

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

ANA LÍVIA DA CONCEIÇÃO DE MELO

SISTEMA PARA CONTROLE DE ESTOQUE

**BANCO DE DADOS II
3º SEMESTRE
PROFESSOR PAULO GIOVANI**

**CAMPOS DO JORDÃO
ANO 2024**

RESUMO

O objetivo deste trabalho é desenvolver um sistema de controle de estoque para um canteiro de obras, utilizando modelos conceitual e lógico de banco de dados relacional SQL. A metodologia incluiu a coleta de requisitos por meio de entrevistas com gestores, a criação de um modelo conceitual usando a notação Entidade-Relacionamento, e a elaboração dos modelos lógico e físico com a ferramenta MySQL Workbench. Os resultados abrangem a implementação do banco de dados em SQL para a gestão de estoque. Concluímos que o sistema desenvolvido proporcionou uma gestão eficiente e precisa dos materiais, otimizando os processos do almoxarifado e melhorando a eficiência operacional do canteiro de obras.

Palavras-Chave: Banco de Dados; Controle Estoque; Modelo Lógico; SQL; Gestão de Materiais; MySQL Workbench.

ABSTRACT

The objective of this work is to develop an inventory control system for a construction site, using conceptual and logical SQL relational database models. The methodology included collecting requirements through interviews with managers, creating a conceptual model using Entity-Relationship notation, and developing logical and physical models using the MySQL Workbench tool. The results cover the implementation of the SQL database for inventory management. We concluded that the system developed provided efficient and accurate management of materials, optimizing warehouse processes and improving the operational efficiency of the construction site.

Keywords: Database; Stock Control; Logic Model; SQL; Materials Management; MySQL Workbench.

LISTA DE ILUSTRAÇÕES

FIGURA 1 – Modelo Conceitual	16
FIGURA 2 – Modelo Lógico	16

LISTA DE TABELAS

TABELA 1 – Consulta 1	25
TABELA 2 – Consulta 2	25
TABELA 3 – Consulta 3	26
TABELA 4 – Consulta 4	26
TABELA 5 – Consulta 5	27
TABELA 6 – Consulta 6	27
TABELA 7 – Consulta 7	28
TABELA 8 – Consulta 8	28
TABELA 9 – Consulta 9	28
TABELA 10 – Consulta 10	29
TABELA 11 – Consulta 11	29
TABELA 12 – Consulta 12	29
TABELA 13 – Consulta 13	30
TABELA 14 – Consulta 14	30
TABELA 15 – Consulta 15	30
TABELA 16 – Consulta 16	31
TABELA 17 – Consulta 17	31
TABELA 18 – Consulta 18	31
TABELA 19 – Consulta 19	32
TABELA 20 – Consulta 20	32
TABELA 21 – Consulta 21	32

TABELA 22 – Consulta 22	33
TABELA 23 – Consulta 23	33
TABELA 24 – Consulta 24	34
TABELA 25 – Consulta 25	34
TABELA 26 – Consulta 26	35
TABELA 27 – Consulta 27	35
TABELA 28 – Consulta 28	36
TABELA 29 – Consulta 29	36
TABELA 30 – Consulta 30	37

LISTA DE ALGORITMOS

ALGORITMO 1 – Criação Banco de Dados	18
ALGORITMO 2 – Tabela 1	18
ALGORITMO 3 – Tabela 2	19
ALGORITMO 4 – Inserção 1	19
ALGORITMO 5 – Inserção 2	20
ALGORITMO 6 – Inserção 3	20
ALGORITMO 7 – Inserção 4	21
ALGORITMO 8 – Inserção 5	21
ALGORITMO 9 – Inserção 6	21
ALGORITMO 10 – Inserção 7	22
ALGORITMO 11 – Inserção 8	22
ALGORITMO 12 – Inserção 9	23
ALGORITMO 13 – Inserção 10	23
ALGORITMO 14 – Inserção 11	24
ALGORITMO 15 – Inserção 12	24
ALGORITMO 16 – Consulta 1	24
ALGORITMO 17 – Consulta 2	25
ALGORITMO 18 – Consulta 3	25
ALGORITMO 19 – Consulta 4	26
ALGORITMO 20 – Consulta 5	26
ALGORITMO 21 – Consulta 6	27

ALGORITMO 22 – Consulta 7	28
ALGORITMO 23 – Consulta 8	28
ALGORITMO 24 – Consulta 9	28
ALGORITMO 25 – Consulta 10	29
ALGORITMO 26 – Consulta 11	29
ALGORITMO 27 – Consulta 12	29
ALGORITMO 28 – Consulta 13	30
ALGORITMO 29 – Consulta 14	30
ALGORITMO 30 – Consulta 15	30
ALGORITMO 31 – Consulta 16	31
ALGORITMO 32 – Consulta 17	31
ALGORITMO 33 – Consulta 18	31
ALGORITMO 34 – Consulta 19	31
ALGORITMO 35 – Consulta 20	32
ALGORITMO 36 – Consulta 21	32
ALGORITMO 37 – Consulta 22	33
ALGORITMO 38 – Consulta 23	33
ALGORITMO 39 – Consulta 24	34
ALGORITMO 40 – Consulta 25	34
ALGORITMO 41 – Consulta 26	35
ALGORITMO 42 – Consulta 27	35
ALGORITMO 43 – Consulta 28	35
ALGORITMO 44 – Consulta 29	36
ALGORITMO 45 – Consulta 30	37

LISTA DE SIGLAS

SQL Structured Query Language

SUMÁRIO

1	INTRODUÇÃO	11
1.1	Objetivos	12
1.2	Justificativa	12
1.3	Aspectos Metodológicos	13
1.4	Aporte Teórico	14
2	PROJETO PROPOSTO	14
2.1	Ferramenta Utilizada para Modelagem	15
2.2	Regras de Negócio	16
2.3	Notação Utilizada	16
3	Resultados Obtidos	16
3.1	Modelo Conceitual	16
3.2	Modelo Lógico	17
3.3	Modelo Físico	18
3.4	Consultas SQL	24
4	Conclusão	37
	Referências	38

1. INTRODUÇÃO

A gestão eficiente de estoques é uma tarefa essencial em qualquer projeto de construção civil, onde a disponibilidade adequada de materiais pode impactar diretamente no andamento e sucesso das obras. O controle rigoroso de entrada e saída de materiais é crucial para evitar desperdícios, assegurar a qualidade e garantir que os recursos estejam disponíveis no momento certo. Neste contexto, o uso de sistemas informatizados para a gestão de estoques tem se tornado uma prática comum, trazendo maior precisão e agilidade aos processos.

