

BeerFactory

Delivery Fast Levantamento de Requisitos

Versão:1.0

Data:26/06/2016

MIT

Histórico de revisões do modelo

Versão (XX.YY)	Data (DD/MMM/YYYY)	Autor	Descrição	Localização
1.0	26/06/2020	Breno Fernandes e Leila Almeida	Primeira versão do projeto	Cada um na sua casa/SP

Aprovadores

Nome	Função
Professora Tassiana	Gerente de Projeto
Breno Fernandes e Leila Almeida	Engenheiro dados
Breno Fernandes e Leila Almeida	Analista de dados
Breno Fernandes e Leila Almeida	Arquitetura de dados

Índice

Visão Geral da Solução	3
Desenho Arquitetural	4
Requisitos Funcionais	4
2.2.1. Objetivos	5
2.2.2. KPI	5
2.2.3. Requisitos	5
Detalhamento da Solução	7
Modelo Relacional – Ambiente Transacional	8
Modelo Dimensional	9
Modelo Flatten	9
Modelo em Grafos	9
Modelo em Documento	13
Visualização dos Indicadores	14

1. Visão Geral da Solução

1.1. Desenho Arquitetural

O primeiro passo da construção da nossa solução foi a produção do nosso modelo de arquitetura de solução e tecnologia. Esse modelo irá servir como base para que seja construída a plataforma de dados de forma end-to-end, ou seja, descrevendo as tecnologias utilizadas na plataforma desde o processo de captura da informação na distribuidora, até o processo de visualização e análise de dados feito pelo time de analistas da BeerFactory. A imagem a seguir apresenta uma visão global das tecnologias e do fluxo do processo.

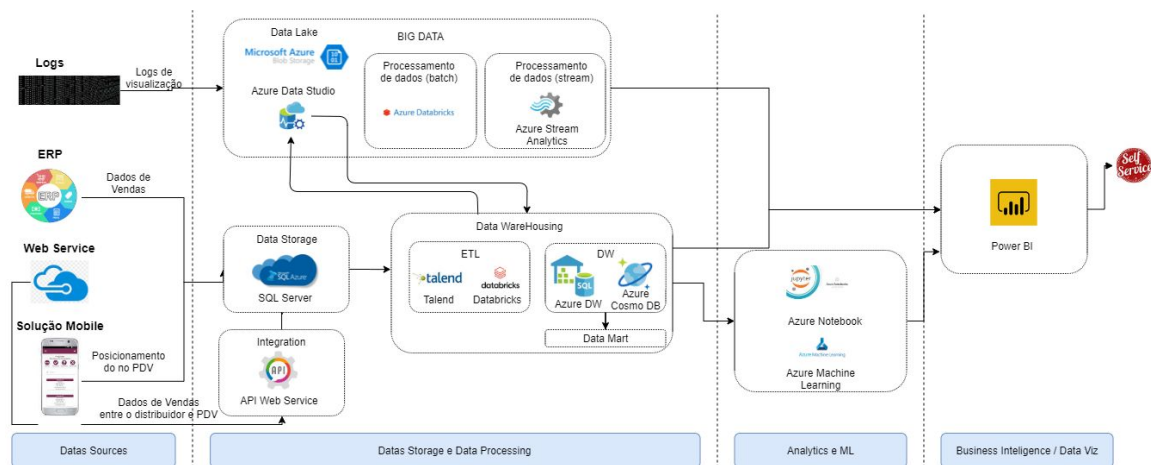


Figura 1: Arquitetura da plataforma de dados

Importante notar que dado o cenário de captura de dados já se encontrar em nuvem, foi adotado a utilização de tecnologias do ambiente da Cloud **Azure**, com o intuito de maximizar os tempos de análise, engenharia e integração.

2. Requisitos Funcionais

2.1. Definição do Produto

O produto principal da empresa é a distribuição de bebidas alcoólicas para vendedores regionais.

