

Relatório: Análise de Vendas



Gonçalo Veríssimo

02/04/2025



Índice

Introdução	3
Desenvolvimento.....	4
1. Criação da Base de Dados no MySQL Workbench.....	4
2. Inserção de dados fictícios	5
3. Consultas SQL para obter informações relevantes.....	6
4. Importação dos Dados para o Power BI.....	7
Conclusão.....	10

Introdução

Este relatório apresenta a criação de um dashboard interativo no Power BI para monitorizar o desempenho das vendas, analisar o faturamento e prever possíveis ruturas de stock. Para alcançar este objetivo, foi criada uma base de dados no MySQL Workbench com dados fictícios, seguida da execução de diversas consultas SQL para obter insights relevantes. Posteriormente, os dados foram importados para o Power BI, onde foram criados gráficos e tabelas dinâmicas para facilitar a análise visual.

A análise inclui métricas como:

- Evolução das vendas ao longo do tempo;
- Identificação dos produtos mais vendidos;
- Percentagem de receita por categoria;
- Clientes com maior volume de compras;
- Produtos com baixo stock e necessidade de reposição.

Desenvolvimento

1. Criação da Base de Dados no MySQL Workbench

A primeira etapa consistiu na criação da base de dados "analise_vendas", incluindo tabelas para armazenar informações sobre produtos, clientes, pedidos e itens de pedido.

As tabelas foram definidas com as seguintes colunas principais:

- Produtos: Contém informações sobre os produtos disponíveis, incluindo nome, categoria, preço e stock atual;
- Clientes: Regista os clientes e as suas respetivas localizações;
- Pedidos: Armazena as transações realizadas, associando cada pedido a um cliente;
- Itens_Pedido: Detalha os produtos comprados em cada pedido, com o respetivo valor unitário.

```

1 • CREATE TABLE Produtos (
2     id_produto INT AUTO_INCREMENT PRIMARY KEY,
3     nome VARCHAR(100) NOT NULL,
4     categoria VARCHAR(50),
5     preco DECIMAL(10,2) NOT NULL,
6     stock_atual INT NOT NULL
7 );
8
9
10 • CREATE TABLE Clientes (
11     id_cliente INT AUTO_INCREMENT PRIMARY KEY,
12     nome VARCHAR(100) NOT NULL,
13     cidade VARCHAR(100),
14     estado VARCHAR(50)
15 );
16
17 • CREATE TABLE Pedidos (
18     id_pedido INT AUTO_INCREMENT PRIMARY KEY,
19     id_cliente INT,
20     data_pedido DATE NOT NULL,
21     valor_total DECIMAL(10,2),
22     FOREIGN KEY (id_cliente) REFERENCES Clientes(id_cliente)
23 );
24
25
26 • CREATE TABLE Itens_Pedido (
27     id_item INT AUTO_INCREMENT PRIMARY KEY,
28     id_pedido INT,
29     id_produto INT,
30     valor_unitario DECIMAL(10,2) NOT NULL,
31     FOREIGN KEY (id_pedido) REFERENCES Pedidos(id_pedido),
32     FOREIGN KEY (id_produto) REFERENCES Produtos(id_produto)
33 );

```

2. Inserção de dados fictícios

Após a criação das tabelas, os dados fictícios foram inseridos utilizando o comando **INSERT INTO**.

```

36 • INSERT INTO Produtos (nome, categoria, preco, stock_atual) VALUES
37   ('Teclado Mecânico RGB', 'Periféricos', 89.99, 40),
38   ('Rato Sem Fios', 'Periféricos', 49.99, 30),
39   ('Monitor LED 27"', 'Monitores', 229.99, 15),
40   ('Portátil Dell XPS 13', 'Computadores', 1299.99, 8),
41   ('Cadeira Gaming', 'Mobiliário', 299.99, 12),
42   ('Disco SSD 512GB', 'Armazenamento', 99.99, 25),
43   ('Impressora Multifuncional', 'Periféricos', 149.99, 10),
44   ('Coluna Bluetooth JBL', 'Áudio', 79.99, 20),
45   ('Smartwatch Samsung', 'Gadgets', 199.99, 18),
46   ('Tablet Lenovo 10"', 'Tablets', 249.99, 14);
47
48
49 • INSERT INTO Clientes (nome, cidade, estado) VALUES
50   ('João Silva', 'Lisboa', 'Lisboa'),
51   ('Diogo Cruz', 'Porto', 'Porto'),
52   ('Carlos Santos', 'Braga', 'Braga'),
53   ('Ana Costa', 'Coimbra', 'Coimbra'),
54   ('Rui Pereira', 'Faro', 'Faro'),
55   ('Sofia Almeida', 'Aveiro', 'Aveiro'),
56   ('Pedro Matos', 'Guimarães', 'Braga'),
57   ('Gonçalo Veríssimo', 'Setúbal', 'Setúbal'),
58   ('Tiago Moreira', 'Évora', 'Évora'),
59   ('Patrícia Rocha', 'Viseu', 'Viseu');
60
61 • INSERT INTO Pedidos (id_cliente, data_pedido, valor_total) VALUES
62   (1, '2025-03-10', 139.98),
63   (2, '2025-03-11', 299.99),
64   (3, '2025-03-12', 229.99),
65   (4, '2025-03-13', 1299.99),
66   (5, '2025-03-14', 99.99),
67   (6, '2025-03-15', 49.99),
68   (7, '2025-03-16', 199.99),
69   (8, '2025-03-17', 249.99),
70   (9, '2025-03-18', 79.99),
71   (10, '2025-03-19', 149.99);
72
73
74 • INSERT INTO Itens_Pedido (id_pedido, id_produto, valor_unitario) VALUES
75   (1, 1, 89.99),
76   (1, 2, 49.99),
77   (2, 5, 299.99),
78   (3, 3, 229.99),
79   (4, 4, 1299.99),
80   (5, 6, 99.99),
81   (6, 2, 49.99),
82   (7, 9, 199.99),
83   (8, 10, 249.99),
84   (9, 8, 79.99),
85   (10, 7, 149.99);