Este relatório apresenta o desenvolvimento e implementação de um sistema de gestão de estoques para um almoxarifado de obra, utilizando um banco de dados relacional SQL. O sistema foi concebido para atender às necessidades específicas do setor de construção, proporcionando um controle detalhado sobre os materiais, desde o recebimento até a utilização no canteiro de obras.

O desenvolvimento do projeto seguiu uma metodologia estruturada, passando pela modelagem conceitual, lógica e física do banco de dados, além da implementação de consultas SQL que permitem a extração de informações relevantes para a gestão do estoque. O objetivo principal foi criar um sistema robusto e eficiente, capaz de suportar as operações diárias de um almoxarifado de obra, oferecendo também funcionalidades para a geração de relatórios gerenciais.

Ao longo deste relatório, serão detalhadas as etapas do projeto, desde as considerações iniciais e coleta de requisitos até a implementação final do banco de dados. Serão apresentados os modelos conceitual, lógico e físico, juntamente com exemplos de consultas SQL que podem ser utilizadas para a extração e análise dos dados. Por fim, serão discutidos os resultados obtidos e propostas sugestões de melhorias futuras para o sistema desenvolvido.

1.1 Objetivos

O objetivo deste relatório é documentar o desenvolvimento e a implementação de um sistema de gestão de estoques para um almoxarifado de obra, utilizando um banco de dados relacional SQL. O sistema visa proporcionar um controle eficiente e preciso dos materiais, desde o seu recebimento até a utilização no canteiro de obras, atendendo às necessidades específicas do setor de construção civil.

Este relatório busca fornecer uma visão completa do processo de desenvolvimento do sistema de gestão de estoques, demonstrando sua importância para a eficiência operacional no setor de construção.

1.2 Justificativa

A gestão eficaz de estoques é fundamental para o sucesso de qualquer projeto de construção civil. Um controle inadequado dos materiais pode levar a diversos problemas, como a falta de insumos essenciais, atrasos na obra, aumento de custos devido à compra emergencial de materiais e desperdício por má conservação ou perda de validade. A implementação de um sistema de gestão de estoques informatizado é, portanto, uma necessidade imperativa para otimizar recursos, garantir a continuidade do trabalho e minimizar perdas financeiras.

A escolha de um banco de dados relacional SQL para a construção deste sistema se justifica por: Bancos de dados relacionais garantem a integridade referencial dos dados, evitando problemas como duplicidade e inconsistências. Isso é crucial para a gestão de estoques, onde a precisão das informações é vital.

A linguagem SQL permite a criação de consultas complexas e a extração de dados de forma eficiente, facilitando a análise de informações e a tomada de decisões baseadas em dados concretos. Sistemas baseados em SQL são altamente escaláveis, permitindo o crescimento do banco de dados conforme as necessidades do projeto aumentam, sem comprometer o desempenho. Oferecem mecanismos robustos

de segurança, incluindo controle de acesso, autenticação e criptografia, garantindo que as informações sensíveis do almoxarifado estejam protegidas.

Ferramentas como MySQL possuem ampla documentação, suporte comunitário e uma vasta gama de recursos adicionais, facilitando o desenvolvimento e a manutenção do sistema.

Além disso, a informatização do controle de estoques no canteiro de obras traz benefícios como a redução de desperdício, com um controle mais preciso dos materiais, é possível evitar compras desnecessárias e reduzir o desperdício por deterioração. Um sistema automatizado acelera o processo de entrada e saída de materiais, reduzindo o tempo e o esforço necessário para essas operações.

Portanto, a implementação de um sistema de gestão de estoques baseado em SQL não só atende a uma necessidade prática do setor de construção civil, mas também proporciona uma série de benefícios que justificam o investimento em tecnologia e desenvolvimento de sistemas.

1.3 Aspectos Metodológicos

O presente estudo fez uso de pesquisas de natureza bibliográfica, que abarcaram a parte escrita, e de campo, que se referem à parte prática. Os aspectos metodológicos do desenvolvimento do sistema de gestão de estoques para almoxarifado em obra incluem diversas etapas: a coleta de requisitos, que envolveu a identificação das necessidades dos usuários e a definição dos requisitos funcionais e não funcionais do sistema; a modelagem conceitual, onde foram definidas as entidades e seus relacionamentos; a modelagem lógica, que consistiu na conversão do modelo conceitual em um esquema de banco de dados relacional; e a modelagem física, que abrangeu a implementação do modelo lógico no banco de dados MySQL, incluindo a criação das tabelas e a inserção de dados para testes. O desenvolvimento de consultas SQL foi crucial para permitir a extração de informações relevantes para a gestão do estoque. A validação do sistema envolveu testes e verificações para garantir que o sistema

atenda aos requisitos definidos e seja capaz de operar de forma eficiente no ambiente de um canteiro de obras. A documentação detalhou todas as etapas do desenvolvimento, incluindo as ferramentas utilizadas, os modelos de dados, as consultas SQL elaboradas e os resultados obtidos.

1.4 Aporte Teórico

A contribuição dos autores Abraham Silberschatz, Carlos Coronel, Raghu Ramakrishnan, Johannes Gehrke e Peter Rob é de suma importância para o desenvolvimento do projeto de sistemas de banco de dados, especialmente no que se refere ao modelo conceitual utilizado para o controle de estoque.

Ao integrar os insights e conhecimentos desses autores, o projeto de controle de estoque se beneficia de uma base teórica robusta e de técnicas comprovadas no desenvolvimento de sistemas de banco de dados. Essa abordagem holística e interdisciplinar é essencial para garantir a eficácia e o sucesso do projeto, desde a concepção até a implementação e operação do sistema de controle de estoque.

2. PROJETO PROPOSTO

O projeto proposto visa desenvolver um sistema de controle de estoque baseado em um modelo relacional de banco de dados, customizado para atender às demandas específicas de um canteiro de obras na indústria da construção civil. A escolha de um modelo relacional de banco de dados é fundamentada em sua capacidade de representar de forma eficiente as relações entre os diversos elementos do sistema, como materiais, fornecedores, pedidos e movimentações de estoque.

Além disso, a utilização da linguagem SQL (Structured Query Language) como parte integrante do sistema de controle de estoque é essencial para a realização de

consultas e operações de manipulação de dados de forma rápida e eficiente. O SQL oferece uma ampla gama de recursos que facilitam a interação com o banco de dados relacional, permitindo a execução de consultas complexas, inserções, atualizações e exclusões de registros conforme necessário.

A metodologia de modelo conceitual, aliada ao levantamento de requisitos, proporciona uma abordagem sistemática e organizada para o desenvolvimento do sistema de controle de estoque. Ao definir claramente os requisitos do sistema e criar um modelo conceitual que represente de forma precisa as entidades e relacionamentos envolvidos, é possível garantir que o sistema atenda às necessidades dos usuários finais e que seja desenvolvido de forma eficaz e eficiente. A metodologia escolhida é eficaz porque permite uma melhor compreensão das necessidades do usuário e dos requisitos do sistema, o que reduz o risco de erros e retrabalho durante o desenvolvimento. Isso ocorre porque o modelo conceitual serve como um guia claro e conciso para a implementação do sistema, garantindo que todas as funcionalidades necessárias sejam contempladas de forma adequada.

