

**INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA DE  
SÃO PAULO**

**ANDRE LUIZ LIMA RODRIGUES**

**SISTEMA DE GERENCIAMENTO DE LOJAS DE MATERIAL  
DE CONTRUÇÃO**

**INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA DE  
SÃO PAULO**

**ANDRE LUIZ LIMA RODRIGUES**

Entrega Parcial de trabalho final  
apresentado ao Instituto Federal  
de São Paulo (IFSP), em cumpri-  
mento a exigência da disciplina  
de Banco de Dados 1, do curso  
de Análise e Desenvolvimento de  
Sistemas.

**PROFESSOR: Paulo Giovani  
de Faria Zeferine.**

**SISTEMA DE GERENCIAMENTO DE LOJAS DE MATERIAL  
DE CONTRUÇÃO**

**CAMPOS DO JORDÃO  
ANO**

## RESUMO

O presente trabalho aborda o desenvolvimento de um sistema inclusivo e dinâmico destinado a atuar em uma loja de vendas peças de moto. Com o aumento da demanda por esse serviço, o estabelecimento identificou a necessidade de um sistema que abrange de maneira eficiente todos os processos, a implementação deste sistema visa melhorar significativamente os benefícios, a precisão e a segurança dos procedimentos realizados. Para atingir esse objetivo, além do desenvolvimento propriamente dito, inclui-se o levantamento de requisitos e modelagem de dados. O resultado é um sistema que atende às necessidades dos clientes, melhorando a eficiência das lojas online de peças de moto e agilizando o trabalho da equipe de desenvolvimento.

**Palavras-Chave:** Banco de Dados; Sistema; Loja de Peças de Moto; Desenvolvimento; Gerenciamento; Serviços;

## ABSTRACT

The present work addresses the development of an inclusive and dynamic system aimed at operating in a motorcycle parts sales store. With the increasing demand for this service, the establishment identified the need for a system that efficiently encompasses all processes. The implementation of this system aims to significantly improve the benefits, accuracy, and security of the procedures performed. To achieve this goal, in addition to the development itself, requirements gathering, and data modeling are included. The result is a system that meets the needs of customers, improving the efficiency of online motorcycle parts stores and streamlining the work of the development team.

**Keywords:** Database; System; Motorcycle Parts Store; Development; Management; Services.

## LISTA DE ILUSTRAÇÕES

<b>FIGURA 1</b> – Modelo relacional do sistema da loja de peças (2024)	16
<b>FIGURA 2</b> – Modelo lógico do sistema da loja de peças (2024)	16
<b>FIGURA 3</b> – Resultado da consulta 1	16
<b>FIGURA 4</b> – Resultado da consulta 2	16
<b>FIGURA 5</b> – Resultado da consulta 3	16
<b>FIGURA 6</b> – Resultado da consulta 4	16
<b>FIGURA 7</b> – Resultado da consulta 5	16
<b>FIGURA 8</b> – Resultado da consulta 6	16
<b>FIGURA 9</b> – Resultado da consulta 7	16
<b>FIGURA 10</b> – Resultado da consulta 8	16
<b>FIGURA 11</b> – Resultado da consulta 9	16
<b>FIGURA 12</b> – Resultado da consulta 10	16
<b>FIGURA 13</b> – Resultado da consulta 11	16
<b>FIGURA 14</b> – Resultado da consulta 12	16
<b>FIGURA 15</b> – Resultado da consulta 13	16
<b>FIGURA 16</b> – Resultado da consulta 14	16
<b>FIGURA 17</b> – Resultado da consulta 15	16
<b>FIGURA 18</b> – Resultado da consulta 16	16
<b>FIGURA 19</b> – Resultado da consulta 17	16
<b>FIGURA 20</b> – Resultado da consulta 18	16
<b>FIGURA 21</b> – Resultado da consulta 19	16
<b>FIGURA 22</b> – Resultado da consulta 20	16

<b>FIGURA 23 – Resultado da consulta 21</b>	16
<b>FIGURA 24 – Resultado da consulta 22</b>	16
<b>FIGURA 25 – Resultado da consulta 23</b>	16
<b>FIGURA 26 – Resultado da consulta 24</b>	16
<b>FIGURA 27 – Resultado da consulta 25</b>	16
<b>FIGURA 28 – Resultado da consulta 26</b>	16
<b>FIGURA 29 – Resultado da consulta 27</b>	16
<b>FIGURA 30 – Resultado da consulta 28</b>	16
<b>FIGURA 31 – Resultado da consulta 29</b>	16
<b>FIGURA 32 – Resultado da consulta 30</b>	16

## **LISTA DE TABELAS**

<b>TABELA 1 – Telefones</b>	<b>16</b>
<b>TABELA 2 – Email</b>	<b>17</b>
<b>TABELA 3 – Clientes</b>	<b>17</b>
<b>TABELA 4 – Cidades</b>	<b>17</b>
<b>TABELA 5 – Compras</b>	<b>17</b>
<b>TABELA 6 – Itens da Compra</b>	<b>17</b>
<b>TABELA 7 – Produtos</b>	<b>17</b>
<b>TABELA 8 – Categoria</b>	<b>17</b>
<b>TABELA 9 – Fornecedores e Produtos</b>	<b>17</b>
<b>TABELA 10 – Fornecedores</b>	<b>17</b>
<b>TABELA 11 – Pagamento</b>	<b>17</b>
<b>TABELA 12 – Telefone Fornecedor</b>	<b>17</b>
<b>TABELA 13 – Email Fornecedor</b>	<b>17</b>
<b>TABELA 14 – Relacionamento</b>	<b>17</b>

## LISTA DE ALGORITMOS

<b>ALGORITMO 1 – Código fonte do banco de dados (MySQL)</b>	18
<b>ALGORITMO 2 – Código de inserção dos dados (MySQL)</b>	18
<b>ALGORITMO 3 – Código da consulta 1</b>	18
<b>ALGORITMO 4 – Código da consulta 2</b>	18
<b>ALGORITMO 5 – Código da consulta 3</b>	18
<b>ALGORITMO 6 – Código da consulta 4</b>	18
<b>ALGORITMO 7 – Código da consulta 5</b>	18
<b>ALGORITMO 8 – Código da consulta 6</b>	18
<b>ALGORITMO 9 – Código da consulta 7</b>	18
<b>ALGORITMO 10 – Código da consulta 8</b>	18
<b>ALGORITMO 11 – Código da consulta 9</b>	18
<b>ALGORITMO 12 – Código da consulta 10</b>	18
<b>ALGORITMO 13 – Código da consulta 11</b>	18
<b>ALGORITMO 14 – Código da consulta 12</b>	18
<b>ALGORITMO 15 – Código f da consulta 13</b>	18
<b>ALGORITMO 16 – Código da consulta 14</b>	18
<b>ALGORITMO 17 – Código da consulta 15</b>	18
<b>ALGORITMO 18 – Código da consulta 16</b>	18
<b>ALGORITMO 19 – Código da consulta 17</b>	18
<b>ALGORITMO 20 – Código da consulta 18</b>	18
<b>ALGORITMO 21 – Código da consulta 19</b>	18
<b>ALGORITMO 22 – Código da consulta 20</b>	18



<b>ALGORITMO 23 – Código da consulta 21</b>	18
<b>ALGORITMO 24 – Código da consulta 22</b>	18
<b>ALGORITMO 25 – Código da consulta 23</b>	18
<b>ALGORITMO 26 – Código da consulta 24</b>	18
<b>ALGORITMO 27 – Código da consulta 25</b>	18
<b>ALGORITMO 28 – Código da consulta 26</b>	18
<b>ALGORITMO 29 – Código da consulta 27</b>	18
<b>ALGORITMO 30 – Código da consulta 28</b>	18
<b>ALGORITMO 31 – Código da consulta 29</b>	18
<b>ALGORITMO 32 – Código da consulta 30</b>	18

## LISTA DE SIGLAS

**IFSP** Instituto Federal de Educação, Ciência e Tecnologia de São Paulo

**DER** *Diagrama Entidade-Relacionamento*

## SUMÁRIO

<b>1</b>	<b>INTRODUÇÃO</b>	<b>12</b>
<b>1.1</b>	<b>Objetivos</b>	<b>12</b>
<b>1.2</b>	<b>Justificativa</b>	<b>12</b>
<b>1.3</b>	<b>Aspectos Metodológicos</b>	<b>12</b>
<b>1.4</b>	<b>Aporte Teórico</b>	<b>13</b>
<b>2</b>	<b>FUNDAMENTAÇÃO TEÓRICA</b>	<b>14</b>
<b>2.1</b>	<b>Primeiro Tópico</b>	<b>14</b>
<b>2.2</b>	<b>Segundo Tópico</b>	<b>14</b>
<b>2.3</b>	<b>Trabalhos Relacionados</b>	<b>15</b>
<b>3</b>	<b>PROJETO PROPOSTO (METODOLOGIA)</b>	<b>16</b>
<b>3.1</b>	<b>Considerações Iniciais</b>	<b>16</b>
<b>3.2</b>	<b>Requisitos</b>	<b>16</b>
<b>3.3</b>	<b>Casos de Uso</b>	<b>16</b>
<b>3.3.1</b>	<b>DIAGRAMA DE CASOS DE USO</b>	<b>16</b>
<b>3.3.2</b>	<b>DESCRIÇÕES DOS CASOS DE USO</b>	<b>16</b>
<b>3.4</b>	<b>Arquitetura</b>	<b>17</b>
<b>3.5</b>	<b>Projeto de Dados</b>	<b>17</b>
<b>3.6</b>	<b>Interfaces</b>	<b>17</b>
<b>3.7</b>	<b>Implementação</b>	<b>18</b>
<b>3.8</b>	<b>Testes e Falhas Conhecidas</b>	<b>18</b>
<b>3.9</b>	<b>Implantação</b>	<b>19</b>
<b>3.10</b>	<b>Manual de Usuário</b>	<b>19</b>

3.11	Resultados Esperados (Entrega Parcial)	19
4	RESULTADOS OBITIDOS	20
4.1	Ferramentas	20
4.2	Resultados	20
4.3.1	MODELO CONCEITUAL	21
4.3.2	MODELO LÓGICO	21
4.3.3	DICIONARIO DE DADOS	21
4.3.4	MODELO FISICO DO SISTEMA	21
4.3.4	CONSULTAS REALIZADAS NO BANCO DE DADOS	21
5	CONCLUSÃO	23
	REFERÊNCIAS	24

## **1 INTRODUÇÃO**

As pequenas e médias lojas de comércio enfrentam inúmeras dificuldades para se incluírem no mercado digital, nesse cenário, gerenciar uma loja sem um sistema de gestão eficiente é uma tarefa quase impossível e extremamente complexa, demandando muitos funcionários e recursos. Portanto, a implementação de um sistema de gestão eficaz se apresenta como a solução ideal para os problemas enfrentados pelas pequenas e médias lojas a fim de proporcionar maior flexibilidade nas relações entre o cliente e o comércio.

