

FIAP  ON

Luis: RM565324

Adeilson: RM566282

DESAFIO 1

- **1º cenário**

Quando um cliente seleciona um produto, a plataforma de e-commerce exibe, adicionalmente, recomendações de outros itens, baseadas nas compras de quem comprou esse produto e em outras promoções correlatas. No contexto atual, esse cálculo está demorando muito tempo para ser feito utilizando estruturas relacionais, dado o volume de dados envolvidos.

Para esse primeiro cenário, a TI propôs o uso de um banco de dados NoSQL do tipo GRAFO.

- Justificativa:

A escolha de um banco de dados de Grafo para o motor de recomendações se baseia em sua capacidade ímpar de tratar os dados não como registros isolados, mas como uma rede de conexões vivas.

Esse tipo possui suporte full ACID (Atomicidade, Consistência, Isolamento e Durabilidade). Isso quer dizer que operações não vão atrapalhar uma as outras, facilitando consulta de múltiplos itens recomendados. Além disso, os nós estão conectados a outros nós através dos relacionamentos. E esses nós podem ter relacionamentos múltiplos, ou até recursivos. Ou seja, é muito mais flexível para recomendação dos itens.

Navegar por essa teia de relações para gerar uma recomendação é a especialidade nativa de um banco de grafos. Ele percorre essas conexões de forma fluida e instantânea. A abordagem relacional, por outro lado, tenta reconstruir essas conexões a cada consulta com JOINs pesados e complexos, o que é como tentar montar um quebra-cabeça de mil peças a cada vez que um cliente clica em um produto, explicando a lentidão atual.

- Exemplo Real:

O Pinterest utiliza bancos de dados de grafo para mapear as complexas relações entre usuários (Pinners), Pins e painéis (Boards). Isso permite que eles recomendem conteúdo novo e relevante em tempo real, entendendo quais Pins estão conectados a quais interesses e usuários.

- **2º cenário**

A definição da entrega de um produto em 24h depende da disponibilidade de estoque do centro de distribuição mais próximo do endereço de entrega. Se o cliente optar por essa entrega fast, é necessário realizar a reserva no centro de distribuição e, automaticamente, atualizar o estoque para atender a outros clientes. Nos testes preliminares com o uso do modelo relacional, o desempenho foi frustrante, influenciado principalmente pelo volume de dados e frequência de atualização.

Para esse segundo cenário, a TI propôs o uso de um banco de dados NoSQL do tipo COLUNAR.

- Justificativa:

A escolha de um banco de dados Colunar é estratégica para a gestão de estoque em tempo real, pois ele é otimizado para analisar e agregar dados em altíssima velocidade.

Nesse tipo, os dados são armazenados em uma família de colunas, o ideal para lidar com estoques. Além disso, podemos identificar diferentes versões do mesmo dado, o que facilita ao procurar pelo mesmo item em outras unidades mais próximas.

Essa arquitetura é brutalmente eficiente para operações de escrita e leitura analítica, que é exatamente o que acontece em um controle de estoque: milhares de atualizações (escritas) e verificações (leituras agregadas) por segundo. Nos testes com o modelo relacional, a frustração veio da necessidade de bloquear e ler linhas inteiras de dados para fazer uma simples atualização de quantidade, criando gargalos e lentidão. O banco colunar foi projetado para suportar esse volume massivo de transações sem travar.

- Exemplo Real:

O Spotify usa o Apache Cassandra (um banco de dados colunar) para gerenciar dados de playlists e históricos de reprodução dos usuários. Esse cenário exige uma taxa de escrita colossal e a capacidade de realizar leituras analíticas rápidas para alimentar seus próprios sistemas de recomendação e dashboards, um caso de uso análogo à necessidade de alta performance em escrita e leitura do controle de estoque.

- **3º cenário**

A tela de detalhes de um produto sempre recebe novas informações e, hoje em dia, possui informações que podem ser armazenadas juntamente com o produto, tais como: reviews do produto; suas versões; informações de entrega; imagens; recomendações; dicas, entre outras.

Para esse terceiro cenário, a TI propôs o uso de um banco de dados NoSQL do tipo DOCUMENTO.

- Justificativa:

Um banco de dados de Documento é a escolha ideal aqui porque trata cada produto como uma ficha catalográfica completa e autônoma, geralmente em formato JSON.

Essa flexibilidade permite que a plataforma evolua no ritmo do mercado, sem que a tecnologia seja um empecilho, permitindo enriquecer a página do produto de forma contínua e sem atritos.