2.2. Levantamento das Necessidades

O principal objetivo do projeto é maximizar o uso de recurso no processo de distribuição e venda de produtos da empresa. Para isso, foi adotado como estratégia de negócio a construção de uma plataforma de dados para facilitar a identificação de pontos de melhorias na distribuição de produto. Assim, miramos os dois principais objetivos como:

2.2.1. Objetivos

- Aumentar o número de vendas através do trade marketing
- Aumentar fidelização do consumidor ao produto

2.2.2. KPI

- Números de vendas
- Vendas de produtos x quantidade distribuídas

2.2.3. Requisitos

Os principais requisitos do projeto são:

- Disponibilização de dados para geração de relatórios de vendas.
- Capacidade de recebimento e processamento de logs dos PDVs via plataforma de data streaming.
- Existência de DW com as informações normalizadas e centralizadas.
- Ferramentas para execução de advanced analytics em Python.

Requisitos secundários;

- Modelo de machine learning para melhorar a distribuição do produto.
- Otimizar campanhas de promoção de produtos e engajamento com as distribuidoras.

2.3. Estrutura de Distribuição da Empresa

A empresa possui diversas divisões internas com o intuito de especializar as funções exercidas pelos colaboradores. Além disso, temos também a separação comercial para atendimento a clientes no varejo e no atacado. O diagrama a seguir apresenta uma visão resumida da nossa hierarquia organizacional.

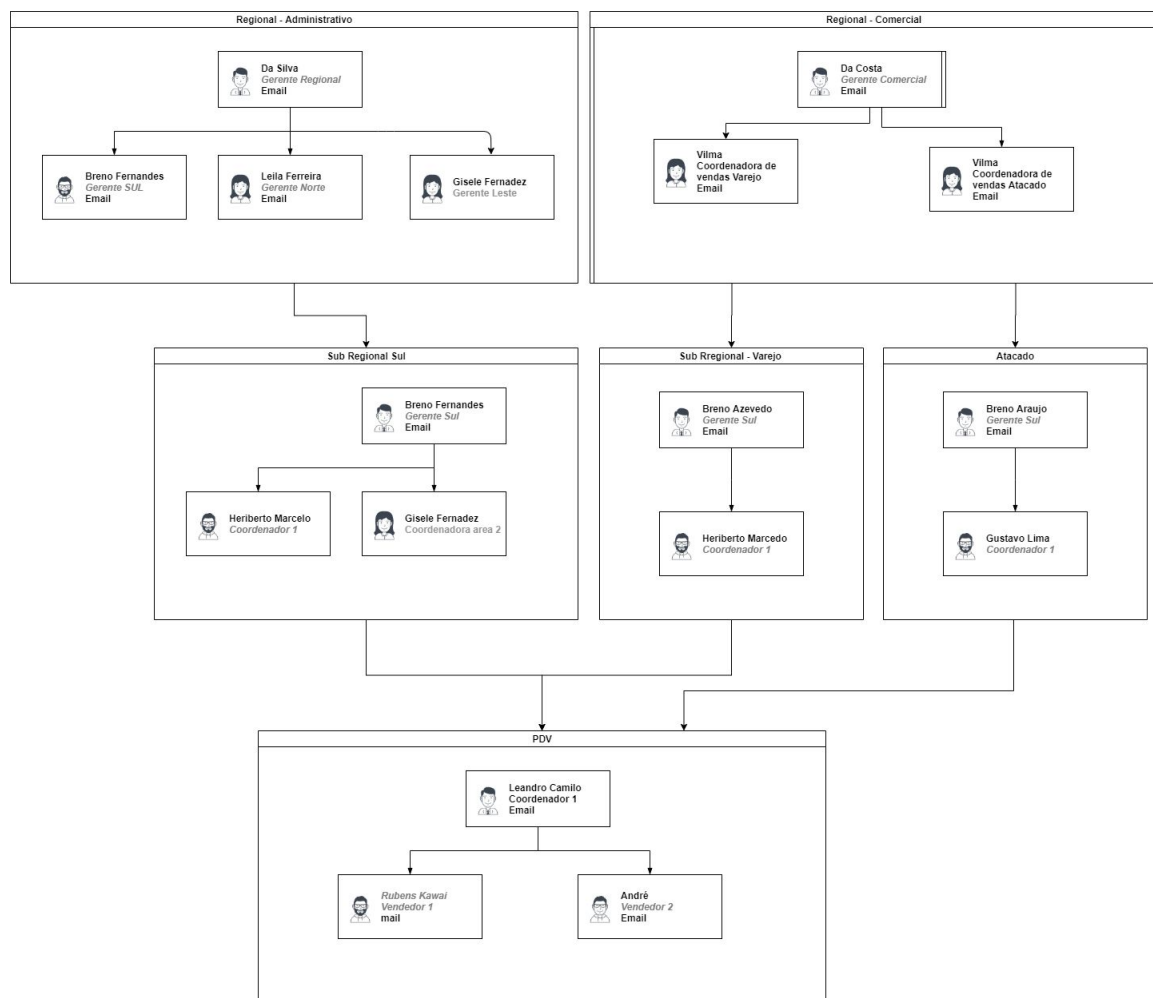


Figura 2: Organograma funcional

2.4. BPM

[illegible]

