

## Relatório: Solução case Americanas

Autor: David Vieira

Data: Dezembro/2025

Assunto: Estratégia de Pricing, Análise de Elasticidade e Implementação Técnica

### 1. Sumário Executivo: Estratégia de Negócio

(Resumo para Diretoria Comercial e Tomadores de Decisão - **Parte 6**)

#### Recomendação Principal

**RECOMENDAÇÃO: NÃO** iniciar campanha de descontos agressivos para os produtos analisados (P2, P5, P8).

Nossa análise indica que a demanda atual desses produtos é **inelástica** ao preço. Isso significa que baixar o preço **não trará volume de vendas suficiente** para compensar a redução de margem, resultando em destruição de lucro.

#### Dados de Suporte

- **Análise Realizada:** Modelagem econométrica em cima de histórico de vendas e monitoramento de concorrência.
- **Cenário Simulado:** Descontos de 10% a 30%.
- **Resultado:** Em 100% dos cenários simulados, o Lucro Incremental foi **Negativo**.
  - *Exemplo Padrão:* Um desconto de 10% geraria um aumento de venda estimado de apenas 2-3%, resultando em uma queda de receita total de ~7%.

#### Próximos Passos & Oportunidades

Em vez de queimar margem com descontos ineficazes, sugerimos:

1. **Defesa de Margem:** Manter preços atuais. A venda não cairá significativamente.
2. **Teste de Aumento de Preço:** Rodar piloto em praça controlada (ex: Região Sul) com aumento de 3-5% no preço do Produto P8. Estimamos potencial aumento direto de lucro líquido dado o comportamento inelástico.
3. **Investigação de Mix:** Focar análise em produtos de "Cauda Longa" que não entraram nesta modelagem inicial, onde a sensibilidade a preço pode ser maior.

## 2. Relatório Técnico: Metodologia e Implementação

(Detalhamento da abordagem técnica, estratégica e de engenharia de dados)

Este consolidado detalha minha abordagem técnica para as 7 etapas do desafio de Pricing e Revenue Growth Management (RGM).

### Parte 1: Engenharia & Preparação dos Dados (GCP)

Para garantir escalabilidade e governança, desenhei uma arquitetura **ELT (Extract, Load, Transform)** utilizando os serviços nativos do GCP.

- **Ingestão (Extract/Load):**

Optei por utilizar o Cloud Storage como Landing Zone para receber os arquivos CSV brutos (sales, competidor, stores). Desenvolvi o script Python `ingestion_gcp.py` que autentica via Service Account e carrega esses dados para a camada `raw_data` no BigQuery. Escolhi o BigQuery por sua capacidade de processar Petabytes sem necessidade de indexação manual.

- **Transformações e Formato:**

Dentro do BigQuery, estruturei as tabelas mantendo a granularidade original. Minha principal transformação foi na criação da Tabela Mestra Analítica. Utilizei a técnica de ASOF JOIN para cruzar as vendas com os preços dos concorrentes. Decidi por isso, pois os dados de concorrentes não são diários; o ASOF JOIN me permitiu encontrar qual era o preço do concorrente "ativo" no exato momento de cada venda, evitando valores nulos que sujariam o modelo.

- **Decisões de Modelagem:**

1. **Granularidade de Transação:** Mantive os dados nível item-nota fiscal. Decidi não agregar por mês/semana porque a elasticidade de preço ocorre no dia-a-dia.
2. **Limpeza Rigorosa:** Identifiquei ruídos propositais na base (preços negativos). Implementei filtros no script `data_prep.py` para remover transações com `price <= 0` ou `cost <= 0`, evitando distorções na regressão logarítmica.

### Parte 2: Entendimento do Problema (RGM)

Enquadrei este desafio sob a ótica de **Eficiência Promocional**. O problema central não é "como vender mais a qualquer custo", mas "como investir margem (desconto) apenas onde ela retorna lucro".

- **Estrutura Lógica:** Diagnóstico (Elasticidade) → Simulação (Cenários de Desconto) → Decisão (Lucro Incremental Positivo).
- **KPIs Definidos:**
  - *Alvo (Maximizar):* Lucro Incremental (\$\$) e ROI Promocional (%).
  - *Guardrail:* Índice de Canibalização e Erosão de Preço (não aceitamos queimar margem para ganhar *market share* neste momento).

## Parte 3: Modelagem

Para simular os cenários, desenvolvi um modelo econométrico de **Regressão Log-Log (Log-Linear)** com a seguinte equação:

$$\ln(\text{Demanda}) = \alpha + \beta \cdot \ln(\text{Preço}) + \gamma \cdot \ln(\text{PreçoConcorrente})$$

Escolhi o Log-Log pela **interpretabilidade**, pois o coeficiente  $\beta$  fornece diretamente a Elasticidade-Preço, permitindo explicar facilmente para o Diretor Comercial o impacto de variações de preço no volume.

## Parte 4: Elasticidade & Concorrência

- **Estimativas de Elasticidade:** Ao rodar o modelo (script `eda_rgm.py`), encontrei um padrão claro de **Inelasticidade**. Para a maioria dos produtos (ex: P2, P5, P8), a elasticidade ficou entre **-0.5 e +0.2**. Isso indica que o consumidor reage pouco a variações de preço.
- **Impacto da Concorrência:** A análise da variável `competitor_price` (Elasticidade Cruzada) mostrou coeficientes próximos de zero ou negativos. Conclui-se que a demanda é guiada mais por fatores internos do que pelo preço da concorrência imediata.

## Parte 5: Decisão de Pricing

Baseado em 45 simulações (por Região e Produto), a recomendação técnica é **NÃO REALIZAR PROMOÇÕES AGRESSIVAS**.

- **Metodologia de Estimativa:**
  1. **Lucro Incremental:**  $(\text{Volume\_Novo} * \text{Margem\_Nova}) - (\text{Volume\_Base} * \text{Margem\_Base})$ .
  2. **Risco de Erosão:** Como a demanda é inelástica, qualquer desconto se torna "investimento a fundo perdido", erodindo valor da marca.

## Parte 7: Produção & Escala (MLOps)

Para implantação em produção no GCP, a arquitetura proposta é:

1. **Orquestração: Cloud Composer (Airflow)** rodando DAGs diárias.
  2. **Processamento: dbt** no BigQuery para transformar dados brutos na Tabela Mestra Analytic.
  3. **Inferência: Vertex AI** para gerenciar o ciclo de vida do modelo e servir previsões.
- **Monitoramento:** Foco em **Data Drift**, especificamente em Sazonalidade e Eventos (ex: Black Friday). Configurei alertas no Vertex AI Model Monitoring para desvios na distribuição de vendas.

Link: [Github do Case](#)