Este trabalho se concentra no desenvolvimento e implementação de um sistema abrangente para o gerenciamento de uma loja de peças de motocicletas, visando não apenas atender às necessidades atuais, mas também migrar de um modo analógico de vendas para um sistema híbrido, tendo maior foco em vendas online antecipando e preparando-se para os requisitos futuros dessa loja.

### **1.1 Objetivos**

Os objetivos propostos para este trabalho foi o desenvolvimento de um sistema otimizado e intuitivo que integre os processos de vendas realizados pela loja de peças, além de contribuir para a melhora da experiência do cliente e o aumento da eficiência da loja.

### **1.2 Justificativa**

A escolha de desenvolver um sistema integrado de gerenciamento para uma loja com foco na venda de peças é decorrente do crescente nível de complexidade que o setor varejista vem ganhando nos últimos anos, somado a demandas que exigem soluções digitais aliada ao dever de oferecer serviços de qualidade. Nessa conjunção, a implantação de soluções tecnológicas é indispensável para superar os desafios e aperfeiçoar a eficiência e qualidade dos serviços prestados.

Este trabalho justifica-se pela necessidade de lidar com os problemas operacionais dentro do ambiente da loja de peças, encaixando-se nas normas e regulamentações vigentes para o enfrentamento dos obstáculos. O desenvolvimento do sistema apresentado busca ser uma solução dinâmica para a loja, projetando afetar positivamente a eficácia dos serviços prestados por ela.

### **1.3 Aspectos Metodológicos**

Para desenvolver o sistema de gestão da loja de peças de motos, foram analisados diversos métodos para determinar o que traria mais benefícios ao cliente e à loja. A coleta de dados foi realizada por meio de entrevistas informais com o proprietário e os funcionários, visando coletar as informações necessárias para a implantação do sistema. Também foi realizado um estudo bibliográfico sobre sistemas de gestão e melhores práticas aplicadas a lojas de peças de motos. Esse método conectou estudos semelhantes, fornecendo uma base teórica robusta para o desenvolvimento do sistema. A análise incluiu livros de referência sobre modelagem de dados, implementação de sistemas e otimização de processos. Ao combinar métodos exploratórios e bibliográficos, foi possível obter uma compreensão completa das necessidades da loja, resultando em um sistema de gestão bem-estruturado e eficaz.

### **1.4 Aporte Teórico**

Os sistemas de gestão são ferramentas essenciais para o controle e organização eficaz dos diversos setores. Eles são projetados para automatizar tarefas e fornecer dados valiosos por meio de relatórios e análises, com bases teóricas sólidas. Autores como Peter Rob (2010) e Carlos Coronel (2010) enfatizam o desenvolvimento de diversos modelos de dados, cruciais para a estruturação eficaz destes sistemas. Como parte de uma loja de peças para motocicletas, um sistema de gestão bem estruturado é fundamental para garantir a eficiência operacional e a satisfação

dos clientes. Apresentamos então a estrutura conceitual de um modelo de dados específico para este ambiente.

## **2 FUNDAMENTAÇÃO TEÓRICA**

Como já apresentado um pouco do projeto, essa seção será de aprofundamento da metodologia escolhida para o levantamento dos requisitos, modelagem de dados, a ferramenta que será utilizada para a realização deste trabalho e a descrição do projeto.

### **2.1 Primeiro Tópico (renomeie esta subseção)**

Aqui você explicará um conceito importante para o entendimento do trabalho, citando as referências estudadas e usadas nesta pesquisa. Exemplo: Segundo Silva (2002) os sistemas X são...

### **2.2 Segundo Tópico (renomeie esta subseção)**

Aqui você pode explicar outro conceito importante para o entendimento do trabalho.

### **2.3 Trabalhos Relacionados**

Descreva trabalhos já existentes relacionados ao seu, o que eles têm igual, o que têm diferente, porque eles não atendem os objetivos do seu trabalho.

Estas foram as bases teóricas que sustentam o projeto de... (falar um pouco do projeto) desenvolvido no presente trabalho..., sobre o qual será relatado na seção seguinte.

### **3 PROJETO PROPOSTO**

Como já foi apresentado um pouco do projeto no final da fundamentação teórica, aqui já será falado tudo sobre o projeto, como foi desenvolvido (crie subseções adequadas ao seu trabalho). Exemplo:

Nesta seção serão apresentadas detalhadamente a metodologia utilizada neste trabalho, porque esta foi a escolhida (porque esta é mais adequada que outras) e suas etapas, os documentos referentes ao sistema proposto, como eles foram elaborados e demais artefatos referentes a este projeto.

#### **3.1 Apresentação de Figuras**

#### **3.2 Apresentação de Tabelas ou Quadros**

#### **3.3 Apresentação de Equações e Fórmulas**

#### **3.4 Apresentação de Algoritmos**



## **4 RESULTADOS OBITIDOS**

Durante o desenvolvimento do projeto de um sistema de gerenciamento para uma loja de peças de moto, conseguimos elaborar um modelo conceitual de dados, conhecido como Diagrama de Entidade-Relacionamento (DER). Esse modelo foi criado para oferecer uma visão introdutória das relações e entidades essenciais do banco de dados do projeto.

### **4.1 Ferramentas**

Para a realização dos trabalhos de modelagem de dados foi utilizado ferramentas que auxiliam no desenvolvimento dos modelos logico e físico juntamente com o código SQL do banco de dados.

A primeira e mais simples é a plataforma DRAWIO que foi usado para elaborar o modelo conceitual. A escolha por essa ferramenta foi a facilidade de criar os diagramas e seus relacionamentos de forma fácil e simples agilizando outras partes do desenvolvimento do projeto.

Outra plataforma que foi muito utilizada no desenvolvimento do projeto do sistema de gerenciamento de banco foi o MySQL Workbench 8.0 uma plataforma especializada nessa área. Com o Workbench foi desenvolvido o modelo logico mais completo e corrigido contendo as tabelas e os atributos com seus respectivos tipos de dados. Ainda sobre o Workbench ele foi responsável por fazer o código do banco de dados conhecido como modelo físico, que após passar por pequenas alterações estava pronto para os testes.

Por último foi usado o Visual Studio Code para a alterações do código e escrita das consultas além de ser a plataforma de onde foi pego os algoritmos para serem colocados no trabalho, ajudando a visualização do código.

### **4.2 Resultados**

Nessa seção, os resultados devem ser descritos de maneira objetiva, sem interpretação, obedecendo uma sequência lógica usando texto, figuras e tabelas. Ela deve ser organizada de tal forma que se destaque as evidências necessárias para responder cada questão de pesquisa ou hipótese que você investigou. Deve ser escrita de forma concisa e objetiva.

**Dicas:**

Quando você estabelece uma hipótese ou questão de pesquisa, os dados do estudo são observados, coletados e analisados de forma que responda as questões. Caso você esteja utilizando uma abordagem mais simples, essa análise é feita visualizando figuras e tabelas, fazendo cálculos de média, desvio padrão etc. Utilizando uma análise mais rebuscada, você pode interpretar uma variedade de testes estatísticos com diferentes técnicas.

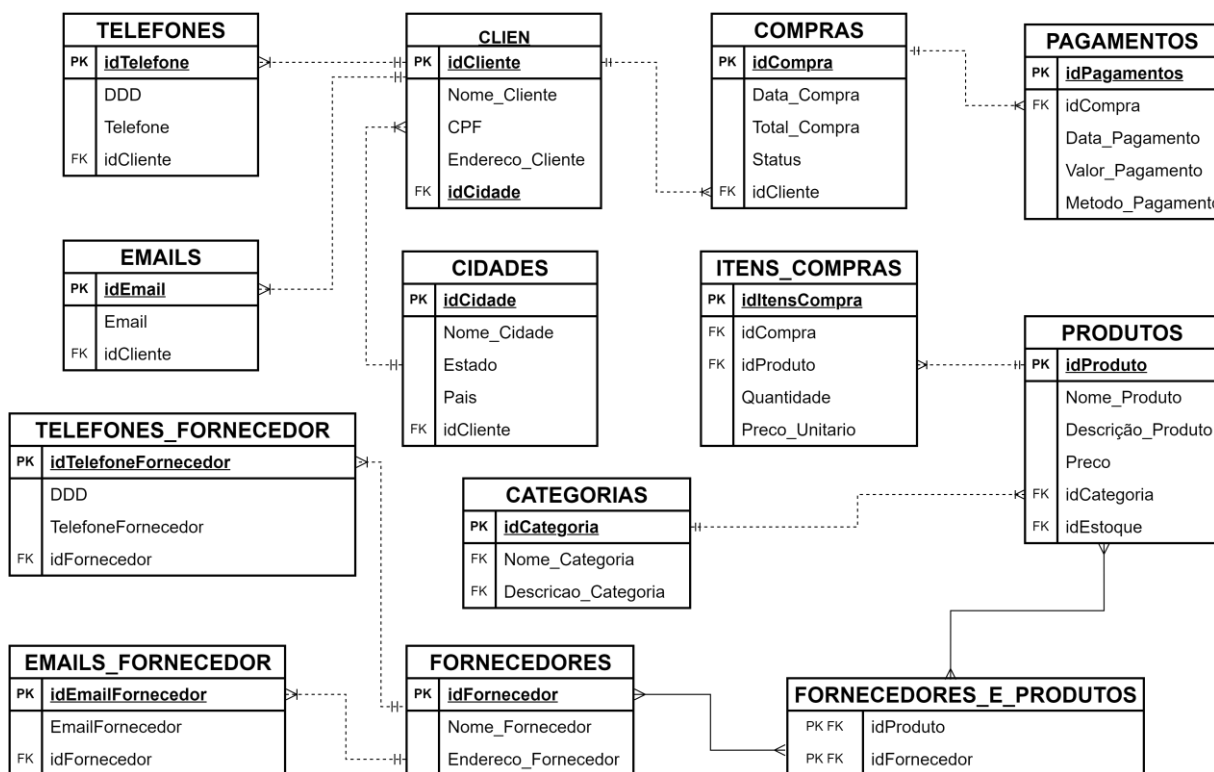
Escreva os resultados para mostrar o maior número possível de informações para o leitor em relação àqueles aspectos analisados e aos seus possíveis relacionamentos. Organize os resultados com base na sequência de figuras e tabelas. Olhe para a tabela e figura e identifique três palavras-chave, isso vai ajudar quando você começar a escrever sobre aquela tabela ou figura.

