



Universidade dos Dados

# Projeto: Análise de NPS

**Live: 18/02/2025**

Python,PostgresSql,Azure Data Factory, docker e Streamlit



Github: <https://github.com/Walterbiel/NPS-live-18-02>

# Requisitos do negócio

## Objetivo:

- Entender a satisfação dos clientes e identificar oportunidades de melhoria nos produtos, serviços e atendimento.
- Correlacionar o NPS com dados de vendas para avaliar impacto em retenção e conversão.
- Extrair insights açãoáveis a partir das análises descritivas dos feedbacks dos clientes.

## Necessidades:

### 1 Coleta de Dados

- Automação da captura do NPS via pesquisas pós-compra, e-mails, WhatsApp ou sistemas de CRM.
- Armazenamento estruturado dos dados em um banco de dados ou plataforma de BI.
- Captura de feedbacks textuais, garantindo pelo menos 50% das respostas com comentários abertos.

### 2 Análises Essenciais

- Distribuição do NPS: Promotores, Neutros e Detratores ao longo do tempo.
- Correlação NPS x Receita: Impacto da satisfação na recorrência e no ticket médio.
- NPS Mensal e Média Trimestral: Monitoramento da evolução da satisfação dos clientes.
- Principais Motivos: Identificação de padrões nos feedbacks, como preço, atendimento e entrega.
- Clusters de Clientes: Agrupamento de clientes por perfis baseados no NPS.

### 3 Visualização e Ações

- Dashboards Interativos com métricas-chave e insights visuais.
- Alertas para NPS baixo para ação rápida em clientes insatisfeitos.
- Acompanhamento de tendências para avaliar melhorias após mudanças estratégicas.

# O que vamos aprender:

## 1 Entendimento de bases de texto e NPS:

Estrutura de uma base de pesquisa NPS

Como capturar e organizar feedbacks de clientes

Tipos de análise: quantitativa (NPS) e qualitativa (textos)

## 2 Event Producer com Python e Faker:

Geração de dados fictícios para simular um ambiente realista

Uso da biblioteca Faker para criar feedbacks de clientes

Publicação de eventos simulados para ingestão de dados

## 3 API com FastAPI:

Construção de uma API rápida e eficiente

Endpoints para recebimento e consulta dos feedbacks

Testes com Postman para validar chamadas à API

## 4 Ingestão de Dados com SQLAlchemy no PostgreSQL:

Modelagem e persistência dos dados no banco

Configuração do SQLAlchemy para interagir com o PostgreSQL

Boas práticas para armazenar NPS e feedbacks textuais

## 5 Azure Data Factory e Postman para Orquestração:

Como automatizar fluxos de dados entre APIs e banco de dados

Uso do Postman para testar chamadas e monitorar requisições

Implementação de pipelines no Azure Data Factory

## 6 Análises com Pandas e Seaborn:

Manipulação dos dados para entender padrões no NPS

Análise de sentimentos dos feedbacks textuais

Visualização de tendências usando Seaborn

## 7 Dashboard Interativo com Streamlit:

Construção de um painel dinâmico para explorar os insights

Aplicação prática de análises visuais para tomada de decisão

Integração com APIs para atualizar os dados em tempo real

# Arquitetura da solução