A combinação de um modelo relacional de banco de dados, a linguagem SQL e a metodologia de modelo conceitual proporcionam uma base sólida e eficaz para o desenvolvimento do sistema de controle de estoque, garantindo sua eficiência, usabilidade e adaptabilidade às necessidades específicas do ambiente de um canteiro de obras na indústria da construção civil.

2.1 Ferramenta Utilizada para Modelagem e Seus Requisitos

Para o Modelo Conceitual foi utilizada a ferramenta Draw.io, que pode ser acessada de forma on-line apenas criando uma conta.

Para a etapa de modelagem, foi utilizada a ferramenta MySQL Workbench, que permite a criação e a manipulação de diagramas de entidade-relacionamento (DER). Foi utilizada essa ferramenta para criar o modelo lógico e o modelo físico do projeto.

Os requisitos para a instalação do MySQL Workbench incluem:

Sistema Operacional: Windows, macOS ou Linux.

MySQL Server instalado.

Processador: Intel ou AMD de 64 bits.
Memória RAM: 4 GB (recomendado).
Espaço em disco: 200 MB para instalação.

2. 2 Regras de Negócio

O projeto de dados foi elaborado com base nas regras de negócio coletadas através de entrevistas com stakeholders e análise de processos de negócios existentes. As principais regras de negócio identificadas foram:

Cada material deve ter um cadastro único com informações detalhadas, como nome, descrição, unidade de medida e quantidade disponível.

As entradas de estoque devem registrar a quantidade de material que chega ao canteiro de obras, juntamente com a data e o fornecedor responsável.

As saídas de estoque devem registrar a quantidade de material que é retirada para uso, juntamente com a data e o destino.

O sistema deve permitir a consulta de estoque disponível e a geração de relatórios gerenciais.

Cada fornecedor deve ter um cadastro único com informações de contato.

Os pedidos de materiais devem ser registrados, vinculando os fornecedores aos materiais solicitados.

2.3 Notação Utilizada para o Modelo Conceitual

No desenvolvimento do modelo conceitual, foi utilizada a notação Entidade-Relacionamento (ER) proposta por Peter Chen. A notação ER é utilizada para a modelagem de dados e oferece uma representação clara e precisa das entidades, atributos e relacionamentos dentro de um sistema de banco de dados. Essa notação facilita o entendimento e a comunicação entre os desenvolvedores e outros stakeholders do projeto, garantindo que todos tenham uma visão comum e detalhada do modelo de dados.

3. RESULTADOS OBTIDOS

3.1 Modelo Conceitual

O modelo conceitual feito a tabela 1, foi desenvolvido com as etapas de Fornecedor, Pedido, EntradaEstoque, Material e SaídaEstoque, nesse modelo precisa ter o número do pedido para efetuar um pedido com os fornecedores, que a partir do pedido vão entregar e dar entrada no estoque na tabela EntradaEstoque, que vai armazenar

e dar a saída de materiais na tabela SaídaEstoque. Tendo assim um controle de entrada, armazenamento e saída do estoque.

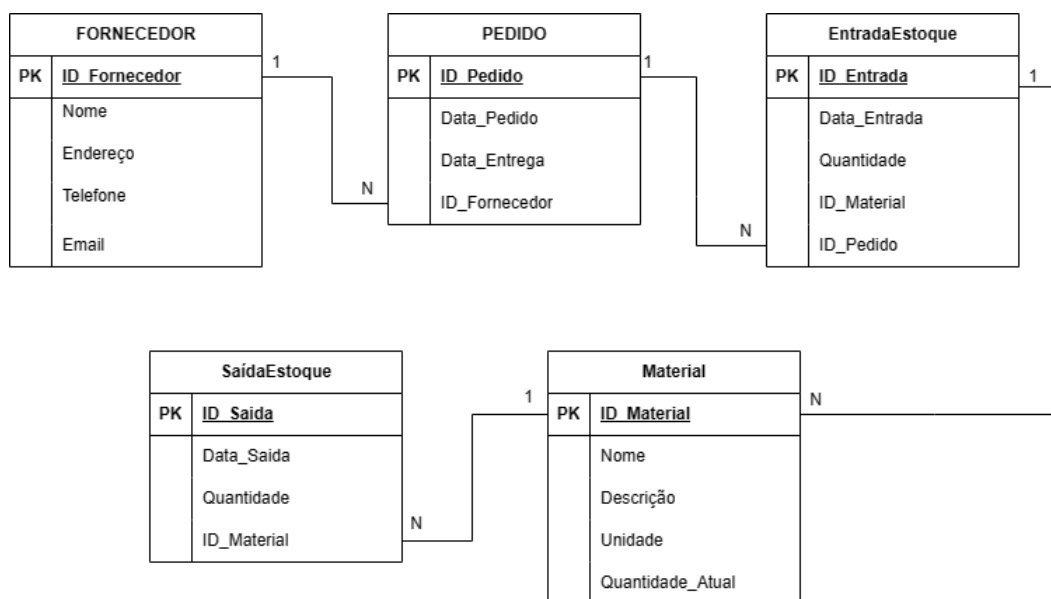


Figura 1 – Modelo Conceitual (Autor)

3.2 Modelo Lógico

O modelo lógico feito na tabela 2, foi projetado para deixar a estrutura do banco de dados de forma organizada e deixando visível seus requisitos específicos para uma aplicação, determinando como os dados serão organizados e relacionados entre si dentro do sistema de banco de dados.