A seção de resultados é feita com base no texto criado para descrever os achados identificados, referenciando as figuras e tabelas sempre que possível. Você deve conduzir o leitor de forma que fique claro os achados do seu estudo. Esses achados vão depender do tipo de questão de pesquisa. Eles podem incluir tendências, diferenças, similaridades, correlações, mínimos, máximos etc.

Caso você não ache o resultado que esperava, isso pode ser algum erro na definição da hipótese ou precisa ser reformulada ou talvez tenha tropeçado em algo inesperado que precisa ser melhor investigado. Em qualquer um desses casos, os resultados são importantes mesmo que eles não deem suporte à sua hipótese. Não ache que resultados diferentes do que você esperava são resultados ruins. Se você fez o estudo com qualidade, mesmo resultados ruins podem gerar importantes descobertas na área. Desta forma, escreva seus resultados honestamente!!!

#### 4.1.1 MODELO CONCEITUAL

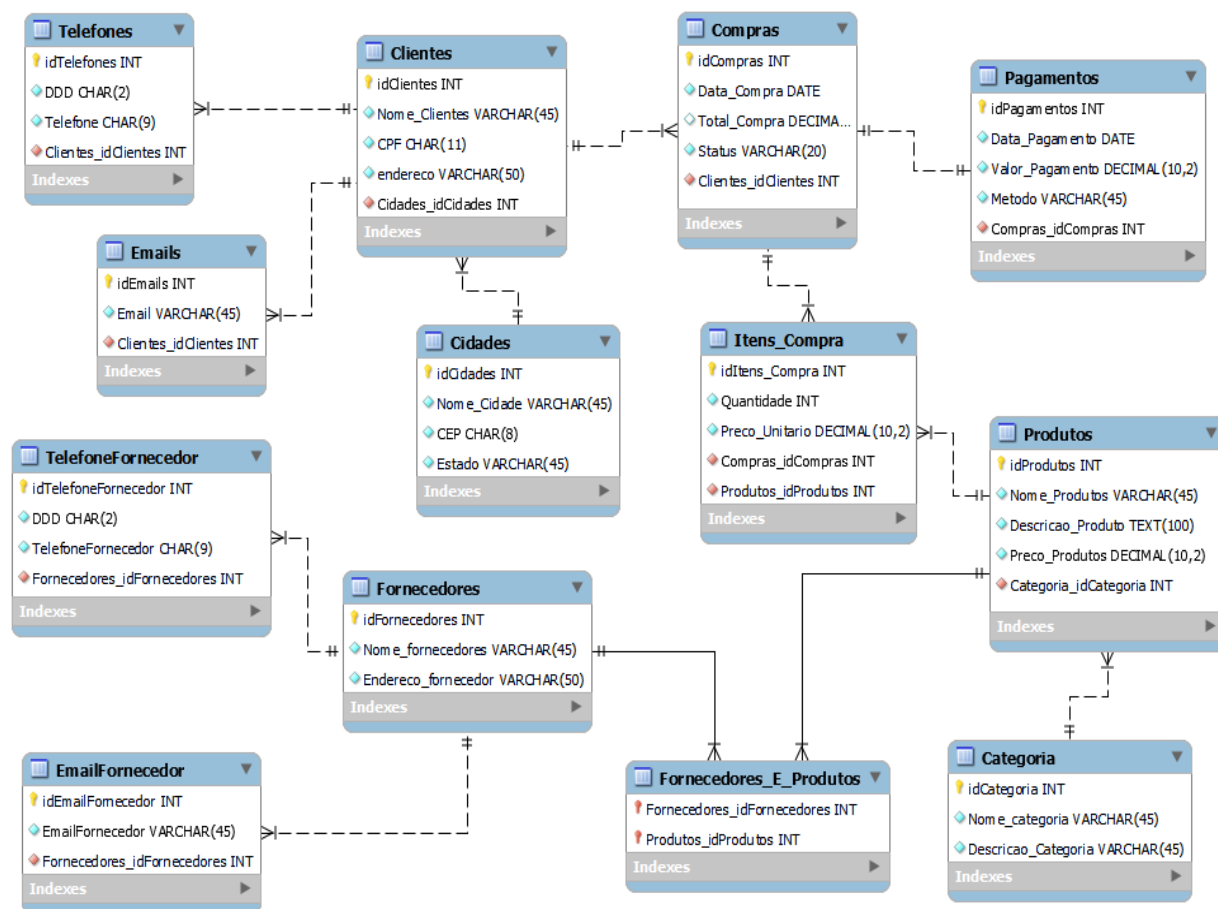
O modelo conceitual simplifica os relacionamentos das tabelas, organizando os campos de forma a representar cada linha como uma entrada única de dados, onde cada coluna designa um atributo específico dessa entrada. Além disso, as tabelas são compostas por chaves primárias únicas e chaves estrangeiras que conectam uma tabela a outra. Os nomes dos campos refletem os dados que serão armazenados.



**Figura 1 – Modelo relacional do sistema da loja de peças (2024)**

#### 4.1.2 MODELO LÓGICO

O modelo lógico é uma representação mais detalhada e estruturada do modelo conceitual, onde os conceitos abstratos do modelo conceitual são traduzidos em estruturas de dados concretas, como tabelas, colunas e chaves. Ele define como os dados serão armazenados e organizados no banco de dados, incluindo os tipos de dados, as restrições de integridade e as relações entre as tabelas.



**Figura 2 – Modelo lógico do sistema da loja de peças (2024)**

#### 4.1.3 DICIONARIO DE DADOS

**Tabela 1 - Telefones**

TELEFONE				
Nome da coluna	Tipo de dado	Restrição	Valor padrão	descrição
idTelefones	Inteiro	PK, NOT NULL, AUTO_INCREMENT	N/D	Chave primaria, referente a código único da tabela Telefone
DDD	Char	N/D	'00'	Referente ao prefixo telefônico do cliente
Telefone	Char	N/D	'000000000'	Referente ao telefone do cliente

idClientes	Inteiro	FK, NOT NULL	N/D	Chave estrangeira que refereência a tabela Clientes
------------	---------	--------------	-----	---

**Tabela 2 - Email**

E-MAIL				
Nome da coluna	Tipo de dado	Restrição	Valor padrão	descrição
idEmails	Inteiro	PK, NOT NULL, AUTO_INCREMENT	N/D	Chave primaria referente ao código único da tabela Email
Email	Varchar	N/D	N/D	Referente ao telefone do cliente
idCliente	Inteiro	FK, NOT NULL	N/D	Chave estrangeira que refereência a tabela Clientes

**Tabela 3 – Clientes**

CLIENTES				
Nome da coluna	Tipo de dado	Restrição	Valor padrão	descrição
idClientes	Inteiro	PK, NOT NULL, AUTO_INCREMENT	N/D	Chave primaria referente ao código único da tabela Clientes
Nome	Varchar	NOT NULL	N/D	Referente ao nome do cliente
CPF	Char	NOT NULL	N/D	Referente ao cpf do cliente
Endereço	Varchar	NOT NULL	N/D	Referente ao endereço do cliente
idCidades	Inteiro	FK, NOT NULL	N/D	Chave estrangeira referente a tabela Cidades

**Tabela 4 – Cidades**

CIDADES				
Nome da coluna	Tipo de dado	Restrição	Valor padrão	descrição
idCidade	Inteiro	PK, NOT NULL, AUTO_INCREMENT	N/D	Chave primária referente ao código único da tabela Cidade
nome	Varchar	NOT NULL	N/D	Referente ao nome da cidade
CEP	Char	NOT NULL	N/D	Referente ao CEP da cidade
Estado	Varchar	NOT NULL	N/D	Referente ao nome do estado

**Tabela 5 - Compras**

COMPRAS				
Nome da coluna	Tipo de dado	Restrição	Valor padrão	descrição
idCompra	Inteiro	PK, NOT NULL, AUTO_INCREMENT	N/D	Chave primária referente ao código único da tabela Compras
Data_compra	Date	NOT NULL	N/D	Referente a data que foi realizada a compra
Total_compra	Decimal	NOT NULL	N/D	Referente ao total de produtos
status	Varchar	NOT NULL	N/D	Referente ao status da compra
idClientes	Inteiro	FK, NOT NULL	N/D	Chave estrangeira que refereência a tabela Clientes

**Tabela 6 - Itens compra**

ITENS COMPRA				
Nome da coluna	Tipo de dado	Restrição	Valor padrão	descrição
idItensCompra	Inteiro	PK, NOT NULL, AUTO_INCREMENT	N/D	Chave primária referente ao código único da tabela Itens Compra
quantidade	inteiro	NOT NULL	N/D	Referente a quantas unidades
Preço_unitario	Decimal	NOT NULL	N/D	Referente ao preço de um único produto
idCompra	inteiro	FK, NOT NULL	N/D	Chave estrangeira que refereência a tabela Compras
idProdutos	Inteiro	FK, NOT NULL	N/D	Chave estrangeira que refereência a tabela Produtos

**Tabela 7 - Produtos**

PRODUTOS				
Nome da coluna	Tipo de dado	Restrição	Valor padrão	descrição
idProdutos	Inteiro	PK, NOT NULL, AUTO_INCREMENT	N/D	Chave primária referente ao código único da tabela Produtos
Nome_produto	Varchar	NOT NULL	N/D	Referente ao nome dos produtos
Descrição_produto	TEXT	NOT NULL	N/D	Descrição do produto
Preço	Decimal	NOT NULL	N/D	Referente ao preco unitario do produto
idCategoria	Inteiro	FK, NOT NULL	N/D	Chave estrangeira referente a tabela categoria

**Tabela 8 - Categoria**

CATEGORIA				
Nome da coluna	Tipo de dado	Restrição	Valor padrão	descrição
idCategoria	Inteiro	PK, NOT NULL, AUTO_INCREMENT	N/D	Chave primária referente ao código único da tabela Compras
Nome-categoria	Varchar	NOT NULL	N/D	Referente a data que foi realizada a compra
Descrição_categoria	Varchar	NOT NULL	N/D	Breve descrição da categoria

**Tabela 9 – Fornecedores E Produtos**

FORNECEDOR E PRODUTO				
Nome da coluna	Tipo de dado	Restrição	Valor padrão	descrição
idFornecedores	Inteiro	PK FK, NOT NULL	N/D	Chave primária referente ao código único da tabela Compras
idProdutos	Inteiro	PK FK, NOT NULL	N/D	Referente a data que foi realizada a compra

**Tabela 10 – Fornecedores**

FORNECEDORES				
Nome da coluna	Tipo de dado	Restrição	Valor padrão	descrição



idFornecedores	Inteiro	PK, NOT NULL, AUTO_INCREMENT	N/D	Chave primária referente ao código único da tabela Compras
Nome_fornecedor	Varchar	NOT NULL	N/D	Referente ao nome do fornecedor
Endereço_fornecedor	Varchar	NOT NULL	N/D	Endereço do fornecedor