- Exemplo Real:

A Adobe utiliza o MongoDB em sua plataforma de experiência criativa para gerenciar uma vasta gama de metadados e conteúdo gerados pelos usuários. Cada projeto ou ativo criativo é um "documento" com uma estrutura variada e complexa, similar à necessidade de armazenar informações heterogêneas sobre cada produto na plataforma da "Melhores Compras".

DESAFIO 2

- Comportamento de Compra (Outliers):

A análise identificou que vendas com 4 ou mais itens são atípicas (outliers). Estes casos sugerem um padrão de compra em atacado, distinto do consumidor final. Excluindo esses pontos, a variação normal nas vendas é mais previsível (desvio padrão de 0.78), representando o cliente comum.

```
# Analisando a 'Quantidade' e os Outliers

# Analisando outliers
Q1 = df['quantidade'].quantile(0.25) # O valor que deixa 25% das vendas para trás.
Q3 = df['quantidade'].quantile(0.75) # O valor que deixa 75% das vendas para trás.
IQR = Q3 - Q1

# Definindo um limite para o outlier
limite_superior = Q3 + 1.5 * IQR

outliers = df[df['quantidade'] > limite_superior]
print("Vendas consideradas 'fora da curva' (outliers):")
print(outliers.head())

# Calculando a variabilidade desconsiderando esses outliers
dados_sem_outliers = df[df['quantidade'] <= limite_superior]
desvio_padrao_normal = dados_sem_outliers['quantidade'].std()
print(f'O desvio padrão das vendas "normais" é: {desvio_padrao_normal}')

[7] ✓ 0.0s

-- Vendas consideradas 'fora da curva' (outliers):
cod_pedido  regiao_pais           produto  valor  quantidade \
3            4      Norte  Biscoito  True Champion 300g  19.0      4.0
5            6      Norte  Biscoito  True Champion 300g  19.0      4.0
9           10  Centro Oeste  Biscoito  True Champion 300g  26.0      4.0
10          11  Centro Oeste  Biscoito  True Champion 300g  28.0      4.0
15          16   Nordeste  Biscoito  True Champion 300g  22.0      4.0

  valor_total_bruto       data        estado  forma_pagto \

```

- Estratégia de Preços (Significância Estatística):

Não foi encontrada nenhuma diferença estatisticamente relevante no preço médio por item entre as diferentes regiões do país ou formas de pagamento. Isso indica uma forte consistência na política de preços e no mix de produtos da empresa em todos os segmentos analisados.

```
# Análise da Média de Preço

media_geral_preco = df['valor'].mean()
print(f"A média de preço de todos os produtos é: R$ {media_geral_preco:.2f}")

# Comparando a média de cada região com a média geral
for regiao in df['regiao_pais'].unique():
    precos_da_regiao = df[df['regiao_pais'] == regiao]['valor']

    # Verificando se as médias regionais são muito diferentes da geral
    t_stat, p_valor = stats.ttest_1samp(precos_da_regiao, media_geral_preco)

    if p_valor < 0.05:
        print(f"A região {regiao} TEM uma média de preço significativamente diferente.")
    else:
        print(f"A região {regiao} NÃO TEM uma média de preço significativamente diferente.")

[8] ✓ 0.1s
...
A média de preço de todos os produtos é: R$ 111.05
A região Norte NÃO TEM uma média de preço significativamente diferente.
A região Centro Oeste NÃO TEM uma média de preço significativamente diferente.
A região Nordeste NÃO TEM uma média de preço significativamente diferente.
A região Sudeste NÃO TEM uma média de preço significativamente diferente.
A região Sul NÃO TEM uma média de preço significativamente diferente.
```