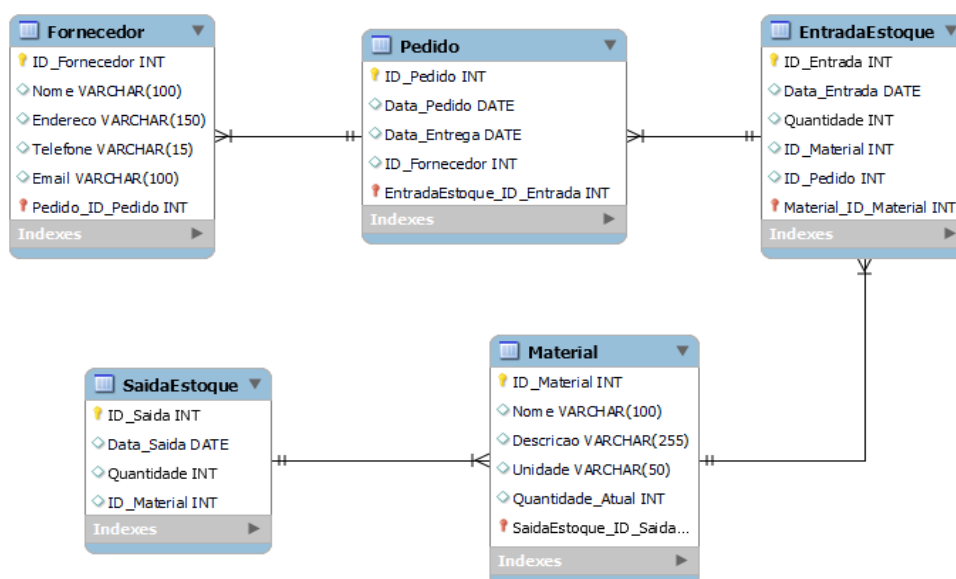


Figura 2 – Modelo Lógico (Autor)

3.3 Modelo Físico

Criando o Banco de Dados:

```
1 • CREATE DATABASE ControleEstoque;
2
3 • USE ControleEstoque;
```

Algoritmo 1 - Criação Banco de Dados (Autor)

Criando as tabelas Fornecedor, Material, Pedido, EntradaEstoque e SaídaEstoque:

```
5 • CREATE TABLE Fornecedor (
6     ID_Fornecedor INT PRIMARY KEY AUTO_INCREMENT,
7     Nome VARCHAR(100) NOT NULL,
8     Endereco VARCHAR(150),
9     Telefone VARCHAR(15),
10    Email VARCHAR(100)
11 );
12
13 • CREATE TABLE Material (
14     ID_Material INT PRIMARY KEY AUTO_INCREMENT,
15     Nome VARCHAR(100) NOT NULL,
16     Descricao VARCHAR(255),
17     Unidade VARCHAR(50),
18     Quantidade_Atual INT DEFAULT 0
19 );
20
21 • CREATE TABLE Pedido (
22     ID_Pedido INT PRIMARY KEY AUTO_INCREMENT,
23     Data_Pedido DATE NOT NULL,
24     Data_Entrega DATE,
25     ID_Fornecedor INT,
26     FOREIGN KEY (ID_Fornecedor) REFERENCES Fornecedor(ID_Fornecedor)
27 );
```

Algoritmo 2 – Tabela 1 (Autor)

```

29 ● CREATE TABLE EntradaEstoque (
30     ID_Entrada INT PRIMARY KEY AUTO_INCREMENT,
31     Data_Entrada DATE NOT NULL,
32     Quantidade INT NOT NULL,
33     ID_Material INT,
34     ID_Pedido INT,
35     FOREIGN KEY (ID_Material) REFERENCES Material(ID_Material),
36     FOREIGN KEY (ID_Pedido) REFERENCES Pedido(ID_Pedido)
37 );
38
39 ● CREATE TABLE SaidaEstoque (
40     ID_Saida INT PRIMARY KEY AUTO_INCREMENT,
41     Data_Saida DATE NOT NULL,
42     Quantidade INT NOT NULL,
43     ID_Material INT,
44     FOREIGN KEY (ID_Material) REFERENCES Material(ID_Material)
45 );

```

Algoritmo 3 – Tabela 2 (Autor)

Inserindo fornecedores na tabela Fornecedor:

```

49 ● INSERT INTO Fornecedor (ID_Fornecedor, Nome, Endereco, Telefone, Email)
50 VALUES (1, 'Ferragens', 'Rua das Flores, 123, Jardins', '1234-5678', 'contato@ferragens.com');
51
52 ● INSERT INTO Fornecedor (ID_Fornecedor, Nome, Endereco, Telefone, Email)
53 VALUES (2, 'Ferramentas', 'Avenida Central, 456, Jardins', '9876-5432', 'vendas@ferramentas.com');
54
55 ● INSERT INTO Fornecedor (ID_Fornecedor, Nome, Endereco, Telefone, Email)
56 VALUES (3, 'Gauchão Materiais', 'Rua Nova, 789, Jardins', '1357-2468', 'info@materiais.com');
57
58 ● INSERT INTO Fornecedor (ID_Fornecedor, Nome, Endereco, Telefone, Email)
59 VALUES (4, 'Deposito Construção', 'Rua Principal, 101, Jardins', '2468-1357', 'atendimento@deposito.com');
60
61 ● INSERT INTO Fornecedor (ID_Fornecedor, Nome, Endereco, Telefone, Email)
62 VALUES (5, 'Fornecendo Sua Casa', 'Avenida Secundária, 202, São Paulo', '3698-7451', 'compras@fornecedore.com');

```

Algoritmo 4 - Inserção 1 (Autor)

Inserindo materiais na tabela Material:

```

69 • INSERT INTO Material (ID_Material, Nome, Descricao, Unidade, Quantidade_Atual)
70   VALUES (1, 'Cimento', 'Cimento Portland tipo I', 'saco', 100);
71
72 • INSERT INTO Material (ID_Material, Nome, Descricao, Unidade, Quantidade_Atual)
73   VALUES (2, 'Areia', 'Areia fina para construção', 'metro cúbico', 50);
74
75 • INSERT INTO Material (ID_Material, Nome, Descricao, Unidade, Quantidade_Atual)
76   VALUES (3, 'Brita', 'Brita 1 para construção', 'metro cúbico', 30);
77
78 • INSERT INTO Material (ID_Material, Nome, Descricao, Unidade, Quantidade_Atual)
79   VALUES (4, 'Tijolo', 'Tijolo cerâmico 8 furos', 'unidade', 5000);
80
81 • INSERT INTO Material (ID_Material, Nome, Descricao, Unidade, Quantidade_Atual)
82   VALUES (5, 'Bloco de Concreto', 'Bloco de concreto 14x19x39', 'unidade', 2000);
83
84 • INSERT INTO Material (ID_Material, Nome, Descricao, Unidade, Quantidade_Atual)
85   VALUES (6, 'Aço CA-50', 'Barra de aço CA-50 12m', 'barra', 150);
86
87 • INSERT INTO Material (ID_Material, Nome, Descricao, Unidade, Quantidade_Atual)
88   VALUES (7, 'Cal Hidratada', 'Cal hidratada CH-III', 'saco', 80);
89
90 • INSERT INTO Material (ID_Material, Nome, Descricao, Unidade, Quantidade_Atual)
91   VALUES (8, 'Tubos PVC', 'Tubo PVC 100mm para esgoto', 'metro', 100);