Figura 3: BPM para distribuição tradicional do produto.

[Link para o arquivo](#)

Modelo	Flatten					
Grao	Visualização / Produto					
Metrica	Tipo de Agregação	Tipo de Origem				
Visualização dos produtos	Count group by	Quente / Semi estruturada				
Uso de sistema operacional	Média	Quente / Semi estruturada				
Preço médio por departamento	Média	Quente / Semi estruturada				
Visualização por Região(IP)	Count	Quente / Semi estruturada				
Dimensão	Tipo de Dimensão					
Produto	Hierarquica					
Data	Tempo					
Preço	Hierarquica					
Departamento	Hierarquica					
IP de Origem	Geográfica					
Matriz Metrica/ Dimensão	Produto	Data	Preço	SO	Departamento	IP de Origem
Visualização dos produtos	X	X		X	X	x
Uso de sistema operacional	X	X		X		
Preço médio por departamento / Data	x	X	x	X	x	X
Visualização por região(IP)	X	X	X	X	X	x

Figura 4: BPM para distribuição e-commerce do produto.

[Link para o arquivo](#)

3. Detalhamento da Solução

3.1. Modelo Relacional – Ambiente Transacional

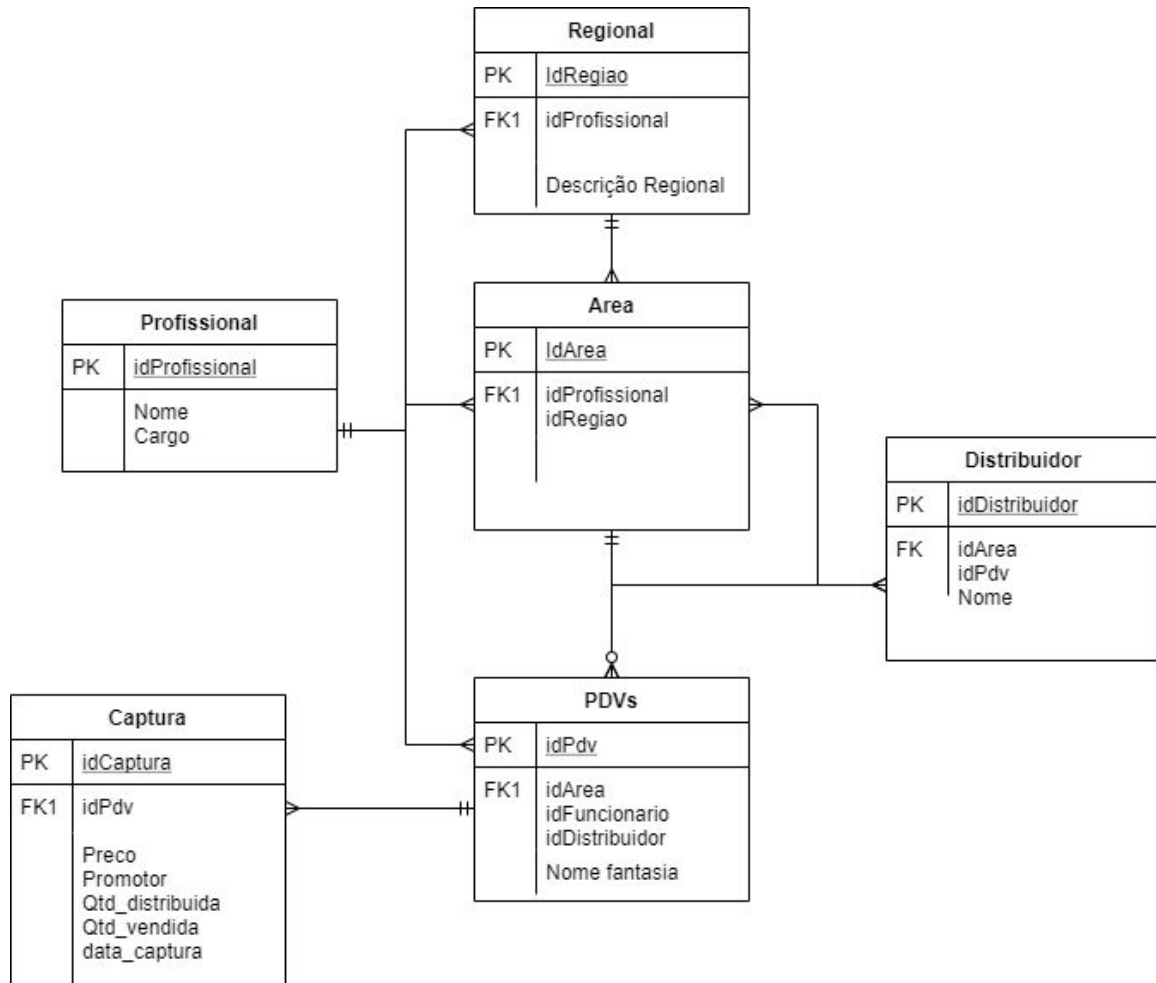


Figura 5: Modelo relacional para venda físicas

[Link para o arquivo](#)

3.2. Modelo Dimensional

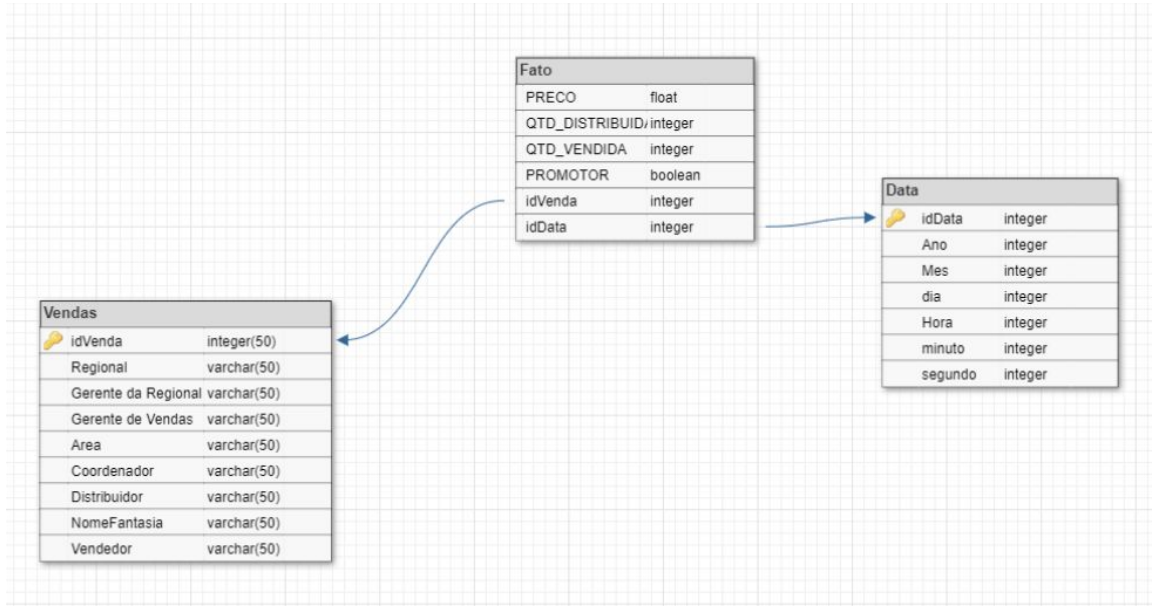


Figura 6: Modelo dimensional para venda física

[Link para o arquivo](#)

3.3. Modelo Flatten

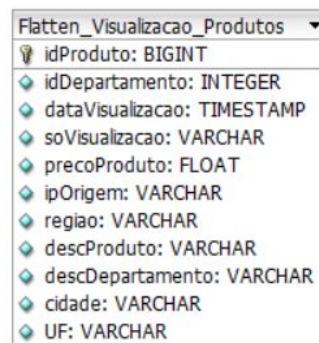


Figura 7: Modelo flatten para ecommerce

[Link para o arquivo](#)

