Validación y Testing - Día 3

Change Status

completed · 40 min

Learning Objectives

1. Entender importancia del testing en pipelines de datos
2. Aprender tipos básicos de tests (unitarios, integración)
3. Comprender validación de calidad de datos
4. Conocer estrategias de testing automatizado

Practical exercise to apply the concepts learned.

Ejercicio: Crear suite básica de tests para un pipeline simple

Definir función a testear:

```
def calcular_total_ventas(ventas):
    """Calcula total de ventas por producto"""
    totales = {}
    for venta in ventas:
        producto = venta['producto']
        cantidad = venta['cantidad']
        precio = venta['precio']
        totales[producto] = totales.get(producto, 0) + (cantidad * precio)
    return totales
```

Crear tests unitarios:

```
def test_calculo_totales():
    # Datos de prueba
    ventas = [
        {'producto': 'A', 'cantidad': 2, 'precio': 10},
        {'producto': 'B', 'cantidad': 1, 'precio': 20},
        {'producto': 'A', 'cantidad': 3, 'precio': 10}
    ]

    resultado = calcular_total_ventas(ventas)

    # Verificaciones
    assert resultado['A'] == 50, f"Producto A: expected 50, got {resultado['A']}"
    assert resultado['B'] == 20, f"Producto B: expected 20, got {resultado['B']}"
    print("✅ Todos los tests pasaron!")
```

Verificación: ¿Por qué es importante testear pipelines de datos? ¿Qué tipos de errores son más comunes en pipelines y cómo detectarlos con tests?

Requerimientos:

Conocimiento básico de Python
Familiaridad con conceptos de testing
Comprensión de calidad de datos