```

Algoritmo 5 - Inserção 2 (Autor)

```

93 • INSERT INTO Material (ID_Material, Nome, Descricao, Unidade, Quantidade_Atual)
94   VALUES (9, 'Telha Cerâmica', 'Telha cerâmica colonial', 'unidade', 1200);
95
96 • INSERT INTO Material (ID_Material, Nome, Descricao, Unidade, Quantidade_Atual)
97   VALUES (10, 'Argamassa', 'Argamassa AC-II para revestimento', 'saco', 200);
98
99 • INSERT INTO Material (Nome, Descricao, Unidade, Quantidade_Atual)
100  VALUES ('Tinta Acrílica', 'Tinta acrílica para paredes', 'litro', 200);
101
102 • INSERT INTO Material (Nome, Descricao, Unidade, Quantidade_Atual)
103  VALUES ('Fio Elétrico', 'Fio elétrico 2,5mm', 'metro', 1000);
104
105 • INSERT INTO Material (Nome, Descricao, Unidade, Quantidade_Atual)
106  VALUES ('Canos de Cobre', 'Canos de cobre 1/2 polegada', 'metro', 500);
107
108 • INSERT INTO Material (Nome, Descricao, Unidade, Quantidade_Atual)
109  VALUES ('Parafuso', 'Parafuso 3cm', 'unidade', 10000);
110
111 • INSERT INTO Material (Nome, Descricao, Unidade, Quantidade_Atual)
112  VALUES ('Porta de Madeira', 'Porta de madeira 2,10 x 0,80', 'unidade', 50);
113
114 • INSERT INTO Material (Nome, Descricao, Unidade, Quantidade_Atual)
115  VALUES ('Janela de Alumínio', 'Janela de alumínio 1,20 x 1,00', 'unidade', 30);

```

Algoritmo 6 - Inserção 3 (Autor)

```

117 • INSERT INTO Material (Nome, Descricao, Unidade, Quantidade_Atual)
118 VALUES ('Piso Cerâmico', 'Piso cerâmico 60x60cm', 'caixa', 150);
119
120 • INSERT INTO Material (Nome, Descricao, Unidade, Quantidade_Atual)
121 VALUES ('Interruptor', 'Interruptor simples', 'unidade', 500);
122
123 • INSERT INTO Material (Nome, Descricao, Unidade, Quantidade_Atual)
124 VALUES ('Lâmpada LED', 'Lâmpada LED 10W', 'unidade', 300);
125
126 • INSERT INTO Material (Nome, Descricao, Unidade, Quantidade_Atual)
127 VALUES ('Fita Isolante', 'Fita isolante 19mm', 'rolo', 200);

```

Algoritmo 7 - Inserção 4 (Autor)

Inserindo pedidos na tabela Pedido, associando cada pedido a um fornecedor:

```

133 • INSERT INTO Pedido (ID_Pedido, ID_Fornecedor, Data_Pedido)
134 VALUES (1, 1, '2024-05-01');
135
136 • INSERT INTO Pedido (ID_Pedido, ID_Fornecedor, Data_Pedido)
137 VALUES (2, 2, '2024-05-02');
138
139 • INSERT INTO Pedido (ID_Pedido, ID_Fornecedor, Data_Pedido)
140 VALUES (3, 3, '2024-05-03');
141
142 • INSERT INTO Pedido (ID_Pedido, ID_Fornecedor, Data_Pedido)
143 VALUES (4, 4, '2024-05-04');
144
145 • INSERT INTO Pedido (ID_Pedido, ID_Fornecedor, Data_Pedido)
146 VALUES (5, 5, '2024-05-05');

```

Algoritmo 8 – Inserção 5 (Autor)

```

180 • INSERT INTO Pedido (ID_Fornecedor, Data_Pedido)
181 VALUES (1, '2024-06-01');
182
183 • INSERT INTO Pedido (ID_Fornecedor, Data_Pedido)
184 VALUES (2, '2024-06-02');
185
186 • INSERT INTO Pedido (ID_Fornecedor, Data_Pedido)
187 VALUES (3, '2024-06-03');
188
189 • INSERT INTO Pedido (ID_Fornecedor, Data_Pedido)
190 VALUES (4, '2024-06-04');
191
192 • INSERT INTO Pedido (ID_Fornecedor, Data_Pedido)
193 VALUES (5, '2024-06-05');

```

Algoritmo 9 - Inserção 6 (Autor)

Inserindo entradas no estoque na tabela EntradaEstoque, associando cada entrada a um pedido e material:

```

149 • INSERT INTO EntradaEstoque (ID_Entrada, ID_Pedido, ID_Material, Quantidade, Data_Entrada)
150 VALUES (1, 1, 1, 100, '2024-05-06'); -- Cimento de Ferragens
151
152 • INSERT INTO EntradaEstoque (ID_Entrada, ID_Pedido, ID_Material, Quantidade, Data_Entrada)
153 VALUES (2, 2, 2, 50, '2024-05-07'); -- Areia de Ferramentas
154
155 • INSERT INTO EntradaEstoque (ID_Entrada, ID_Pedido, ID_Material, Quantidade, Data_Entrada)
156 VALUES (3, 3, 3, 30, '2024-05-08'); -- Brita de Gauchao Materiais
157
158 • INSERT INTO EntradaEstoque (ID_Entrada, ID_Pedido, ID_Material, Quantidade, Data_Entrada)
159 VALUES (4, 4, 4, 5000, '2024-05-09'); -- Tijolo de Deposito Construcão
160
161 • INSERT INTO EntradaEstoque (ID_Entrada, ID_Pedido, ID_Material, Quantidade, Data_Entrada)
162 VALUES (5, 5, 5, 2000, '2024-05-10'); -- Bloco de Concreto de Fornecendo Sua Casa
163
164 • INSERT INTO EntradaEstoque (ID_Entrada, ID_Pedido, ID_Material, Quantidade, Data_Entrada)
165 VALUES (6, 1, 6, 150, '2024-05-11'); -- Aço CA-50 de Ferragens
166
167 • INSERT INTO EntradaEstoque (ID_Entrada, ID_Pedido, ID_Material, Quantidade, Data_Entrada)
168 VALUES (7, 2, 7, 80, '2024-05-12'); -- Cal Hidratada de Ferramentas
169
170 • INSERT INTO EntradaEstoque (ID_Entrada, ID_Pedido, ID_Material, Quantidade, Data_Entrada)
171 VALUES (8, 3, 8, 100, '2024-05-13'); -- Tubos PVC de Gauchao Materiais

```

Algoritmo 10 - Inserção 7 (Autor)

```

173 • INSERT INTO EntradaEstoque (ID_Entrada, ID_Pedido, ID_Material, Quantidade, Data_Entrada)
174 VALUES (9, 4, 9, 1200, '2024-05-14'); -- Telha Cerâmica de Deposito Construcão
175
176 • INSERT INTO EntradaEstoque (ID_Entrada, ID_Pedido, ID_Material, Quantidade, Data_Entrada)
177 VALUES (10, 5, 10, 200, '2024-05-15'); -- Argamassa de Fornecendo Sua Casa