**Tabela 11 - Pagamento**

PAGAMENTO				
Nome da coluna	Tipo de dado	Restrição	Valor padrão	descrição
idPagamento	Inteiro	PK, NOT NULL, AUTO_INCREMENT	N/D	Chave primária referente ao código único da tabela Pagamento
Data_pagamento	DATE	NOT NULL	N/D	Referente a data de pagamento
Valor_Pagamento	Decimal	NOT NULL	N/D	Referente ao valor total da compra
metodo	Varchar	NOT NUL	N/D	Forma de pagamento que o cliente optou

**Tabela 12 – Telefones dos Fornecedores**

TELEFONE_FORNECEDORES				
Nome da coluna	Tipo de dado	Restrição	Valor padrão	descrição
idTelefones-Fornecedores	Inteiro	PK, NOT NULL, AUTO_INCREMENT	N/D	Chave primária, referente a código único

				da tabela telefones Fornecedores
DDD	Char	N/D	'00'	Referente ao prefixo telefônico do fornecedor
Telefone-Fornecedores	Char	N/D	'000000000'	Referente aos telefones dos Fornecedores
idFornecedores	Inteiro	FK, NOT NULL	N/D	Chave estrangeira que referência a tabela Fornecedores

**Tabela 13 – Email dos Fornecedores**

E-MAIL_FORNECEDORES				
Nome da coluna	Tipo de dado	Restrição	Valor padrão	descrição
idEmails-Fornecedores	Inteiro	PK, NOT NULL, AUTO_INCREMENT	N/D	Chave primária referente ao código único da tabela Email Fornecedores
EmailFornecedor	Varchar	N/D	N/D	Referente aos E-mails dos fornecedores
idFornecedores	Inteiro	FK, NOT NULL	N/D	Chave estrangeira que referência a tabela Fornecedores

**Tabela 14 - Relacionamentos**

RELACIONAMENTOS			
Nome da tabela	Relacionado com	Nome do relacionamento	descrição
Clientes	Telefone	possui	Telefones que o cliente pode possuir
Clientes	Email	possui	E-mails que o cliente pode possuir
Clientes	Cidade	reside	O cliente mora em uma cidade

Compras	Clientes	fazem	O cliente faz uma compra
Itens Compra	Compras	tem	Compras tem itens
Produtos	Itens Compra	Não tem	Itens são um ou mais produtos
produtos	Fornecedor e Produtos	Não tem	Produto tem um fornecedor ou vários
Fornecedor e Produto	Fornecedor	Não tem	Um fornecedor tem vários produtos que fornece
Fornecedor	Telefone	possui	Telefones que os fornecedores podem possuir
Fornecedor	Email	possui	E-mails que os fornecedores podem possuir
Categorias	Produtos	Tem uma	Cada produto tem uma categoria
Compras	Pagamento	geram	Cada compra gera um pagamento a ser realizado

#### 4.1.4 MODELO FISICO DO SISTEMA DE UMA LOJA DE VENDA DE PEÇAS DE MOTO

O modelo físico consiste no código MYSQL de criação do banco, tabelas que fazem parte e a inserção dos dados que serão apresentados através das consultas no banco.

```
-- -----
-- Schema BDD
-- -----

CREATE SCHEMA IF NOT EXISTS `BDD` DEFAULT CHARACTER SET utf8 ;
USE `BDD` ;

-- -----
-- `Cidades`
-- -----

CREATE TABLE `Cidades` (
  `idCidades` INT NOT NULL AUTO_INCREMENT,
  `Nome_Cidade` VARCHAR(45) NOT NULL,
  `CEP` CHAR(8) NOT NULL,
  `Estado` VARCHAR(45) NOT NULL,
  PRIMARY KEY (`idCidades`))
ENGINE = InnoDB;
```

```

-- -----
-- `Clientes`
-- -----

CREATE TABLE `Clientes` (
  `idClientes` INT NOT NULL AUTO_INCREMENT,
  `Nome_Clientes` VARCHAR(45) NOT NULL,
  `CPF` CHAR(11) NOT NULL,
  `endereco` VARCHAR(50) NOT NULL,
  `Cidades_idCidades` INT NOT NULL,
  PRIMARY KEY (`idClientes`),
  UNIQUE INDEX `CPF_UNIQUE` (`CPF` ASC) VISIBLE,
  UNIQUE INDEX `idClientes_UNIQUE` (`idClientes` ASC) VISIBLE,
  INDEX `fk_Clientes_Cidades1_idx` (`Cidades_idCidades` ASC) VISIBLE,
  CONSTRAINT `fk_Clientes_Cidades1`
    FOREIGN KEY (`Cidades_idCidades`)
    REFERENCES `BDD`.`Cidades` (`idCidades`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION)
ENGINE = InnoDB;

-- -----
-- `Telefones`
-- -----

CREATE TABLE `Telefones` (
  `idTelefones` INT NOT NULL AUTO_INCREMENT,
  `DDD` CHAR(2) NOT NULL,
  `Telefone` CHAR(9) NOT NULL,
  `Clientes_idClientes` INT NOT NULL,
  PRIMARY KEY (`idTelefones`),
  INDEX `fk_Telefones_Clientes_idx` (`Clientes_idClientes` ASC) VISIBLE,
  CONSTRAINT `fk_Telefones_Clientes`
    FOREIGN KEY (`Clientes_idClientes`)
    REFERENCES `BDD`.`Clientes` (`idClientes`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION)
ENGINE = InnoDB;

-- -----
-- `Emails`
-- -----

CREATE TABLE `Emails` (
  `idEmails` INT NOT NULL AUTO_INCREMENT,
  `Email` VARCHAR(45) NOT NULL,

```

```

`Clientes_idClientes` INT NOT NULL,
PRIMARY KEY (`idEmails`),
INDEX `fk_Emails_Clientes1_idx` (`Clientes_idClientes` ASC) VISIBLE,
CONSTRAINT `fk_Emails_Clientes1`
  FOREIGN KEY (`Clientes_idClientes`)
  REFERENCES `BDD`.`Clientes` (`idClientes`)
  ON DELETE NO ACTION
  ON UPDATE NO ACTION)
ENGINE = InnoDB;

-- -----
-- `Compras`
-- -----

CREATE TABLE `Compras` (
  `idCompras` INT UNSIGNED NOT NULL,
  `Data_Compra` DATE NOT NULL,
  `Total_Compra` DECIMAL(10,2) NULL,
  `Status` VARCHAR(20) NOT NULL,
  `Clientes_idClientes` INT NOT NULL,
  PRIMARY KEY (`idCompras`),
  INDEX `fk_Compras_Clientes1_idx` (`Clientes_idClientes` ASC) VISIBLE,
  CONSTRAINT `fk_Compras_Clientes1`
    FOREIGN KEY (`Clientes_idClientes`)
    REFERENCES `BDD`.`Clientes` (`idClientes`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION)
ENGINE = InnoDB;

-- -----
-- `Pagamentos`
-- -----

CREATE TABLE `Pagamentos` (
  `idPagamentos` INT NOT NULL,
  `Data_Pagamento` DATE NOT NULL,
  `Valor_Pagamento` DECIMAL(10,2) NOT NULL,
  `Metodo` VARCHAR(45) NOT NULL,
  `Compras_idCompras` INT UNSIGNED NOT NULL,
  PRIMARY KEY (`idPagamentos`),
  INDEX `fk_Pagamentos_Compras1_idx` (`Compras_idCompras` ASC) VISIBLE,
  CONSTRAINT `fk_Pagamentos_Compras1`
    FOREIGN KEY (`Compras_idCompras`)
    REFERENCES `BDD`.`Compras` (`idCompras`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION)
ENGINE = InnoDB;

```

```

-- -----
-- `Categoria`
-- -----

CREATE TABLE `Categoria` (
  `idCategoria` INT NOT NULL,
  `Nome_categoria` VARCHAR(45) NOT NULL,
  `Descricao_Categoria` VARCHAR(45) NOT NULL,
  PRIMARY KEY (`idCategoria`))
ENGINE = InnoDB;

-- -----
-- `Produtos`
-- -----

CREATE TABLE `Produtos` (
  `idProdutos` INT NOT NULL,
  `Nome_Produtos` VARCHAR(45) NOT NULL,
  `Descricao_Produto` TEXT(100) NOT NULL,
  `Preco_Produtos` DECIMAL(10,2) NOT NULL,
  `Categoria_idCategoria` INT NOT NULL,
  PRIMARY KEY (`idProdutos`),
  INDEX `fk_Produtos_Categoria1_idx` (`Categoria_idCategoria` ASC) VISIBLE,
  CONSTRAINT `fk_Produtos_Categoria1`
    FOREIGN KEY (`Categoria_idCategoria`)
      REFERENCES `BDD`.`Categoria` (`idCategoria`)
      ON DELETE NO ACTION
      ON UPDATE NO ACTION)
ENGINE = InnoDB;

-- -----
-- `Itens_Compra`
-- -----

CREATE TABLE `Itens_Compra` (
  `idItens_Compra` INT NOT NULL,
  `Quantidade` INT NOT NULL,
  `Preco_Unitario` DECIMAL(10,2) NOT NULL,
  `Compras_idCompras` INT UNSIGNED NOT NULL,
  `Produtos_idProdutos` INT NOT NULL,
  PRIMARY KEY (`idItens_Compra`),
  INDEX `fk_Itens_Compra_Compras1_idx` (`Compras_idCompras` ASC) VISIBLE,
  INDEX `fk_Itens_Compra_Produtos1_idx` (`Produtos_idProdutos` ASC) VISIBLE,
  CONSTRAINT `fk_Itens_Compra_Compras1`
    FOREIGN KEY (`Compras_idCompras`)
      REFERENCES `BDD`.`Compras` (`idCompras`)