```

3. Consultas SQL para obter informações relevantes

```

114 -- Valor Total das Vendas em Euros (€) por Dia
115 • SELECT data_pedido, SUM(valor_total) AS total_vendas
116 FROM Pedidos
117 GROUP BY data_pedido
118 ORDER BY data_pedido;
119
120 -- Produtos Mais Vendidos
121 • SELECT p.nome, SUM(ip.valor_unitario) AS soma_valor_unitario
122 FROM Itens_Pedido ip
123 JOIN Produtos p ON ip.id_produto = p.id_produto
124 GROUP BY p.nome;
125
126 -- Percentagem do Valor Total em Euros (€) por Categoria
127 • SELECT
128     p.categoria,
129     (SUM(i.valor_unitario) / (SELECT SUM(valor_unitario) FROM itens_pedido)) * 100 AS percentagem_receita
130 FROM itens_pedido i
131 JOIN produtos p ON i.id_produto = p.id_produto
132 GROUP BY p.categoria
133 ORDER BY percentagem_receita DESC;
134
135 -- Clientes que Mais Compraram
136 • SELECT C.nome, SUM(P.valor_total) AS total_gasto
137 FROM Pedidos P
138 JOIN Clientes C ON P.id_cliente = C.id_cliente
139 GROUP BY C.nome
140 ORDER BY total_gasto DESC
141 LIMIT 5;
142
143 -- Produtos com Baixo Stock (<20 unidades)
144 • SELECT nome, stock_atual
145 FROM Produtos
146 WHERE stock_atual < 20
147 ORDER BY stock_atual ASC;

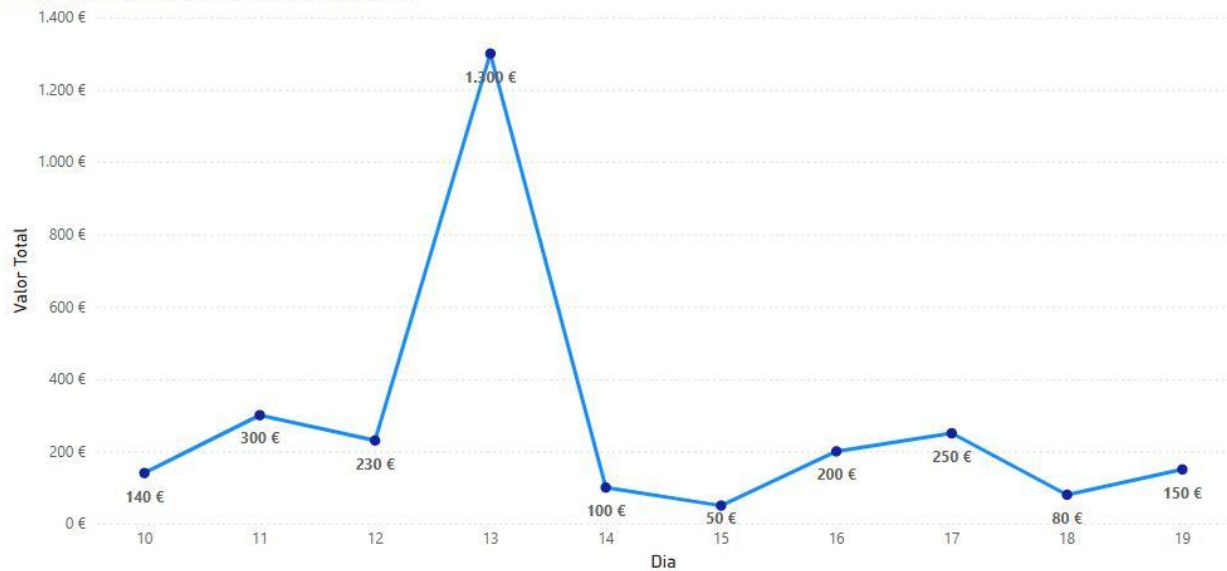
```