```

Algoritmo 11 - Inserção 8 (Autor)

```

196 • INSERT INTO EntradaEstoque (ID_Pedido, ID_Material, Quantidade, Data_Entrada)
197 VALUES (1, 11, 200, '2024-06-06'); -- Tinta Acrílica de Ferragens
198
199 • INSERT INTO EntradaEstoque (ID_Pedido, ID_Material, Quantidade, Data_Entrada)
200 VALUES (2, 12, 1000, '2024-06-07'); -- Fio Elétrico de Ferramentas
201
202 • INSERT INTO EntradaEstoque (ID_Pedido, ID_Material, Quantidade, Data_Entrada)
203 VALUES (3, 13, 500, '2024-06-08'); -- Canos de Cobre de Gauchão Materiais
204
205 • INSERT INTO EntradaEstoque (ID_Pedido, ID_Material, Quantidade, Data_Entrada)
206 VALUES (4, 14, 10000, '2024-06-09'); -- Parafuso de Depósito Construção
207
208 • INSERT INTO EntradaEstoque (ID_Pedido, ID_Material, Quantidade, Data_Entrada)
209 VALUES (5, 15, 50, '2024-06-10'); -- Porta de Madeira de Fornecendo Sua Casa
210
211 • INSERT INTO EntradaEstoque (ID_Pedido, ID_Material, Quantidade, Data_Entrada)
212 VALUES (1, 16, 30, '2024-06-11'); -- Janela de Alumínio de Ferragens
213
214 • INSERT INTO EntradaEstoque (ID_Pedido, ID_Material, Quantidade, Data_Entrada)
215 VALUES (2, 17, 150, '2024-06-12'); -- Piso Cerâmico de Ferramentas
216
217 • INSERT INTO EntradaEstoque (ID_Pedido, ID_Material, Quantidade, Data_Entrada)
218 VALUES (3, 18, 500, '2024-06-13'); -- Interruptor de Gauchão Materiais

```

Algoritmo 12 - Inserção 9 (Autor).

```

220 • INSERT INTO EntradaEstoque (ID_Pedido, ID_Material, Quantidade, Data_Entrada)
221 VALUES (4, 19, 300, '2024-06-14'); -- Lâmpada LED de Depósito Construção
222
223 • INSERT INTO EntradaEstoque (ID_Pedido, ID_Material, Quantidade, Data_Entrada)
224 VALUES (5, 20, 200, '2024-06-15'); -- Fita Isolante de Fornecendo Sua Casa

```

Algoritmo 13 - Inserção 10 (Autor).

Inserindo as saídas no estoque na tabela Saída:

```

231 • INSERT INTO SaidaEstoque (ID_Saida, Data_Saida, Quantidade, ID_Material)
232 VALUES (1, '2024-06-01', 20, 1); -- Saída de 20 unidades de Cimento
233
234 • INSERT INTO SaidaEstoque (ID_Saida, Data_Saida, Quantidade, ID_Material)
235 VALUES (2, '2024-06-02', 10, 2); -- Saída de 10 metros cúbicos de Areia
236
237 • INSERT INTO SaidaEstoque (ID_Saida, Data_Saida, Quantidade, ID_Material)
238 VALUES (3, '2024-06-03', 5, 3); -- Saída de 5 metros cúbicos de Brita
239
240 • INSERT INTO SaidaEstoque (ID_Saida, Data_Saida, Quantidade, ID_Material)
241 VALUES (4, '2024-06-04', 1000, 4); -- Saída de 1000 unidades de Tijolo
242
243 • INSERT INTO SaidaEstoque (ID_Saida, Data_Saida, Quantidade, ID_Material)
244 VALUES (5, '2024-06-05', 500, 5); -- Saída de 500 unidades de Bloco de Concreto
245
246 • INSERT INTO SaidaEstoque (ID_Saida, Data_Saida, Quantidade, ID_Material)
247 VALUES (6, '2024-06-06', 30, 6); -- Saída de 30 barras de Aço CA-50
248
249 • INSERT INTO SaidaEstoque (ID_Saida, Data_Saida, Quantidade, ID_Material)
250 VALUES (7, '2024-06-07', 10, 7); -- Saída de 10 sacos de Cal Hidratada
251
252 • INSERT INTO SaidaEstoque (ID_Saida, Data_Saida, Quantidade, ID_Material)
253 VALUES (8, '2024-06-08', 20, 8); -- Saída de 20 metros de Tubos PVC

```

Algoritmo 14 - Inserção 11 (Autor).

```

255 • INSERT INTO SaidaEstoque (ID_Saida, Data_Saida, Quantidade, ID_Material)
256 VALUES (9, '2024-06-09', 200, 9); -- Saída de 200 unidades de Telha Cerâmica
257
258 • INSERT INTO SaidaEstoque (ID_Saida, Data_Saida, Quantidade, ID_Material)
259 VALUES (10, '2024-06-10', 50, 10); -- Saída de 50 sacos de Argamassa

```

Algoritmo 15 - Inserção 12 (Autor).

3.4 Consultas SQL

1) Listar os fornecedores e os materiais fornecidos por cada um:

```

1 • SELECT f.Nome AS Fornecedor, m.Nome AS Material
2 FROM Fornecedor f
3 JOIN Pedido p ON f.ID_Fornecedor = p.ID_Fornecedor
4 JOIN EntradaEstoque e ON p.ID_Pedido = e.ID_Pedido
5 JOIN Material m ON e.ID_Material = m.ID_Material
6 ORDER BY f.Nome, m.Nome;

```

Algoritmo 16 – Consulta 1 (Autor)

	Fornecedor	Material
►	Deposito Construção	Lâmpada LED
	Deposito Construção	Parafuso
	Deposito Construção	Telha Cerâmica
	Deposito Construção	Tijolo
	Ferragens	Aço CA-50
	Ferragens	Cimento
	Ferragens	Janela de Alumínio
	Ferragens	Tinta Acrílica
	Ferramentas	Areia
	Ferramentas	Cal Hidratada
	Ferramentas	Fio Elétrico
	Ferramentas	Piso Cerâmico
	Fornecendo Sua Casa	Argamassa

Tabela 1 – Consulta 1 (Autor)

2) Calcular a quantidade total de materiais recebidos por mês em 2024:

```

9 • SELECT MONTH(Data_Entrada) AS Mes, SUM(Quantidade) AS QuantidadeTotal
10 FROM EntradaEstoque
11 WHERE YEAR(Data_Entrada) = 2024
12 GROUP BY MONTH(Data_Entrada);
--

```

Algoritmo 17 – Consulta 2 (Autor)

