Optimización y Performance - Día 4

Change Status

pending 40 min

Learning Objectives

1. Entender conceptos básicos de optimización de performance
2. Aprender técnicas de mejora de velocidad en pipelines
3. Comprender monitoreo de recursos y cuellos de botella
4. Conocer estrategias de escalado horizontal y vertical

Practical exercise to apply the concepts learned.

Ejercicio: Optimizar una función de procesamiento de datos

Función original (lenta):

```
def procesar_datos_lento(datos):
    """Procesamiento ineficiente"""
    resultado = []
    for fila in datos:
        # Procesamiento secuencial
        fila_procesada = fila.copy()
        fila_procesada['total'] = fila['precio'] * fila['cantidad']
        resultado.append(fila_procesada)
    return resultado
```

Función optimizada:

```
def procesar_datos_rapido(datos):
    """Procesamiento optimizado con comprensión de listas"""
    return [
        {**fila, 'total': fila['precio'] * fila['cantidad']}
        for fila in datos
    ]
```

Comparar performance:

```
import time

datos_prueba = [{'precio': i, 'cantidad': i%10} for i in range(10000)]

# Medir versión lenta
inicio = time.time()
procesar_datos_lento(datos_prueba)
tiempo_lento = time.time() - inicio

# Medir versión rápida
inicio = time.time()
procesar_datos_rapido(datos_prueba)
tiempo_rapido = time.time() - inicio

print(f"Lento: {tiempo_lento:.3f}s, Rápido: {tiempo_rapido:.3f}s")
print(f"Mejora: {tiempo_lento/tiempo_rapido:.1f}x más rápido")
```

Verificación: ¿Cuándo deberías optimizar performance? ¿Qué trade-offs considerar entre velocidad y complejidad?

Requerimientos:

Conocimiento básico de Python
Familiaridad con conceptos de performance
Comprensión de algoritmos básicos