4. Importação dos Dados para o Power BI

Com as consultas SQL finalizadas, os dados foram importados para o Power BI utilizando o MySQL Connector. Após a importação, foram criadas visualizações interativas, nomeadamente:

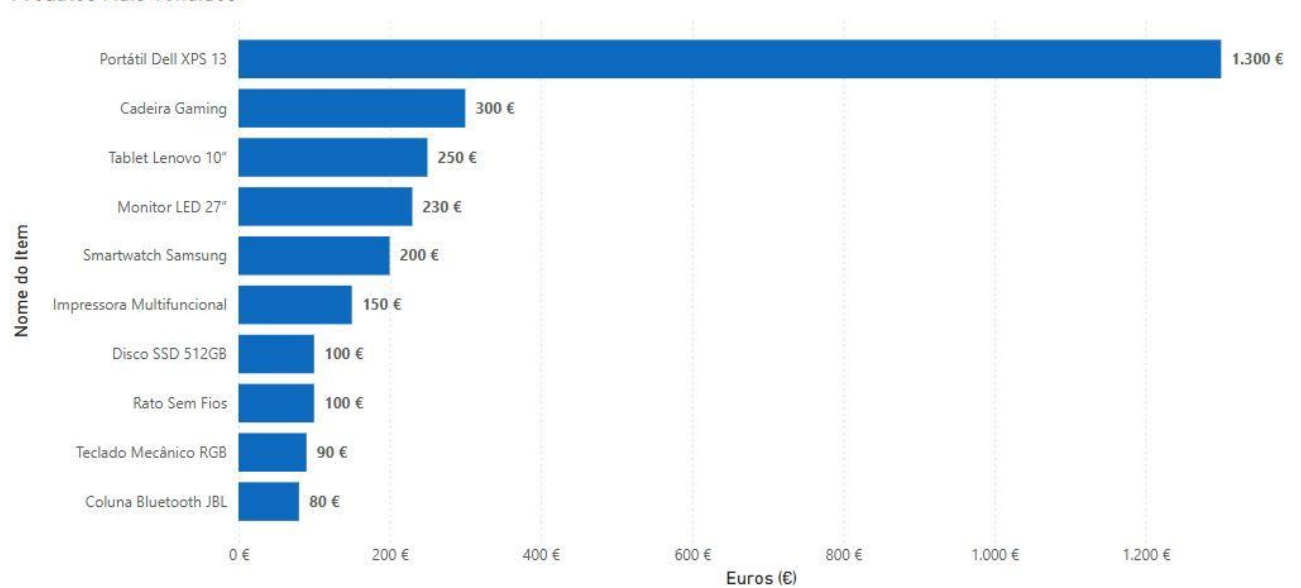
- Gráfico de linhas para a evolução das vendas ao longo do tempo:

Valor Total das Vendas em Euros (€) por Dia



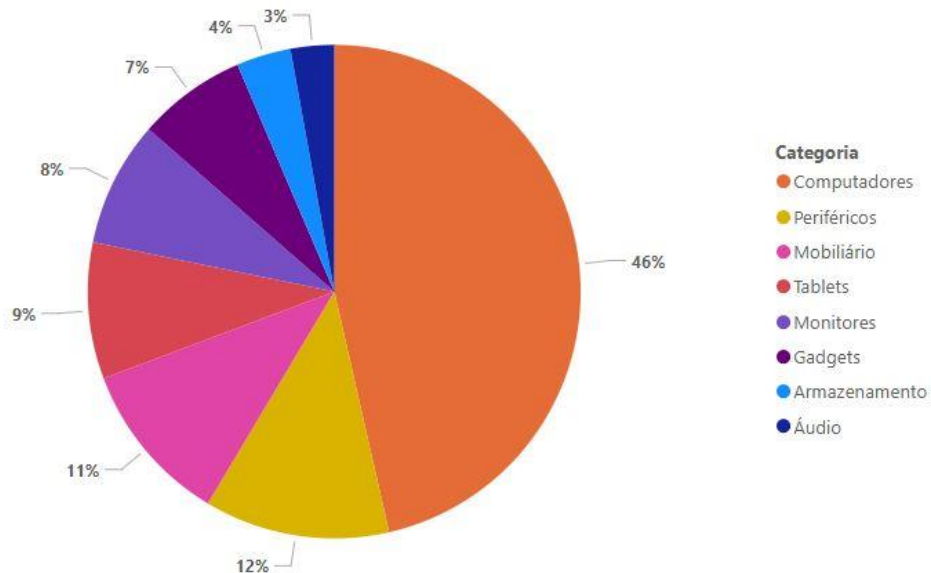
- Gráfico de barras empilhadas para os produtos mais vendidos:

Produtos Mais Vendidos



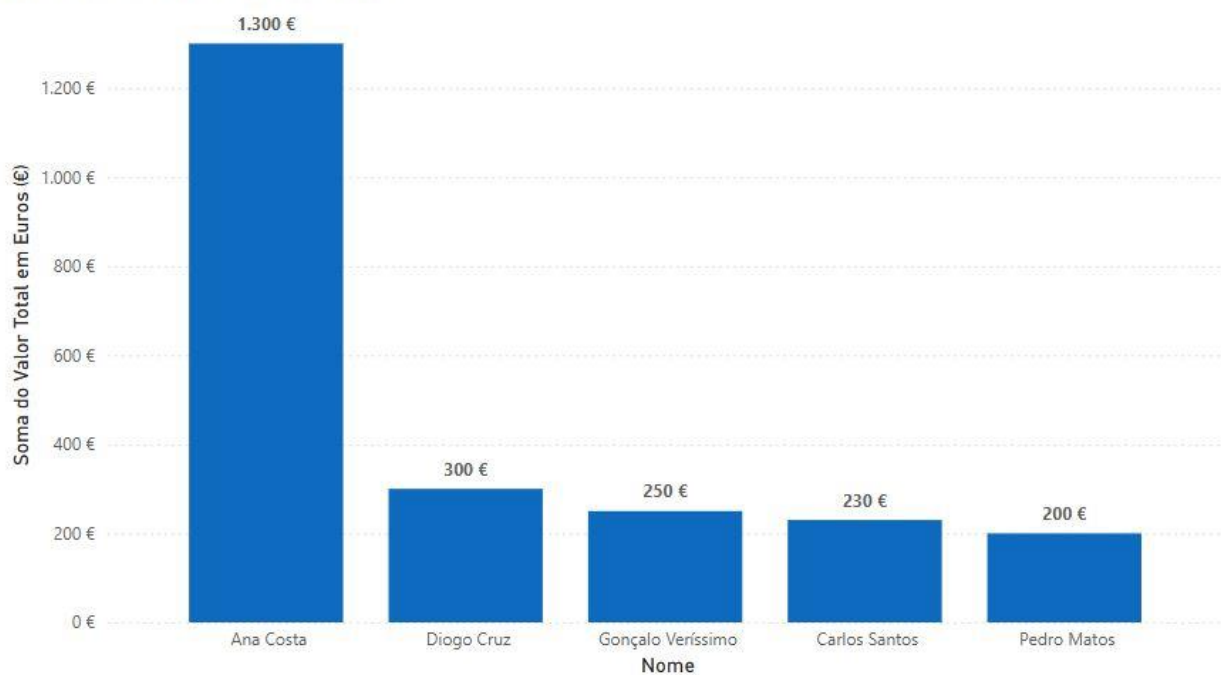
- Gráfico de pizza para a percentagem de receita por categoria:

Percentagem do Valor Total em Euros (€) por Categoria



- Gráfico de colunas empilhadas os clientes que mais compraram (Top 5 Clientes):

Top 5 Clientes que mais Compraram



- Tabela para os produtos com baixo stock (<20 produtos):

Produtos com Baixo Stock (<20)

nome	stock_atual
Portátil Dell XPS 13	8
Impressora Multifuncional	10
Cadeira Gaming	12
Tablet Lenovo 10"	14
Monitor LED 27"	15
Smartwatch Samsung	18

Conclusão

O presente trabalho demonstrou como é possível estruturar e analisar dados de vendas utilizando MySQL e Power BI. A partir das consultas SQL, foi possível extrair informações relevantes para a gestão do negócio, como a tendência de vendas, a identificação de produtos mais vendidos e a previsão de rupturas de stock.

A visualização dos dados no Power BI tornou a análise mais acessível e interativa, permitindo que os insights fossem interpretados de forma rápida e eficiente.

Este estudo reforça a importância da análise de dados na tomada de decisões empresariais, demonstrando o impacto que ferramentas como SQL e Power BI podem ter na otimização da gestão de vendas.