3.4. Modelo em Grafos

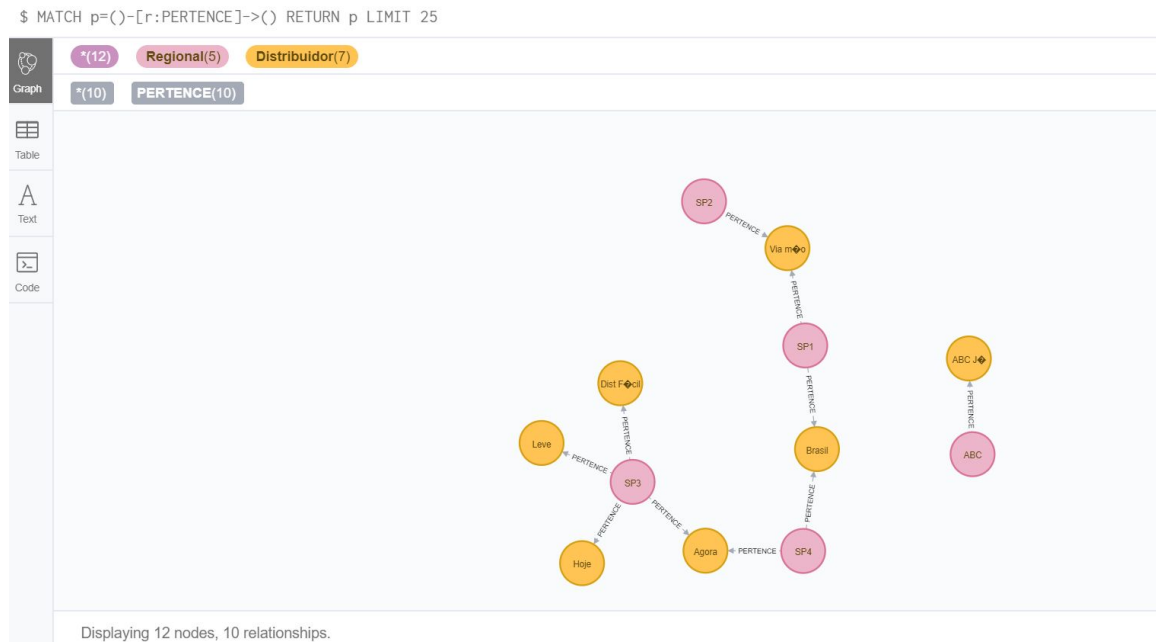


Figura 8: Graph de distribuição de regional e distribuidor

```

LOAD CSV WITH HEADERS FROM
'https://ucbd0ef700fcb64d97735574db97.dl.dropboxusercontent.com/cd/0/get/A6a19xZbha9F14dfL27M14awF_wX3LLt68Xb
9X4xYIY1588-SLaT5xvKvDjli3gbRclpLes95TCR8qCUGpMD6FWvAwQmm5WP20kpXXzDehwdOzA51MeF2hXpbMkD
X_c9L1A/file#' AS line
CREATE(:Regional{regiao:line.Regiao})
CREATE(:Distribuidor{distribuidor:line.Distribuidor})

MATCH (p:Distribuidor)
WITH p
ORDER BY p.distribuidor, size((p)--()) DESC
WITH p.distribuidor as distribuidor, collect(p) AS nodes
WHERE size(nodes) > 1
UNWIND nodes[1..] AS n
DETACH DELETE n

MATCH (p:Regional)
WITH p
ORDER BY p.regiao, size((p)--()) DESC
WITH p.regiao as regiao, collect(p) AS nodes
WHERE size(nodes) > 1
UNWIND nodes[1..] AS n
DETACH DELETE n

MATCH (p:Regional),(c:Distribuidor)
WHERE p.regiao = 'SP1' AND c.distribuidor = 'Brasil'
CREATE (p)-[r:PERTENCE]->(c)

MATCH (p:Regional),(c:Distribuidor)
WHERE p.regiao = 'SP1' AND c.distribuidor = 'Via m♦o'
CREATE (p)-[r:PERTENCE]->(c)

MATCH (p:Regional),(c:Distribuidor)

