Guía de laboratorio

Laboratorio de Contenido 2: Quick Sort en priorización de alertas
Explicación básica
Quick Sort también usa Divide y Vencerás, pero selecciona un pivote y divide los datos en menores y mayores al pivote. Tiene rendimiento promedio O(n log n), aunque puede degradarse a $O(n^2)$.

Objetivos

- Comprender el concepto de pivote.
- Evaluar ventajas prácticas frente a otros algoritmos.
- Aplicarlo a priorización de eventos.

Caso práctico
Un SIEM necesita ordenar alertas por nivel de severidad (1–100).

Problema
Ordenar niveles de severidad:
[70, 15, 90, 45, 30, 85, 60]

Ejemplo paso a paso
Pivote: 60
Menores:
[15,45,30]
Mayores:
[70,90,85]
Resultado final:
[15,30,45,60,70,85,90]

Implementación en Python

```
def quick_sort(arr):
    if len(arr) <= 1:
        return arr

    pivot = arr[len(arr)//2]
    left = [x for x in arr if x < pivot]
    middle = [x for x in arr if x == pivot]
    right = [x for x in arr if x > pivot]

    return quick_sort(left) + middle + quick_sort(right)
```

```
datos = [70, 15, 90, 45, 30, 85, 60]
print(quick_sort(datos))
```

Conclusiones

Quick Sort es extremadamente rápido en la práctica y usa poca memoria. Es ideal cuando se requiere rapidez promedio en clasificación de eventos, aunque se debe cuidar la selección del pivote.