	Mes	QuantidadeTotal
►	5	8910
	6	12930

Tabela 2 - Consulta 2 (Autor)

3) Contar quantos pedidos cada fornecedor recebeu:

```

18 • SELECT f.Nome, COUNT(p.ID_Pedido) AS TotalPedidos
19 FROM Fornecedor f
20 JOIN Pedido p ON f.ID_Fornecedor = p.ID_Fornecedor
21 GROUP BY f.Nome;

```

Algoritmo 18 – Consulta 3 (Autor)

	Nome	TotalPedidos
►	Ferragens	1
	Ferramentas	1
	Gauchaão Materiais	1
	Deposito Construção	1
	Fornecendo Sua Casa	1

Tabela 3 – Consulta 3 (Autor)

4) Listar os materiais e suas respectivas entradas no estoque:

```

23 • SELECT m.Nome, e.Quantidade, e.Data_Entrada
24 FROM Material m
25 JOIN EntradaEstoque e ON m.ID_Material = e.ID_Material;

```

Algoritmo 19 – Consulta 4 (Autor)

	Nome	Quantidade	Data_Entrada
►	Cimento	100	2024-05-06
	Brita	30	2024-05-08
	Areia	50	2024-05-07
	Tijolo	5000	2024-05-09
	Bloco de Concreto	2000	2024-05-10
	Aço CA-50	150	2024-05-11
	Cal Hidratada	80	2024-05-12
	Tubos PVC	100	2024-05-13
	Telha Cerâmica	1200	2024-05-14
	Argamassa	200	2024-05-15
	Tinta Acrílica	200	2024-06-06
	Fio Elétrico	1000	2024-06-07
	Canos de Cobre	500	2024-06-08
	Parafuso	10000	2024-06-09

Tabela 4 – Consulta 4 (Autor)

5) Listar os materiais ordenados pela quantidade atual em estoque, do maior para o menor:

```

27 • SELECT *
28 FROM Material
29 ORDER BY Quantidade_Atual DESC;

```

Algoritmo 20 – Consulta 5 (Autor)

	ID_Material	Nome	Descricao	Unidade	Quantidade_Atual
▶	14	Parafuso	Parafuso 3cm	unidade	10000
	4	Tijolo	Tijolo cerâmico 8 furos	unidade	5000
	5	Bloco de Concreto	Bloco de concreto 14x19x39	unidade	2000
	9	Telha Cerâmica	Telha cerâmica colonial	unidade	1200
	12	Fio Elétrico	Fio elétrico 2,5mm	metro	1000
	13	Canos de Cobre	Canos de cobre 1/2 polegada	metro	500
	18	Interruptor	Interruptor simples	unidade	500
	19	Lâmpada LED	Lâmpada LED 10W	unidade	300
	10	Argamassa	Argamassa AC-II para revestimento	saco	200
	11	Tinta Acrílica	Tinta acrílica para paredes	litro	200
	20	Fita Isolante	Fita isolante 19mm	rolo	200
	6	Aço CA-50	Barra de aço CA-50 12m	barra	150
	17	Piso Cerâmico	Piso cerâmico 60x60cm	caixa	150
	1	Cimento	Cimento Portland tipo I	saco	100
	8	Tubos PVC	Tubo PVC 100mm para esgoto	metro	100
	7	Cal Hidratada	Cal hidratada CH-III	saco	80
	2	Areia	Areia fina para construção	metro c...	50
	15	Porta de Madeira	Porta de madeira 2,10 x 0,80	unidade	50

Tabela 5 – Consulta 5 (Autor)

6) Listar todas as entradas no estoque feitas após 1° de junho de 2024:

```

31 ● SELECT *
32 FROM EntradaEstoque
33 WHERE Data_Entrada > '2024-06-01';

```

Algoritmo 21 – Consulta 6 (Autor)

	ID_Entrada	Data_Entrada	Quantidade	ID_Material	ID_Pedido
▶	11	2024-06-06	200	11	1
	12	2024-06-07	1000	12	2
	13	2024-06-08	500	13	3
	14	2024-06-09	10000	14	4
	15	2024-06-10	50	15	5
	16	2024-06-11	30	16	1
	17	2024-06-12	150	17	2
	18	2024-06-13	500	18	3
	19	2024-06-14	300	19	4
	20	2024-06-15	200	20	5

Tabela 6 – Consulta 6 (Autor)

7) Listar os fornecedores que tem e-mail cadastrado:

```

35 ● SELECT *
36 FROM Fornecedor
37 WHERE Email IS NOT NULL;

```

Algoritmo 22 – Consulta 7 (Autor)

	ID_Fornecedor	Nome	Endereco	Telefone	Email
▶	1	Ferragens	Rua das Flores, 123, Jardins	1234-5678	contato@fornecedora.com
	2	Ferramentas	Avenida Central, 456, Jardins	9876-5432	vendas@fornecedorb.com
	3	Gaúcho Materiais	Rua Nova, 789, Jardins	1357-2468	info@fornecedorc.com
	4	Deposito Construção	Rua Principal, 101, Jardins	2468-1357	atendimento@fornecedord.com
	5	Fornecendo Sua Casa	Avenida Secundária, 202, São Paulo	3698-7451	compras@fornecedore.com

Tabela 7 – Consulta 7 (Autor)

8) Listar todos os materiais com descrição contendo a palavra 'cimento':

```

36 ● SELECT *
37 FROM Material
38 WHERE Descricao LIKE '%cimento%';

```

Algoritmo 23 – Consulta 8 (Autor)

	ID_Material	Nome	Descricao	Unidade	Quantidade_Atual
▶	1	Cimento	Cimento Portland tipo I	saco	100

Tabela 8 – Consulta 8 (Autor)

9) Calcular o valor médio dos materiais em estoque:

```

40 ● SELECT AVG(Quantidade_Atual) AS QuantidadeMedia
41 FROM Material;

```

Algoritmo 24 – Consulta 9 (Autor)

	QuantidadeMedia
▶	1092.0000

Tabela 9 – Consulta 9 (Autor)

10) Listar as quantidades totais de cada material recebidas pelo fornecedor 'Gaúcho Materiais':

```

44 ● SELECT m.Nome, SUM(e.Quantidade) AS QuantidadeTotal
45 FROM Material m
46 JOIN EntradaEstoque e ON m.ID_Material = e.ID_Material
47 JOIN Pedido p ON e.ID_Pedido = p.ID_Pedido
48 JOIN Fornecedor f ON p.ID_Fornecedor = f.ID_Fornecedor
49 WHERE f.Nome = 'Gauchão Materiais'
50 GROUP BY m.Nome;

```

Algoritmo 25 – Consulta 10 (Autor)

	Nome	QuantidadeTotal
►	Brita	30
	Tubos PVC	100
	Canos de Cobre	500
	Interruptor	500