```

```

WHERE p.regiao = 'SP2' AND c.distribuidor = 'Via m♦o'
CREATE (p)-[r:PERTENCE]->(c)

MATCH (p:Regional),(c:Distribuidor)
WHERE p.regiao = 'SP3' AND c.distribuidor = 'Hoje'
CREATE (p)-[r:PERTENCE]->(c)

MATCH (p:Regional),(c:Distribuidor)
WHERE p.regiao = 'SP3' AND c.distribuidor = 'Leve'
CREATE (p)-[r:PERTENCE]->(c)

MATCH (p:Regional),(c:Distribuidor)
WHERE p.regiao = 'SP3' AND c.distribuidor = 'Agora'
CREATE (p)-[r:PERTENCE]->(c)

MATCH (p:Regional),(c:Distribuidor)
WHERE p.regiao = 'SP3' AND c.distribuidor = 'Dist F♦cil'
CREATE (p)-[r:PERTENCE]->(c)


MATCH (p:Regional),(c:Distribuidor)
WHERE p.regiao = 'SP4' AND c.distribuidor = 'Brasil'
CREATE (p)-[r:PERTENCE]->(c)

MATCH (p:Regional),(c:Distribuidor)
WHERE p.regiao = 'SP4' AND c.distribuidor = 'Agora'
CREATE (p)-[r:PERTENCE]->(c)

MATCH (p:Regional),(c:Distribuidor)
WHERE p.regiao = 'ABC' AND c.distribuidor = 'ABC J♦'
CREATE (p)-[r:PERTENCE]->(c)

```

Tabela 1: Código para geração do grafo



The screenshot shows a Neo4j Cypher query editor. The query is: `$ MATCH (me)-[:PERTENCE*1..5]-(remote_friend:Regional) WHERE me.distrib...`. The results are displayed in a table view with the title `remote_friend.regiao`. The table contains three rows with the values `"SP4"`, `"SP1"`, and `"SP3"`.

remote_friend.regiao
"SP4"
"SP1"
"SP3"

Figura 9: Possibilidade de atendimento da distribuidora de atender regionais em até 5 saltos

```

MATCH (me)-[:PERTENCE*1..5]-(remote_friend:Regional)
WHERE me.distribuidor = 'Leve'
RETURN remote_friend.regiao