```

```

        ON DELETE NO ACTION
        ON UPDATE NO ACTION,
    CONSTRAINT `fk_Itens_Compra_Produtos1`
        FOREIGN KEY (`Produtos_idProdutos`)
        REFERENCES `BDD`.`Produtos` (`idProdutos`)
        ON DELETE NO ACTION
        ON UPDATE NO ACTION)
ENGINE = InnoDB;

-- -----
-- `Fornecedores`
-- -----

CREATE TABLE `Fornecedores` (
    `idFornecedores` INT NOT NULL,
    `Nome_fornecedores` VARCHAR(45) NOT NULL,
    `Endereco_fornecedor` VARCHAR(50) NOT NULL,
    PRIMARY KEY (`idFornecedores`))
ENGINE = InnoDB;

-- -----
-- `Fornecedores_E_Produtos`
-- -----

CREATE TABLE `Fornecedores_E_Produtos` (
    `Fornecedores_idFornecedores` INT NOT NULL,
    `Produtos_idProdutos` INT NOT NULL,
    PRIMARY KEY (`Fornecedores_idFornecedores`, `Produtos_idProdutos`),
    INDEX `fk_Fornecedores_has_Produtos_Produtos1_idx` (`Produtos_idProdutos`
ASC) VISIBLE,
    INDEX `fk_Fornecedores_has_Produtos_Fornecedores1_idx` (`Fornecedores_idFor-
necedores` ASC) VISIBLE,
    CONSTRAINT `fk_Fornecedores_has_Produtos_Fornecedores1`
        FOREIGN KEY (`Fornecedores_idFornecedores`)
        REFERENCES `BDD`.`Fornecedores` (`idFornecedores`)
        ON DELETE NO ACTION
        ON UPDATE NO ACTION,
    CONSTRAINT `fk_Fornecedores_has_Produtos_Produtos1`
        FOREIGN KEY (`Produtos_idProdutos`)
        REFERENCES `BDD`.`Produtos` (`idProdutos`)
        ON DELETE NO ACTION
        ON UPDATE NO ACTION)
ENGINE = InnoDB;

-- -----
-- `TelefoneFornecedor`

```

```

-----
CREATE TABLE `TelefoneFornecedor` (
  `idTelefoneFornecedor` INT NOT NULL,
  `DDD` CHAR(2) NOT NULL,
  `TelefoneFornecedor` CHAR(9) NOT NULL,
  `Fornecedores_idFornecedores` INT NOT NULL,
  PRIMARY KEY (`idTelefoneFornecedor`),
  INDEX `fk_TelefoneFornecedor_Fornecedores1_idx` (`Fornecedores_idFornecedores`
  ASC) VISIBLE,
  CONSTRAINT `fk_TelefoneFornecedor_Fornecedores1`
    FOREIGN KEY (`Fornecedores_idFornecedores`)
    REFERENCES `BDD`.`Fornecedores` (`idFornecedores`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION)
ENGINE = InnoDB;

-----
-- `EmailFornecedor`
-----

CREATE TABLE `EmailFornecedor` (
  `idEmailFornecedor` INT NOT NULL,
  `EmailFornecedor` VARCHAR(45) NOT NULL,
  `Fornecedores_idFornecedores` INT NOT NULL,
  PRIMARY KEY (`idEmailFornecedor`),
  INDEX `fk_EmailFornecedor_Fornecedores1_idx` (`Fornecedores_idFornecedores`
  ASC) VISIBLE,
  CONSTRAINT `fk_EmailFornecedor_Fornecedores1`
    FOREIGN KEY (`Fornecedores_idFornecedores`)
    REFERENCES `BDD`.`Fornecedores` (`idFornecedores`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION)
ENGINE = InnoDB;

```

**Algoritmo 1** – código fonte do banco de dados (MYSQL)

```

-----
-- Inserir dados na tabela `Cidades`
INSERT INTO `Cidades` (`Nome_Cidade`, `CEP`, `Estado`) VALUES
('São Paulo', '01001000', 'São Paulo'),
('Rio de Janeiro', '20040000', 'Rio de Janeiro'),
('Belo Horizonte', '30140071', 'Minas Gerais'),
('Salvador', '40020000', 'Bahia'),
('Fortaleza', '60010000', 'Ceará');

-- Inserir dados na tabela `Clientes`

```



```

INSERT INTO `Clientes` (`Nome_Clientes`, `CPF`, `endereco`, `Cidades_idCidades`)
VALUES
('João Silva', '12345678901', 'Rua das Flores, 123', 1),
('Maria Oliveira', '23456789012', 'Avenida Atlântica, 456', 2),
('Pedro Souza', '34567890123', 'Praça da Liberdade, 789', 3),
('Ana Santos', '45678901234', 'Rua da Bahia, 101', 4),
('Carlos Pereira', '56789012345', 'Avenida Beira Mar, 202', 5);

-- Inserir dados na tabela `Telefones`
INSERT INTO `Telefones` (`DDD`, `Telefone`, `Clientes_idClientes`) VALUES
('11', '912345678', 1),
('21', '923456789', 2),
('31', '934567890', 3),
('71', '945678901', 4),
('85', '956789012', 5);

-- Inserir dados na tabela `Emails`
INSERT INTO `Emails` (`Email`, `Clientes_idClientes`) VALUES
('joao.silva@example.com', 1),
('maria.oliveira@example.com', 2),
('pedro.souza@example.com', 3),
('ana.santos@example.com', 4),
('carlos.pereira@example.com', 5);

-- Inserir dados na tabela `Compras`
INSERT INTO `Compras` (`idCompras`, `Data_Compra`, `Total_Compra`, `Status`,
`Clientes_idClientes`) VALUES
(1, '2024-01-10', 250.75, 'Concluída', 1),
(2, '2024-01-15', 320.50, 'Concluída', 2),
(3, '2024-01-20', 150.00, 'Pendente', 3),
(4, '2024-01-25', 400.99, 'Concluída', 4),
(5, '2024-01-30', 520.80, 'Cancelada', 5);

-- Inserir dados na tabela `Pagamentos`
INSERT INTO `Pagamentos` (`idPagamentos`, `Data_Pagamento`, `Valor_Pagamento`,
`Metodo`, `Compras_idCompras`) VALUES
(1, '2024-01-11', 250.75, 'Cartão de Crédito', 1),
(2, '2024-01-16', 320.50, 'Boleto', 2),
(3, '2024-01-21', 150.00, 'Pix', 3),
(4, '2024-01-26', 400.99, 'Cartão de Débito', 4),
(5, '2024-01-31', 520.80, 'Cartão de Crédito', 5);

-- Inserir dados na tabela `Categoria`
INSERT INTO `Categoria` (`idCategoria`, `Nome_categoria`, `Descricao_Catego-
ria`) VALUES
(1, 'Motor', 'Peças relacionadas ao motor'),
(2, 'Freios', 'Peças relacionadas ao sistema de freios'),

```

```

(3, 'Suspensão', 'Peças relacionadas à suspensão'),
(4, 'Transmissão', 'Peças relacionadas à transmissão'),
(5, 'Acessórios', 'Acessórios para motos');

-- Inserir dados na tabela `Produtos`
INSERT INTO `Produtos` (`idProdutos`, `Nome_Produtos`, `Descricao_Produto`,
`Preco_Produtos`, `Categoria_idCategoria`) VALUES
(1, 'Pistão', 'Pistão de alta performance', 150.00, 1),
(2, 'Disco de Freio', 'Disco de freio ventilado', 120.00, 2),
(3, 'Amortecedor', 'Amortecedor traseiro', 200.00, 3),
(4, 'Corrente', 'Corrente de transmissão reforçada', 80.00, 4),
(5, 'Capacete', 'Capacete com viseira anti-risco', 300.00, 5);

-- Inserir dados na tabela `Itens_Compra`
INSERT INTO `Itens_Compra` (`idItens_Compra`, `Quantidade`, `Preco_Unitario`,
`Compras_idCompras`, `Produtos_idProdutos`) VALUES
(1, 2, 150.00, 1, 1),
(2, 1, 120.00, 2, 2),
(3, 2, 200.00, 3, 3),
(4, 3, 80.00, 4, 4),
(5, 1, 300.00, 5, 5);

-- Inserir dados na tabela `Fornecedores`
INSERT INTO `Fornecedores` (`idFornecedores`, `Nome_fornecedores`, `Ende-
reco_fornecedor`) VALUES
(1, 'Motos Distribuidora', 'Rua das Motos, 1000'),
(2, 'Peças Rápidas', 'Avenida das Peças, 500'),
(3, 'Mototech', 'Praça das Motos, 250'),
(4, 'Moto Mercado', 'Rua das Peças, 800'),
(5, 'Acessórios Moto', 'Avenida dos Motociclistas, 1200');

-- Inserir dados na tabela `Fornecedores_E_Produtos`
INSERT INTO `Fornecedores_E_Produtos` (`Fornecedores_idFornecedores`, `Produ-
tos_idProdutos`) VALUES
(1, 1),
(2, 2),
(3, 3),
(4, 4),
(5, 5);

-- Inserir dados na tabela `TelefoneFornecedor`
INSERT INTO `TelefoneFornecedor` (`idTelefoneFornecedor`, `DDD`, `TelefoneFor-
necedor`, `Fornecedores_idFornecedores`) VALUES
(1, '11', '987654321', 1),
(2, '21', '987654322', 2),
(3, '31', '987654323', 3),
(4, '71', '987654324', 4),

```

```
(5, '85', '987654325', 5);

-- Inserir dados na tabela `EmailFornecedor`
INSERT INTO `EmailFornecedor` (`idEmailFornecedor`, `EmailFornecedor`, `Fornecedores_idFornecedores`) VALUES
(1, 'contato@motosdistribuidora.com.br', 1),
(2, 'vendas@pecasrapidas.com.br', 2),
(3, 'suporte@mototech.com.br', 3),
(4, 'contato@motomercado.com.br', 4),
(5, 'vendas@acessoriosmoto.com.br', 5);
```

**Algoritmo 2** – código de inserção de dados (MYSQL)

Os dados foram inseridos diretamente no código devido à quantidade reduzida de dados para esse teste. Essa abordagem simplificou a demonstração, considerando o pequeno número de registros disponíveis.

