

Implementación de Pipeline End-to-End - Día 2

Change Status

pending 40 min

Learning Objectives

- 1 Entender componentes básicos de un pipeline end-to-end
- 2 Aprender gestión de estado y dependencias entre tareas
- 3 Comprender estrategias de manejo de errores
- 4 Conocer importancia del monitoreo y debugging

Theory

Practice

Evidence

Quiz

<> Practical exercise to apply the concepts learned.

Ejercicio: Diseñar pipeline básico con manejo de errores

Definir componentes del pipeline:

```
pipeline_simple = {
    'paso_1': 'Capturar datos de API',
    'paso_2': 'Validar y limpiar datos',
    'paso_3': 'Guardar en base de datos'
}
```

Implementar manejo básico de errores:

```
def ejecutar_pipeline_con_errores(pipeline):
    for paso, descripcion in pipeline.items():
        try:
            print(f"Ejecutando {paso}: {descripcion}")
            # Simular ejecución
            if paso == 'paso_2':
                raise ValueError("Error de validación")
        except Exception as e:
            print(f"Error en {paso}: {e}")
            # Aquí iría lógica de recovery
```

Verificación: ¿Qué diferencia hay entre un pipeline que falla silenciosamente y uno con buen manejo de errores? ¿Cómo decidirías cuándo reintentar vs abortar una ejecución?

Requerimientos:

Conocimiento básico de Python

Familiaridad con conceptos de logging

Comprensión de sistemas distribuidos