```

Tabela 2: Código para busca da primeira consulta

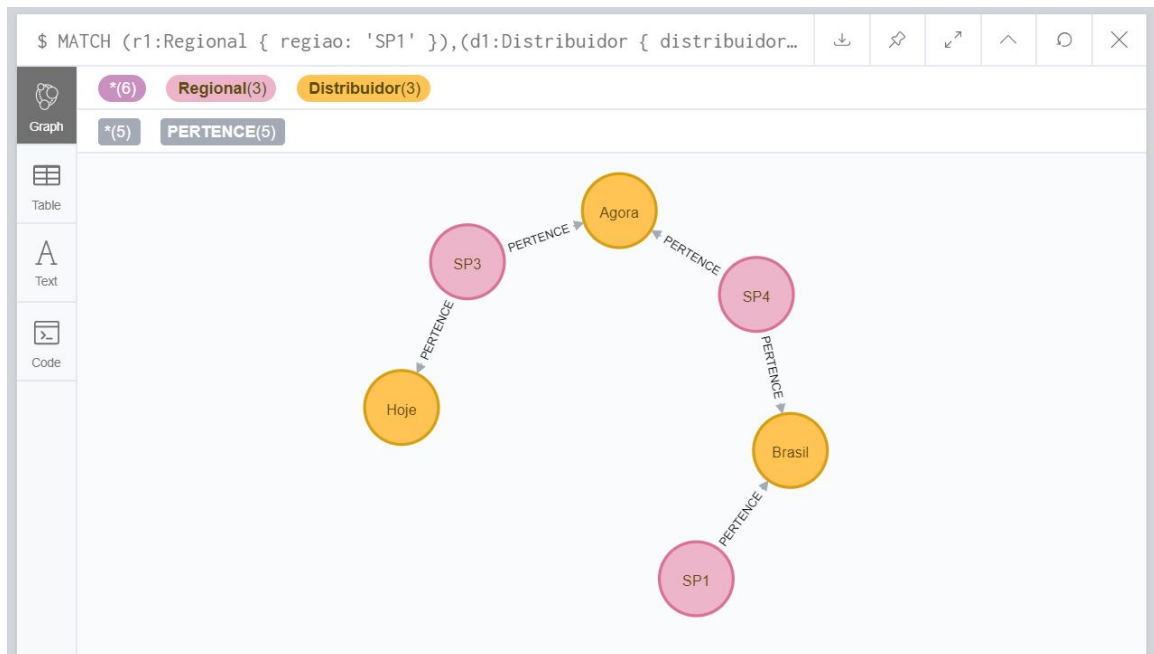


Figura 10: Menor caminho entre a região SP1 e distribuidora Hoje MAX 10 Saltos

```
MATCH (r1:Regional { regioao: 'SP1' }),(d1:Distribuidor { distribuidor:'Hoje' }),
p = shortestPath((r1)-[*..10]-(d1))
RETURN p
```

3.5. Modelo em Documento

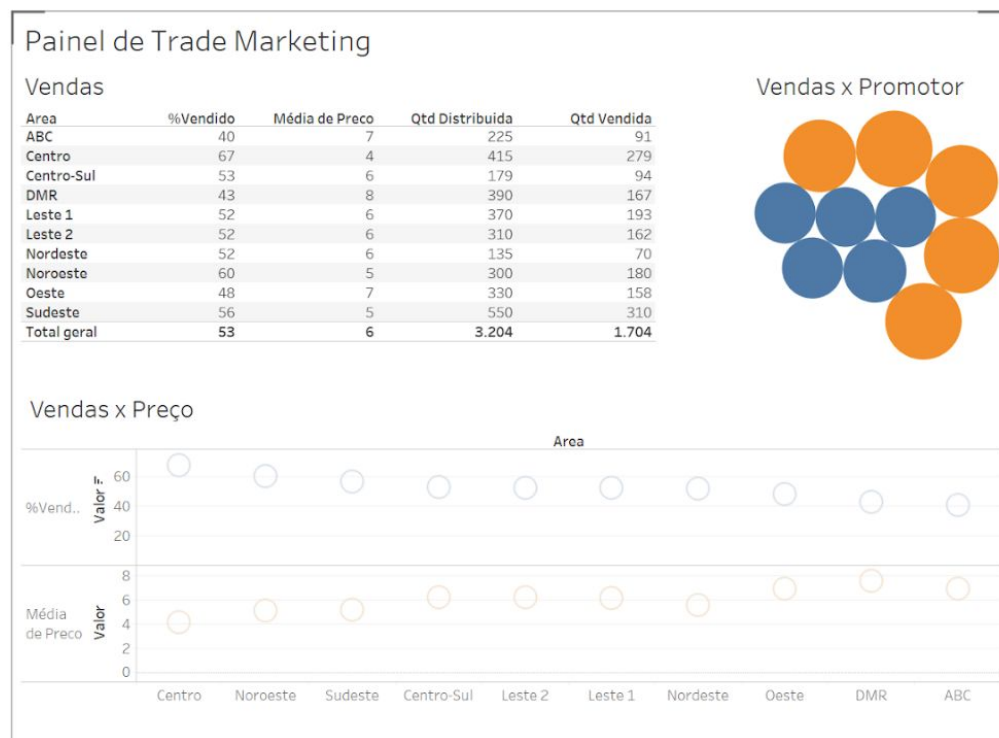


Figura 11: Painel em documento para vendas físicas
[Link para o arquivo](#)

