

## 1. Capa

- **Título:** Automação de Processos de ETL: Do Power Query ao Python (Pandas)
  - **Autor:** Vinicius alves da silva
  - **Data:** Janeiro de 2026
  - **Tecnologias:** Python, Pandas, Numpy, Excel.
- 

## 2. Introdução

Neste projeto, apresento a migração de um fluxo de tratamento de dados (ETL) originalmente planejado para o Power Query para o ecossistema Python. O objetivo principal foi demonstrar como a biblioteca **Pandas** pode oferecer maior controle e escalabilidade no tratamento de grandes volumes de dados.

---

## 3. O Desafio (Problema)

A base de dados bruta apresentava inconsistências comuns que impediam a análise imediata:

- Nomes de jogos e plataformas concatenados em uma única string.
  - Prefixos desnecessários em códigos de lote (Batch).
  - Valores nulos representados por caracteres especiais (---).
  - Preços em formato de texto com diferentes símbolos monetários.
- 

## 4. Implementação da Solução (O Código)

*Dica: Use uma fonte monoespaciada como Courier New para o código abaixo.*

```
import pandas as pd
from datetime import datetime
import numpy as np

print('Iniciando a leitura do arquivo Excel...')
#lê o arquivo Excel
df = pd.read_excel('prova/dio.xlsx')
```

```
nintendo = df.copy()

print('leitura concluída')

#altera a coluna 'Game' para separar o nome do jogo e a plataforma
nintendo[['Game', 'Platform']] = nintendo['Game'].str.rsplit(' - ', n=1,
expand=True)

#remove o prefixo 'Batch-' da coluna 'Batch' e espaços em branco
nintendo['Batch'] =
nintendo['Batch'].str.removeprefix('Batch-').str.strip()

#substitui os valores '----', '---', '----' por NaN
nintendo = nintendo.replace(['----', '---', '----'], np.nan)

#remove linhas onde todas as colunas são NaN
nintendo = nintendo.dropna(how='all')

#altera a coluna 'Send Date' para o formato datetime
nintendo['Send Date'] = pd.to_datetime(nintendo['Send Date'],
dayfirst=True, errors='coerce')

#mapeia os valores da coluna 'Region' para nomes completos
TRADUCAO = {
    'PT': 'Português',
    'EN': 'Inglês',
    'JP': 'Japonês'
}

nintendo['Region'] = nintendo['Region'].map(TRADUCAO)

#remove o símbolo de moeda e converte a coluna 'Preço Unit Price' para
float
nintendo['Preço Unit Price'] = nintendo['Preço Unit Price'].apply(lambda x
: float(str(x).replace('$',
```

```
    '').replace('R$', '').replace(',', '.').strip()) if isinstance(x, str) else x
)

#cria a coluna 'preco_total' como o produto da 'Stock Quantity' e 'Preço
Unit Price'
nintendo['preco_total'] = nintendo ['Stock Quantity'] * nintendo ['Preço
Unit Price']

print('Iniciando a exportação do arquivo Excel...')
#exporta o DataFrame limpo para um novo arquivo Excel
nintendo.to_excel('prova/nitendo.xlsx', index=False)

print('Arquivo exportado com sucesso!')
```

## 5. Conclusão e Resultados

Com a implementação em Python, o processo tornou-se:

1. **Reprodutível:** O script pode ser executado para novos arquivos instantaneamente.
2. **Auditável:** Cada etapa da limpeza está documentada em linhas de código claras.
3. **Flexível:** A facilidade de lidar com diferentes moedas e padrões de texto supera as limitações de interface visual do Power Query.