#### 4.1.5 CONSULTAS REALIZADAS NO BANCO DE DADOS

Consulta que mostra os clientes e seu telefone, Email, cidade e estado

```
-- consulta 1
SELECT Cli.Nome_Clientes, Tel.DDD, Tel.Telefone, Email.Email, Cid.Nome_Cidade,
Cid.Estado
FROM Clientes AS Cli
JOIN Telefones AS Tel ON Cli.idClientes = Tel.Clientes_idClientes
JOIN Emails AS Email ON Cli.idClientes = Email.Clientes_idClientes
JOIN Cidades AS Cid ON Cli.Cidades_idCidades = Cid.idCidades;
```

**Algoritmo 3** – Algoritmo da consulta 1 (MYSQL)

Result Grid   Filter Rows:   Export:   Wrap Cell Content:						
	Nome_Clientes	DDD	Telefone	Email	Nome_Cidade	Estado
▶	João Silva	11	912345678	joao.silva@example.com	São Paulo	São Paulo
	Maria Oliveira	21	923456789	maria.oliveira@example.com	Rio de Janeiro	Rio de Janeiro
	Pedro Souza	31	934567890	pedro.souza@example.com	Belo Horizonte	Minas Gerais
	Ana Santos	71	945678901	ana.santos@example.com	Salvador	Bahia
	Carlos Pereira	85	956789012	carlos.pereira@example.com	Fortaleza	Ceará

**Figura 3** – Resultado da consulta 1

Essa consulta mostra informações sobre compras de um cliente específico

```
-- consulta 2
```

```
SELECT Compras.idCompras, Compras.Data_Compra, Compras.Total_Compra, Com-
pras.Status, Pagamentos.Metodo, Pagamentos.Valor_Pagamento
FROM Compras
JOIN Pagamentos ON Compras.idCompras = Pagamentos.Compras_idCompras
JOIN Clientes ON Compras.Clientes_idClientes = Clientes.idClientes
WHERE Clientes.idClientes = 4;
```

**Algoritmo 4** – algoritmo da consulta 2 (MYSQL)

idCompras	Data_Compra	Total_Compra	Status	Metodo	Valor_Pagamento
4	2024-01-25	400.99	Concluída	Cartão de Débito	400.99
5	2024-01-30	520.80	Cancelada	Cartão de Crédito	520.80

**Figura 4** - Resultado da consulta 2

Consulta que lista todos os fornecedores e seus produtos

```
-- consulta 3
SELECT Fornecedores.Nome_fornecedores, Produtos.Nome_Produtos
FROM Fornecedores
JOIN Fornecedores_E_Produtos ON Fornecedores.idFornecedores = Fornecedo-
res_E_Produtos.Fornecedores_idFornecedores
JOIN Produtos ON Fornecedores_E_Produtos.Produtos_idProdutos = Produtos.idPro-
dutos;
```

**Algoritmo 5** – Algoritmo da consulta 3 (MYSQL)

Nome_fornecedores	Nome_Produtos
Motos Distribuidora	Pistão
Peças Rápidas	Disco de Freio
Mototech	Amortecedor
Moto Mercado	Corrente
Acessórios Moto	Capacete

**Figura 4** – Resultado da consulta 3

Obter o número total de clientes em cada cidade

```
-- consulta 4
SELECT Cidades.Nome_Cidade, COUNT(Clientes.idClientes) AS NumeroDeClientes
FROM Cidades
JOIN Clientes ON Cidades.idCidades = Clientes.Cidades_idCidades
GROUP BY Cidades.Nome_Cidade;
```

**Algoritmo 6** – Algoritmo da consulta 4 (MYSQL)

Result Grid			Filter Rows:	Export:	Wrap Cell Content:
	Nome_Cidade	NumeroDeClientes			
▶	São Paulo	1			
	Rio de Janeiro	1			
	Belo Horizonte	1			
	Salvador	1			
	Fortaleza	1			

**Figura 5** – Resultado da consulta 4

Lista todos os clientes que compraram um item específico

```
-- consulta 5
SELECT DISTINCT Clientes.Nome_Clientes
FROM Clientes
JOIN Compras ON Clientes.idClientes = Compras.Clientes_idClientes
JOIN Itens_Compra ON Compras.idCompras = Itens_Compra.Compras_idCompras
WHERE Itens_Compra.Produtos_idProdutos = 1;
```

**Algoritmo 7** – Algoritmo da consulta 5 (MYSQL)

Result Grid			Filter Rows:	Export:	Wrap Cell Content:
	Nome_Clientes				
▶	João Silva				

**Figura 6** – Resultado da consulta 5

Obter o valor total das compras feitas por cada cliente

```
-- consulta 6
SELECT Clientes.Nome_Clientes, SUM(Compras.Total_Compra) AS ValorTotalCompras
FROM Clientes
JOIN Compras ON Clientes.idClientes = Compras.Clientes_idClientes
GROUP BY Clientes.Nome_Clientes;
```

**Algoritmo 8** – Algoritmo da consulta 6 (MYSQL)

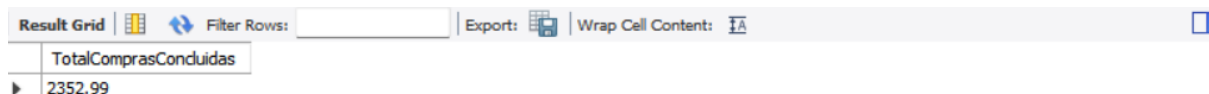
Result Grid			Filter Rows:	Export:	Wrap Cell Content:
	Nome_Clientes	ValorTotalCompras			
▶	João Silva	791.50			
	Maria Oliveira	320.50			
	Pedro Souza	990.00			
	Ana Santos	1522.79			

**Figura 7** – Resultado da consulta 6

Obter a soma total de todas as compras concluídas

```
-- consulta 7
SELECT SUM(Total_Compra) AS TotalComprasConcluidas
FROM Compras
WHERE Status = 'Concluída';
```

**Algoritmo 9** – Algoritmo da consulta 7 (MYSQL)



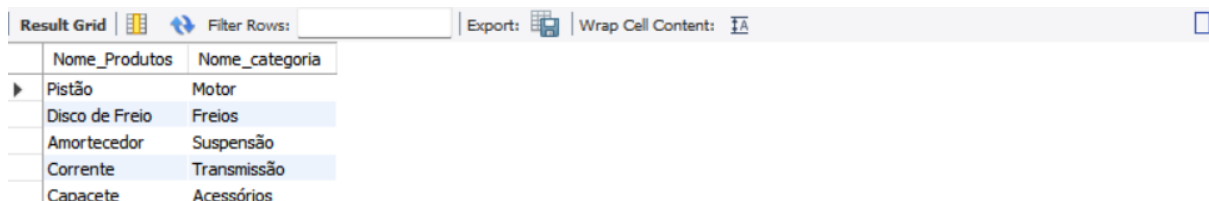
TotalComprasConcluidas
2352.99

**Figura 9** – Resultado da consulta 7

Lista os produtos e suas categorias

```
-- consulta 8
SELECT Produtos.Nome_Produtos, Categoria.Nome_categoria
FROM Produtos
JOIN Categoria ON Produtos.Categoria_idCategoria = Categoria.idCategoria;
```

**Algoritmo 10**– Algoritmo da consulta 8 (MYSQL)



Nome_Produtos	Nome_categoria
Pistão	Motor
Disco de Freio	Freios
Amortecedor	Suspensão
Corrente	Transmissão
Capacete	Acessórios

**Figura 10** – Resultado da consulta 8

Listar todas as compras realizadas em um período específico

```
-- consulta 9
SELECT * FROM Compras
WHERE Data_Compra BETWEEN '2024-01-01' AND '2024-01-31';
```

**Algoritmo 11** – Algoritmo da consulta 9 (MYSQL)

Result Grid

Filter Rows:

Edit:

Export/Import:

Wrap Cell Content:

	idCompras	Data_Compra	Total_Compra	Status	Cientes_idClientes
▶	1	2024-01-10	250.75	Concluída	1
	2	2024-01-15	320.50	Concluída	2
	3	2024-01-20	150.00	Pendente	3
	4	2024-01-25	400.99	Concluída	4
	5	2024-01-30	520.80	Cancelada	4
*	NULL	NULL	NULL	NULL	NULL

**Figura 11** – Resultado da consulta 9

Obter detalhes dos fornecedores que fornecem mais de um produto. Um adendo sobre essa tabela é que ela não retorna nenhum dado pelo fato de os fornecedores apenas fornecerem um único produto

```
-- consulta 10
SELECT Fornecedores.Nome_fornecedores, COUNT(Fornecedores_E_Produtos.Produ-
tos_idProdutos) AS NumeroDeProdutos
FROM Fornecedores
JOIN Fornecedores_E_Produtos ON Fornecedores.idFornecedores = Fornecedo-
res_E_Produtos.Fornecedores_idFornecedores
GROUP BY Fornecedores.idFornecedores
HAVING NumeroDeProdutos > 1;
```

**Algoritmo 12** – Algoritmo da consulta 10 (MYSQL)

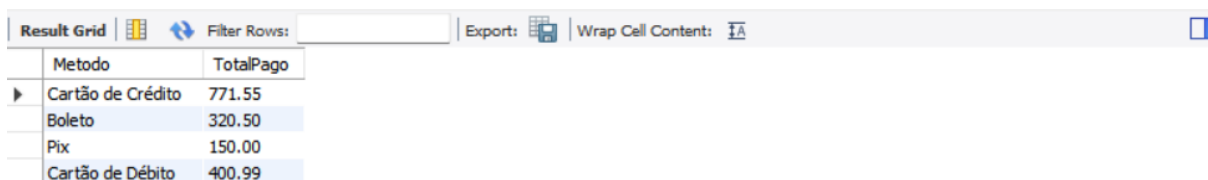
Result Grid	
Filter Rows:	
Export:	
Wrap Cell Content:	
Nome_fornecedores	NumeroDeProdutos

**Figura 12** – Resultado da consulta 10

Obter o total de pagamentos realizados por métodos de pagamento

```
-- consulta 11
SELECT Metodo, SUM(Valor_Pagamento) AS TotalPago
FROM Pagamentos
GROUP BY Metodo;
```

**Algoritmo 13** – Algoritmo da consulta 11 (MYSQL)



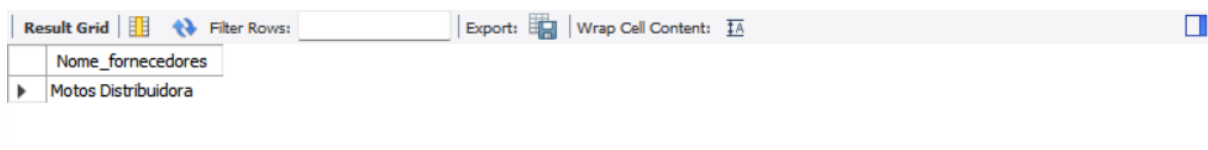
Metodo	TotalPago
Cartão de Crédito	771.55
Boleto	320.50
Pix	150.00
Cartão de Débito	400.99

**Figura 13** – Resultado da consulta 11

Listar todos os fornecedores que fornecem produtos de uma categoria específica

```
-- consulta 12
SELECT Fornecedores.Nome_fornecedores
FROM Fornecedores
JOIN Fornecedores_E_Produtos ON Fornecedores.idFornecedores = Fornecedores_E_Produtos.Fornecedores_idFornecedores
JOIN Produtos ON Fornecedores_E_Produtos.Produtos_idProdutos = Produtos.idProdutos
WHERE Produtos.Categoria_idCategoria = 1;
```