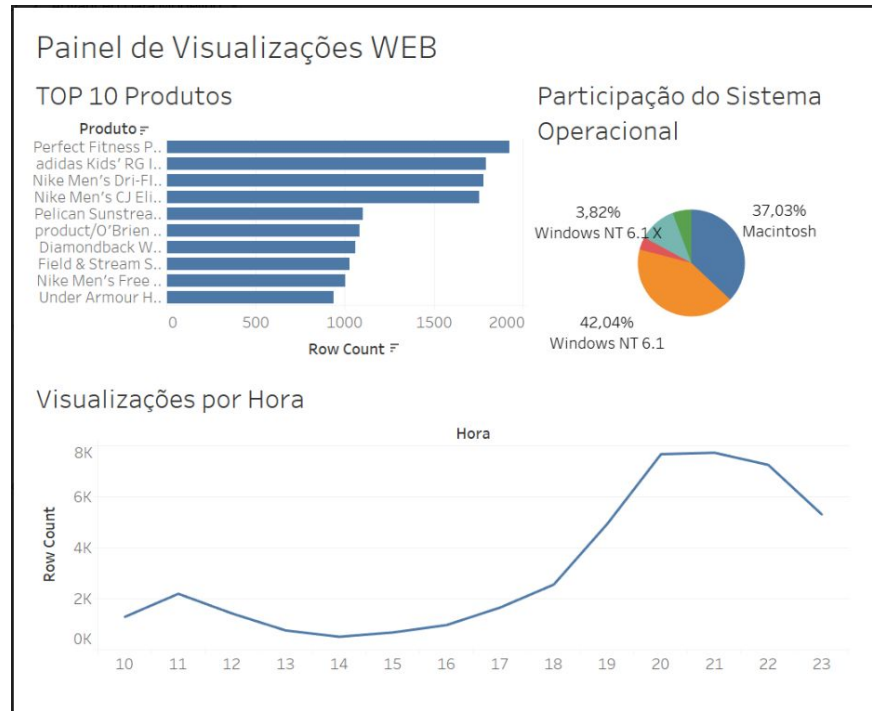


Figura 12: Painel em documento para e-commerce
[Link para o arquivo](#)

3.6. Visualização dos Indicadores

Para realizar a visualização dos indicadores o primeiro passo é identificar como as variáveis se correlacionam entre si, para isso, utilizamos a biblioteca pandas do Python via Google Colab para obtenção das métricas.



Figura 13: Métricas de correlação para a base a importada

Os valores obtidos mostram que temos uma alta correlação positiva entre as variáveis QTD_DISTRIBUIDA E QTD_VENDIDA. O que faz sentido, visto que, em cenários que temos uma distribuição maior de produtos, também teremos uma vazão maior de saída, e consequentemente, um retorno financeiro maior para o produto.

```
[12] scatter_plot_2 = plt.scatter(df['QTD_DISTRIBUIDA'], df['QTD_VENDIDA'], alpha=0.5, c=df['QTD_VENDIDA'])
```

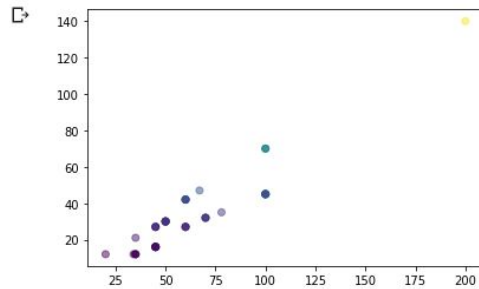


Figura 14 : Gráfico de dispersão para as variáveis QTD_DISTRIBUIDA E QTD_VENDIDA

O nosso segundo cenário de correção ocorre entre às variáveis PROMOTOR e PRECO, porém, diferente do que acontece no cenário anterior, teremos uma correção negativa para esse caso. Esse cenário acontece, pois em cenário que temos a presença de PROMOTOR, temos uma redução no valor. No entanto, na lógica essa medida não se justifica visto que a presença de um promotor não deveria impactar a questão do preço. Logo, se faz necessário uma avaliação colocando outras variáveis no processo.

```
[14] scatter_plot_3 = plt.scatter(df['PROMOTOR'], df['PRECO'], alpha=0.5, c=df['PROMOTOR'])
```

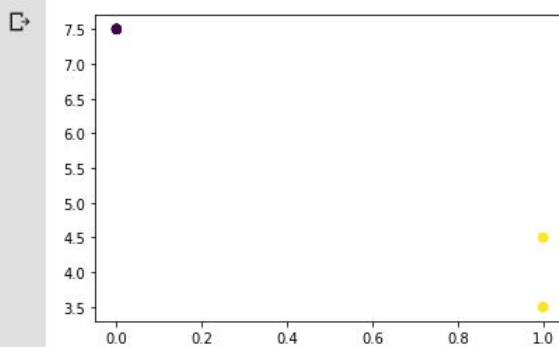


Figura 15: Correlação entre as variáveis PROMOTOR e PRECO