidas y con pérdidas para cada tick de tiempo, hasta llegar al tiempo deseado para que el dispositivo cumpla su tarea.

Fórmula:

$$temp\ nueva = temp\ inicial + tasa\ de\ aumento * tiempo - \frac{p\acute{e}rdida}{masa * calor\ esp} (temp\ anterior - 15) * tiempo$$

$$temp\ nueva = 15 + 0,1664 * tiempo - \frac{13288}{1.5 * 4180} (temp\ anterior - 15) * tiempo$$



```
import numpy as np

import matplotlib.pyplot as plt

# Datos

t_discreto = np.arange(0, 421, 5) # Tiempo en segundos (discreto,
intervalo de 5 segundos)

tasa_aumento = 0.1664 # Tasa de aumento de temperatura (°C por segundo)

temperatura_inicial = 15 # Temperatura inicial en °C

tasa_perdida_calor = 1.3288 # Tasa de pérdida de calor (W/°C)
```

```

masa_termo = 1.5 # Masa del termo en kilogramos

calor_especifico_agua = 4180 # Calor específico del agua en J/kg·°C

# Calculamos la temperatura en función del tiempo (sin pérdida de calor)
temperatura_sin_perdida = tasa_aumento * t_discreto + temperatura_inicial

# Calculamos la temperatura con pérdida
temperatura_con_perdida = []

temperatura_anterior = temperatura_inicial

for t in t_discreto:

    #delta_temp = (tasa_aumento * t - tasa_perdida_calor) / (masa_termo *
calor_especifico_agua)

    #temperatura_nueva = temperatura_anterior + delta_temp *
(temperatura_anterior - temperatura_inicial)

    delta_temp = 1.3288 * t * (temperatura_anterior - 15) / (1.5 * 4180)

    temperatura_nueva = 15 + tasa_aumento * t - delta_temp

    temperatura_con_perdida.append(temperatura_nueva)

    temperatura_anterior = temperatura_nueva

# Graficamos ambas temperaturas

plt.figure(figsize=(8, 6))

plt.plot(t_discreto, temperatura_sin_perdida, marker="o", linestyle="--",
label="Sin pérdida")

plt.plot(t_discreto, temperatura_con_perdida, marker="x", linestyle="--",
label="Con pérdida")

plt.xlabel("Tiempo (segundos)")

plt.ylabel("Temperatura (°C)")

```

```
plt.title("Calentamiento de Agua para Mate")

plt.grid(True)

plt.legend()

plt.show()
```