**Algoritmo 14** – Algoritmo da consulta 12 (MYSQL)



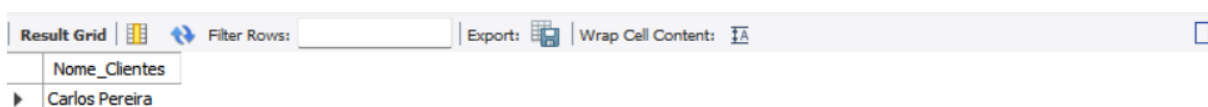
Nome_fornecedores
Motos Distribuidora

**Figura 14** – Resultado da consulta 12

Listar todos os clientes que não realizaram nenhuma compra

```
-- consulta 13
SELECT Clientes.Nome_Clientes
FROM Clientes
LEFT JOIN Compras ON Clientes.idClientes = Compras.Clientes_idClientes
WHERE Compras.idCompras IS NULL;
```

**Algoritmo 15** – Algoritmo da consulta 13 (MYSQL)



Nome_Clientes
Carlos Pereira

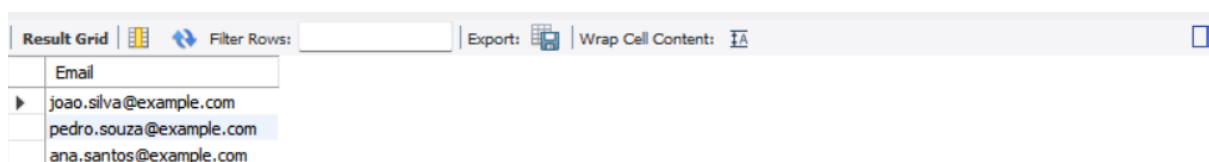
**Figura 15** – Resultado da consulta 13



Obter a lista de e-mails dos clientes que realizaram compras com um valor acima de uma quantia específica

```
-- consulta 14
SELECT Emails.Email
FROM Emails
JOIN Clientes ON Emails.Clientes_idClientes = Clientes.idClientes
JOIN Compras ON Clientes.idClientes = Compras.Clientes_idClientes
GROUP BY Emails.Email
HAVING SUM(Compras.Total_Compra) > 500.00;
```

**Algoritmo 16** – Algoritmo da consulta 14 (MYSQL)



The screenshot shows a database interface with a 'Result Grid' tab. The grid has a single column labeled 'Email'. It contains three rows of data: 'joao.silva@example.com', 'pedro.souza@example.com', and 'ana.santos@example.com'. The second row is highlighted with a blue background. Above the grid, there are controls for 'Filter Rows', 'Export', and 'Wrap Cell Content'.

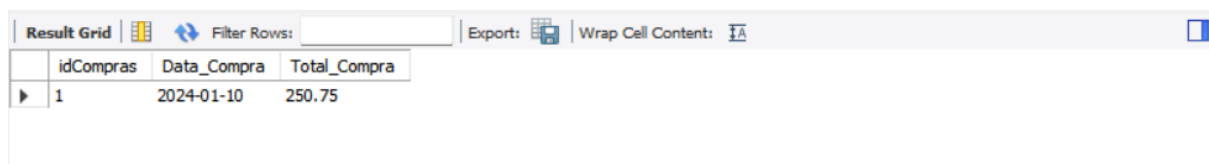
Email
joao.silva@example.com
pedro.souza@example.com
ana.santos@example.com

**Figura 16** – Resultado da consulta 14

Listar todas as compras que incluem um produto especifica-os

```
-- consulta 15
SELECT Compras.idCompras, Compras.Data_Compra, Compras.Total_Compra
FROM Compras
JOIN Itens_Compra ON Compras.idCompras = Itens_Compra.Compras_idCompras
WHERE Itens_Compra.Produtos_idProdutos = 1;
```

**Algoritmo 17** – Algoritmo da consulta 15 (MYSQL)



The screenshot shows a database interface with a 'Result Grid' tab. The grid has three columns: 'idCompras', 'Data\_Compra', and 'Total\_Compra'. It contains one row of data: '1', '2024-01-10', and '250.75'. The first row is highlighted with a blue background. Above the grid, there are controls for 'Filter Rows', 'Export', and 'Wrap Cell Content'.

idCompras	Data_Compra	Total_Compra
1	2024-01-10	250.75

**Figura 17** – Resultado da consulta 15

Listar todos os produtos cujo preço está entre dois outros valores especificados

```
-- consulta 16
SELECT * FROM Produtos
```

```
WHERE Preco_Produtos BETWEEN 100.00 AND 200.00;
```

### Algoritmo 18 – Algoritmo da consulta 16 (MYSQL)

Result Grid					
Filter Rows:					
Edit: Export/Import: Wrap Cell Content:					
	idProdutos	Nome_Produtos	Descricao_Produto	Preco_Produtos	Categoria_idCategoria
▶	1	Pistão	Pistão de alta performance	150.00	1
	2	Disco de Freio	Disco de freio ventilado	120.00	2
	3	Amortecedor	Amortecedor traseiro	200.00	3
*	NULL	NULL	NULL	NULL	NULL

**Figura 18 – Resultado da consulta 16**

Listar todos os produtos que têm “Motor” em sua descrição, adendo que por não haver descrições com o parâmetro especificado não há o retorno de nenhum dado

```
-- consulta 17
SELECT * FROM Produtos
WHERE Descricao_Produto LIKE '%Motor%';
```

### Algoritmo 19 – Algoritmo da consulta 17 (MYSQL)

Result Grid					
Filter Rows:					
Edit: Export/Import: Wrap Cell Content:					
	idProdutos	Nome_Produtos	Descricao_Produto	Preco_Produtos	Categoria_idCategoria
*	NULL	NULL	NULL	NULL	NULL

**Figura 19 – Resultado da consulta 17**

Esse código SQL realiza uma consulta para obter informações sobre compras canceladas feitas por clientes que vivem em uma cidade específica.

```
-- consulta 18
SELECT Compras.idCompras, Clientes.Nome_Clientes, Cidades.Nome_Cidade
FROM Compras
JOIN Clientes ON Compras.Clientes_idClientes = Clientes.idClientes
JOIN Cidades ON Clientes.Cidades_idCidades = Cidades.idCidades
WHERE Compras.Status = 'Cancelada' AND LOWER(Cidades.Nome_Cidade) = LOWER('Salvador');
```

### Algoritmo 20 – Algoritmo da consulta 18 (MYSQL)

Result Grid			Filter Rows:	Export:	Wrap Cell Content:
idCompras	Nome_Clientes	Nome_Cidade			
5	Ana Santos	Salvador			

**Figura 20** – Resultado da consulta 18

Seleciona os nomes distintos dos clientes que compraram produtos de uma categoria específica.

```
-- consulta 19
SELECT DISTINCT Clientes.Nome_Clientes
FROM Clientes
JOIN Compras ON Clientes.idClientes = Compras.Clientes_idClientes
JOIN Itens_Compra ON Compras.idCompras = Itens_Compra.Compras_idCompras
JOIN Produtos ON Itens_Compra.Produtos_idProdutos = Produtos.idProdutos
WHERE Produtos.Categoria_idCategoria = 1;
```

**Algoritmo 21** – Algoritmo da consulta 19 (MYSQL)

Result Grid		Filter Rows:	Export:	Wrap Cell Content:
Nome_Clientes				
João Silva				

**Figura 21** – Resultado da consulta 19

Esse código SQL realiza uma consulta para obter a soma do total gasto em cada compra e organiza os resultados em ordem decrescente do total gasto.

```
-- consulta 20
SELECT Compras.idCompras, SUM(Compras.Total_Compra) AS TotalGasto
FROM Compras
GROUP BY Compras.idCompras
ORDER BY TotalGasto DESC;
```

**Algoritmo 22** – Algoritmo da consulta 20 (MYSQL)

Result Grid		
Filter Rows:		
Export:		
Wrap Cell Content:		
	idCompras	TotalGasto
▶	5	520.80
	7	420.00
	10	420.00
	4	400.99
	2	320.50
	8	300.50
	11	300.50
	1	250.75
	6	180.25
	9	180.25
	12	180.25
	3	150.00

**Figura 22** – Resultado da consulta 20

Calcula a média dos preços dos produtos para cada categoria.

```
-- consulta 21
SELECT Categoria.Nome_categoria, AVG(Produtos.Preco_Produtos) AS PrecoMedio
FROM Categoria
JOIN Produtos ON Categoria.idCategoria = Produtos.Categoria_idCategoria
GROUP BY Categoria.Nome_categoria;
```

**Algoritmo 23** – Algoritmo da consulta 21 (MYSQL)

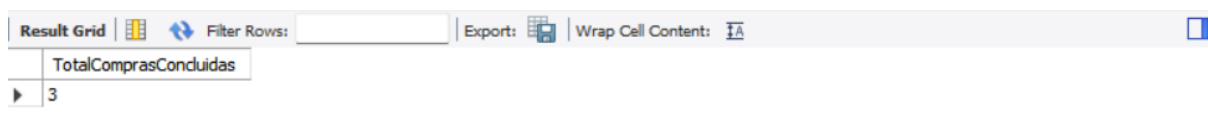
Result Grid		
Filter Rows:		
Export:		
Wrap Cell Content:		
	Nome_categoria	PrecoMedio
▶	Motor	150.000000
	Freios	120.000000
	Suspensão	200.000000
	Transmissão	80.000000
	Acessórios	300.000000

**Figura 23** – Resultado da consulta 21

Conta o número total de compras concluídas em janeiro de 2024.

```
-- consulta 22
SELECT COUNT(*) AS TotalComprasConcluidas
FROM Compras
WHERE Status = 'Concluída' AND MONTH(Data_Compra) = 1 AND YEAR(Data_Compra) = 2024;
```

### Algoritmo 24 – Algoritmo da consulta 22 (MYSQL)



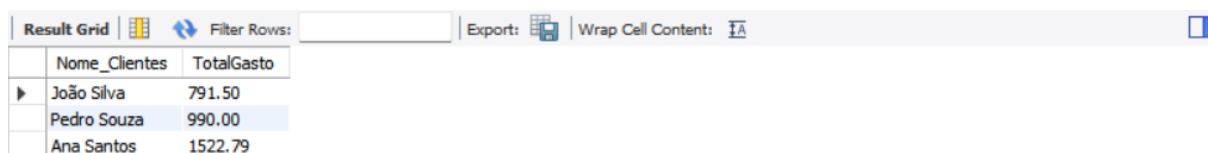
TotalComprasConcluidas
3

**Figura 24 – Resultado da consulta 22**

Listar os clientes que gastaram mais de 500.00 em compras, juntamente com o total gasto por cada cliente.

```
-- consulta 23
SELECT Clientes.Nome_Clientes, SUM(Compras.Total_Compra) AS TotalGasto
FROM Clientes
JOIN Compras ON Clientes.idClientes = Compras.Clientes_idClientes
GROUP BY Clientes.idClientes
HAVING TotalGasto > 500.00;
```

