# Transformaciones Avanzadas y Enriquecimiento - Día 3

Change Status

pending · 40 min

## Learning Objectives

1. Entender operaciones avanzadas de transformación
2. Aprender joins y merges entre datasets
3. Comprender agregaciones y cálculos derivados
4. Conocer validaciones de integridad



Practical exercise to apply the concepts learned.

**Ejercicio**: Transformaciones avanzadas en dataset de e-commerce

**Datos base:**

```
import pandas as pd
import numpy as np

# Clientes
clientes = pd.DataFrame({
    'cliente_id': range(1, 6),
    'nombre': ['Ana', 'Juan', 'María', 'Pedro', 'Laura'],
    'segmento': ['Premium', 'Regular', 'Premium', 'Regular', 'VIP']
})

# Pedidos
pedidos = pd.DataFrame({
    'pedido_id': range(1, 11),
    'cliente_id': np.random.choice(range(1, 6), 10),
    'producto': np.random.choice(['A', 'B', 'C', 'D'], 10),
    'precio': np.random.uniform(50, 500, 10).round(2),
    'fecha': pd.date_range('2024-01-01', periods=10)
})

print("Clientes y pedidos cargados")
```

**Enriquecer datos con joins:**

```
# Unir pedidos con información de clientes
pedidos_enriquecidos = pd.merge(
    pedidos,
    clientes,
    on='cliente_id',
    how='left'
)

print("Pedidos con información de clientes:")
print(pedidos_enriquecidos.head())
```

**Calcular métricas derivadas:**

```
# Calcular métricas por cliente
metricas_cliente = pedidos_enriquecidos.groupby(['cliente_id', 'nombre', 'segmento']).agg({
    'pedido_id': 'count',
    'precio': ['sum', 'mean', 'max'],
    'fecha': 'max'  # Última compra
}).round(2)

# Aplanar columnas multi-nivel
metricas_cliente.columns = ['num_pedidos', 'total_gastado', 'gasto_promedio', 'gasto_maximo', 'ultima_compra']
metricas_cliente = metricas_cliente.reset_index()

print("\nMétricas por cliente:")
print(metricas_cliente)
```

**Validar reglas de negocio:**

```python
def validar_reglas_negocio(df):
    validaciones = []

    # VIP deben tener al menos 2 pedidos
    vip_insuficientes = df[(df['segmento'] == 'VIP') & (df['num_pedidos'] < 2)]
    if len(vip_insuficientes) > 0:
        validaciones.append(f"VIPs con pocos pedidos: {len(vip_insuficientes)}")

    # Premium no deben exceder gasto máximo
    premium_excesivos = df[(df['segmento'] == 'Premium') & (df['gasto_maximo'] > 800)]
    if len(premium_excesivos) > 0:
        validaciones.append(f"Premiums con gastos excesivos: {len(premium_excesivos)}")

    return validaciones

reglas_incumplidas = validar_reglas_negocio(metricas_cliente)
print(f"\nReglas de negocio incumplidas: {reglas_incumplidas}")
```

**Verificación**: ¿Qué tipo de join usarías cuando quieres mantener todos los registros de una tabla principal? ¿Cómo decides qué métricas calcular para un análisis específico?

**Requerimientos:**

Pandas para manipulación de datos
Comprensión de operaciones de conjunto
Conocimiento de reglas de negocio