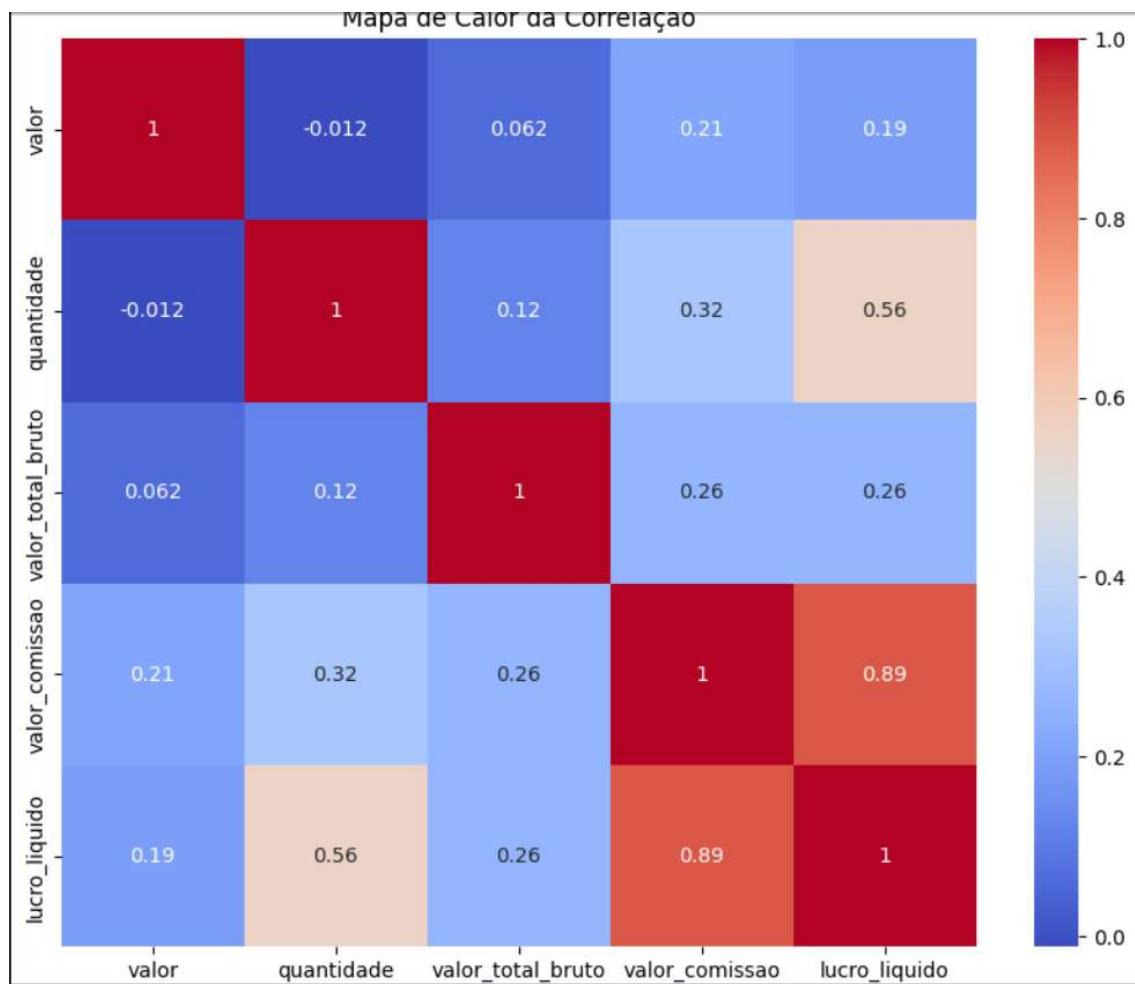
POTENCIALIZANDO O DESEMPENHO COM NOSQL

- Fatores de Lucratividade (Correlação):

A análise de correlação revelou dois pontos-chave para o lucro:

A comissão está fortemente ligada ao lucro líquido (correlação de 0.89), mostrando um sistema de incentivo alinhado à lucratividade.

A quantidade de itens vendidos tem uma correlação moderada a forte com o lucro líquido (0.56), sendo um fator mais impactante para o lucro do que o valor bruto total do pedido.



REFERÊNCIAS

“FIAP EAD - Login.” Fiap.com.br, 2025, on.fiap.com.br/local/salavirtual/conteudo-digital.php.

Foster, Brian. “Graph Service: Why Pinterest Modernized with Distributed SQL.” TiDB, 6 Dec. 2024, www.pingcap.com/blog/why-pinterest-modernized-graph-service-distributed-sql/.

Procter, Alexander. “How Spotify Built the Tech behind Streaming Millions of Songs | Okoone.” Okoone, 11 Apr. 2025, www.okoone.com/spark/technology-innovation/how-spotify-built-the-tech-behind-streaming-millions-of-songs/.

“Adobe Experience Manager with MongoDB | Adobe Experience Manager.” Adobe.com, 2025, experienceleague.adobe.com/en/docs/experience-manager-65/content/implementing/deploying/introduction/aem-with-mongodb.