### Algoritmo 25 – Algoritmo da consulta 23 (MYSQL)



Nome_Clientes	TotalGasto
João Silva	791.50
Pedro Souza	990.00
Ana Santos	1522.79

**Figura 25 – Resultado da consulta 23**

**Objetivo:** Selecionar todos os produtos cujo nome começa com a letra 'P'.

```
-- consulta 24
SELECT * FROM Produtos
WHERE Nome_Produtos LIKE 'P%';
```

### Algoritmo 26 – Algoritmo da consulta 24 (MYSQL)

Result Grid				
Filter Rows:				
Edit: Export/Import: Wrap Cell Content: I A				
	idProdutos	Nome_Produtos	Descricao_Produto	Preco_Produtos
▶	1	Pistão	Pistão de alta performance	150.00
*	NULL	NULL	NULL	NULL

**Figura 26** – Resultado da consulta 24

Retorna o nome dos clientes juntamente com o total de cada compra que fizeram.

```
-- consulta 25
SELECT cl.Nome_Clientes, co.Total_Compra
FROM Clientes cl
INNER JOIN Compras co ON cl.idClientes = co.Clientes_idClientes;
```

**Algoritmo 27** – Algoritmo da consulta 25 (MYSQL)

Result Grid		
Filter Rows:		
Export: Wrap Cell Content: I A		
	Nome_Clientes	Total_Compra
▶	João Silva	250.75
	João Silva	180.25
	João Silva	180.25
	João Silva	180.25
	Maria Oliveira	320.50
	Pedro Souza	150.00
	Pedro Souza	420.00
	Pedro Souza	420.00
	Ana Santos	400.99
	Ana Santos	520.80
	Ana Santos	300.50
	Ana Santos	300.50

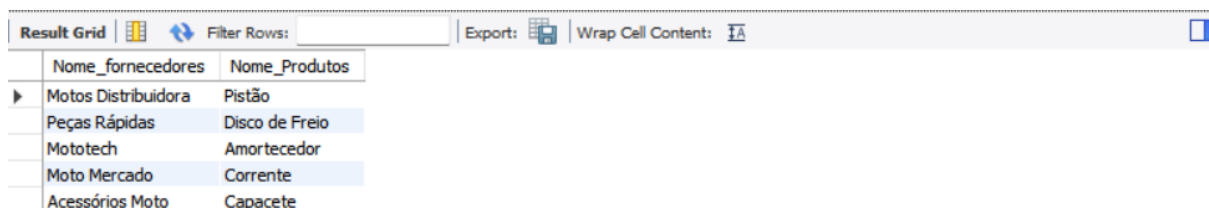
**Figura 27** – Resultado da consulta 25

selecionando o nome dos fornecedores e o nome dos produtos que eles fornecem.

```
-- consulta 26
SELECT F.Nome_fornecedores, P.Nome_Produtos
FROM Fornecedores AS F
```

```
JOIN Fornecedores_E_Produtos AS FP ON F.idFornecedores = FP.Fornecedores_idFornecedores
JOIN Produtos AS P ON FP.Produtos_idProdutos = P.idProdutos;
```

**Algoritmo 28 – Algoritmo da consulta 26 (MYSQL)**



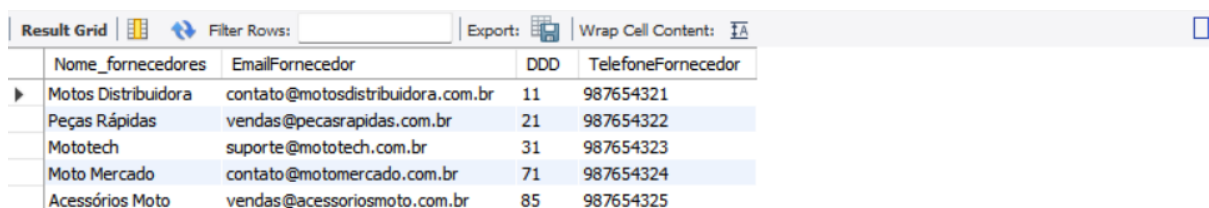
Nome_fornecedores	Nome_Produtos
Motos Distribuidora	Pistão
Peças Rápidas	Disco de Freio
Mototech	Amortecedor
Moto Mercado	Corrente
Acessórios Moto	Capacete

**Figura 28 – Resultado da consulta 26**

consulta retorna os nomes dos fornecedores, seus e-mails e números de telefone correspondentes.

```
-- consulta 27
SELECT F.Nome_fornecedores, EF.EmailFornecedor, TF.DDD, TF.TelefoneFornecedor
FROM Fornecedores AS F
LEFT JOIN EmailFornecedor AS EF ON F.idFornecedores = EF.Fornecedores_idFornecedores
LEFT JOIN TelefoneFornecedor AS TF ON F.idFornecedores = TF.Fornecedores_idFornecedores;
```

**Algoritmo 29 – Algoritmo da consulta 27 (MYSQL)**



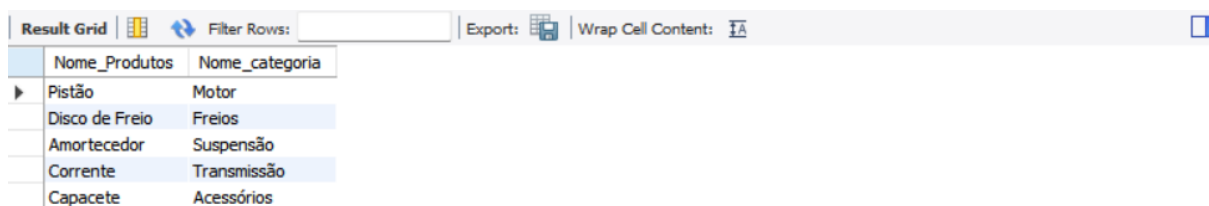
Nome_fornecedores	EmailFornecedor	DDD	TelefoneFornecedor
Motos Distribuidora	contato@motosdistribuidora.com.br	11	987654321
Peças Rápidas	vendas@pecasrapidas.com.br	21	987654322
Mototech	suporte@mototech.com.br	31	987654323
Moto Mercado	contato@motomercado.com.br	71	987654324
Acessórios Moto	vendas@acessoriosmoto.com.br	85	987654325

**Figura 29 – Resultado da consulta 27**

Essa consulta retorna o nome dos produtos e o nome das categorias às quais pertencem.

```
-- consulta 28
SELECT P.Nome_Produtos, C.Nome_categoria
FROM Produtos AS P
JOIN Categoria AS C ON P.Categoria_idCategoria = C.idCategoria;
```

**Algoritmo 30** – Algoritmo da consulta 28 (MYSQL)



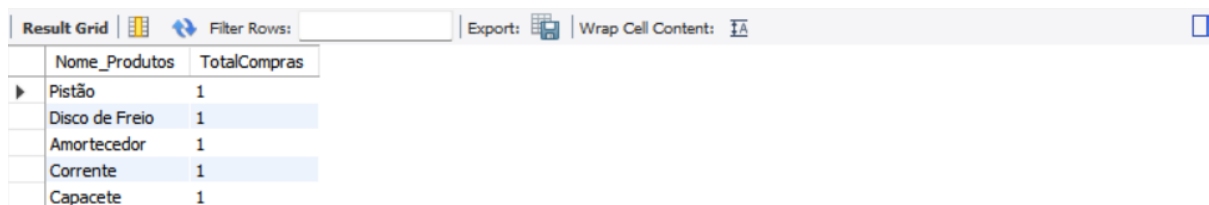
Nome_Produtos	Nome_categoria
Pistão	Motor
Disco de Freio	Freios
Amortecedor	Suspensão
Corrente	Transmissão
Capacete	Acessórios

**Figura 30** – Resultado da consulta 28

Esta consulta lista os produtos mais comprados, mostrando o total de vezes que cada produto foi adquirido.

```
-- consulta 29
SELECT P.Nome_Produtos, COUNT(IC.Produtos_idProdutos) AS TotalCompras
FROM Produtos AS P
LEFT JOIN Itens_Compra AS IC ON P.idProdutos = IC.Produtos_idProdutos
GROUP BY P.Nome_Produtos
ORDER BY TotalCompras DESC;
```

**Algoritmo 31** – Algoritmo da consulta 29 (MYSQL)



Nome_Produtos	TotalCompras
Pistão	1
Disco de Freio	1
Amortecedor	1
Corrente	1
Capacete	1

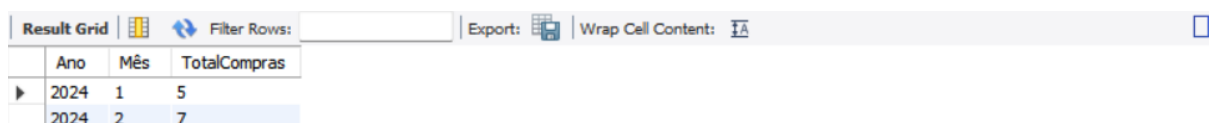
**Figura 31** – Resultado da consulta 29



Essa consulta calcula o total de compras realizadas em cada mês, agrupando os resultados por ano e mês.

```
-- consulta 30
SELECT YEAR(Data_Compra) AS Ano, MONTH(Data_Compra) AS Mês, COUNT(*) AS Total-
Compras
FROM Compras
GROUP BY YEAR(Data_Compra), MONTH(Data_Compra);
```

**Algoritmo 32** – Algoritmo da consulta 30



The screenshot shows a database interface with a 'Result Grid' tab. The grid contains three columns: 'Ano', 'Mês', and 'TotalCompras'. There are two rows of data for the year 2024: month 1 with a total of 5, and month 2 with a total of 7. The interface also includes a 'Filter Rows' field, an 'Export' button, and a 'Wrap Cell Content' checkbox.

Ano	Mês	TotalCompras
2024	1	5
2024	2	7

**Figura 32** – Resultado da consulta 30

## 5 CONCLUSÃO

O estudo sobre sistemas de gerenciamento de banco de dados para uma loja de peças de motos enfatiza a importância crítica de implementar um banco de dados bem elaborado. Essa implementação eficiente é essencial para aprimorar os processos de gerenciamento de estoque, vendas e interação com os clientes. A utilização de tecnologias como o MySQL não apenas assegura a integridade dos dados, mas também agiliza as operações, proporcionando uma experiência mais satisfatória tanto para a equipe quanto para os clientes da loja. O sistema de banco de dados assume um papel indispensável na eficiência operacional e na competitividade do negócio, fornecendo insights valiosos e suporte essencial para o sucesso duradouro da loja de peças de motos.

## REFERÊNCIAS

ROB, Peter; CORONEL, Carlos. **Sistema de Banco de Dados:** Projeto, Implantação e Administração. 1.ed. São Paulo: Cengage Learni

GUEDES, Gilleanes T. A. **UML.2:** Uma Abordagem Pratica. 3.ed. São Paulo: Nova-tec, 2018.