Tabela 10 – Consulta 10 (Autor)

11) Calcular o total de materiais diferentes recebidos de cada fornecedor:

```

58 ● SELECT f.Nome, COUNT(DISTINCT e.ID_Material) AS TotalMateriais
59 FROM Fornecedor f
60 JOIN Pedido p ON f.ID_Fornecedor = p.ID_Fornecedor
61 JOIN EntradaEstoque e ON p.ID_Pedido = e.ID_Pedido
62 GROUP BY f.Nome;

```

Algoritmo 26 – Consulta 11 (Autor)

	Nome	TotalMateriais
►	Deposito Construção	4
	Ferragens	4
	Ferramentas	4
	Fornecendo Sua Casa	4
	Gauchão Materiais	4

Tabela 11 – Consulta 12 (Autor)

12) Listar todos os fornecedores localizados no bairro 'Jardins':

```

65 ● SELECT *
66 FROM Fornecedor
67 WHERE Endereco LIKE '%Jardins%';

```

Algoritmo 27 – Consulta 12 (Autor)

	ID_Fornecedor	Nome	Endereco	Telefone
►	1	Ferragens	Rua das Flores, 123, Jardins	1234-5678
	2	Ferramentas	Avenida Central, 456, Jardins	9876-5432
	3	Gauchão Materiais	Rua Nova, 789, Jardins	1357-2468
	4	Deposito Construção	Rua Principal. 101. Jardins	2468-1357

Tabela 12 – Consulta 12 (Autor)

13) Listar todos os materiais recebidos no pedido com ID 1:

```
72 • SELECT m.Nome, m.Descricao
73 FROM Material m
74 JOIN EntradaEstoque e ON m.ID_Material = e.ID_Material
75 WHERE e.ID_Pedido = 1;
```

Algoritmo 28 – Consulta 13 (Autor)

	Nome	Descricao
►	Cimento	Cimento Portland tipo I
	Aço CA-50	Barra de aço CA-50 12m
	Tinta Acrílica	Tinta acrílica para paredes
	Janela de Alumínio	Janela de alumínio 1,20 x 1,00

Tabela 13– Consulta 13 (Autor)

14) Selecionar os materiais cujo estoque é menor que 100 unidades:

```
77 • SELECT Nome, Quantidade_Atual
78 FROM Material
79 WHERE Quantidade_Atual < 100;
```

Algoritmo 29 - Consulta 14 (Autor)

	Nome	Quantidade_Atual
►	Areia	50
	Brita	30
	Cal Hidratada	80
	Porta de Madeira	50
	Janela de Alumínio	30

Tabela 14 - Consulta 16 (Autor)

15) Calcular a quantidade total de 'Areia' no estoque:

```
82 • SELECT SUM(Quantidade) AS TotalAreia
83 FROM EntradaEstoque
84 WHERE ID_Material = (SELECT ID_Material FROM Material WHERE Nome = 'Areia');
```

Algoritmo 30 - Consulta 15 (Autor)

	TotalAreia
►	50

Tabela 15 – Consulta 15 (Autor)

16) Listar todas as entradas de estoque de 'Cimento':

```
86 • SELECT e.*
87 FROM EntradaEstoque e
88 JOIN Material m ON e.ID_Material = m.ID_Material
89 WHERE m.Nome = 'Cimento';
```

Algoritmo 31 - Consulta 16 (Autor)

	ID_Entrada	Data_Entrada	Quantidade	ID_Material	ID_Pedido
▶	1	2024-05-06	100	1	1

Tabela 16 - Consulta 16 (Autor)

17) Listar os pedidos feitos ao fornecedor 'Ferramentas':

```
92 • SELECT p.*
93 FROM Pedido p
94 JOIN Fornecedor f ON p.ID_Fornecedor = f.ID_Fornecedor
95 WHERE f.Nome = 'Ferramentas';
```

Algoritmo 32 - Consulta 17 (Autor)

	ID_Pedido	Data_Pedido	Data_Entrega	ID_Fornecedor
▶	2	2024-06-02	NULL	2

Tabela 17 - Consulta 17 (Autor)

18) Contar o número total de materiais:

```
99 • SELECT COUNT(*) AS TotalMateriais FROM Material;
```

Algoritmo 33 - Consulta 18 (Autor)

	TotalMateriais
▶	20

Tabela 18 - Consulta 18 (Autor)

19) Listar os fornecedores e a quantidade total de materiais que eles forneceram:

```
103 • SELECT f.Nome, SUM(e.Quantidade) AS TotalMateriaisFornecidos
104 FROM Fornecedor f
105 JOIN Pedido p ON f.ID_Fornecedor = p.ID_Fornecedor
106 JOIN EntradaEstoque e ON p.ID_Pedido = e.ID_Pedido
107 GROUP BY f.Nome;
```

Algoritmo 34 - Consulta 19 (Autor)

	Nome	TotalMateriaisFornecidos
►	Ferragens	480
	Ferramentas	1280
	Gaúcho Materiais	1130
	Deposito Construção	16500
	Fornecendo Sua Casa	2450

Tabela 19 - Consulta 19 (Autor)

20) Listar os materiais que tem quantidade em estoque maior que a média geral de estoque:

```

109 • SELECT *
110 FROM Material
111 WHERE Quantidade_Atual > (SELECT AVG(Quantidade_Atual) FROM Material);

```

Algoritmo 35 - Consulta 20 (Autor)

	ID_Material	Nome	Descricao	Unidade	Quantidade_Atual
►	4	Tijolo	Tijolo cerâmico 8 furos	unidade	5000
	5	Bloco de Concreto	Bloco de concreto 14x19x39	unidade	2000
	9	Telha Cerâmica	Telha cerâmica colonial	unidade	1200
	14	Parafuso	Parafuso 3cm	unidade	10000

Tabela 20 - Consulta 20 (Autor)

21) Calcular a quantidade total de cada material recebida em cada pedido:

```

115 • SELECT p.ID_Pedido, m.Nome, SUM(e.Quantidade) AS QuantidadeTotal
116 FROM Pedido p
117 JOIN EntradaEstoque e ON p.ID_Pedido = e.ID_Pedido
118 JOIN Material m ON e.ID_Material = m.ID_Material
119 GROUP BY p.ID_Pedido, m.Nome;

```

Algoritmo 36 - Consulta 21 (Autor)

	ID_Pedido	Nome	QuantidadeTotal
►	1	Cimento	100
	1	Aço CA-50	150
	1	Tinta Acrilica	200
	1	Janela de Alumínio	30
	2	Areia	50
	2	Cal Hidratada	80
	2	Fio Elétrico	1000
	2	Piso Cerâmico	150
	3	Brita	30

Tabela 21 - Consulta 21 (Autor)

22) Listar os materiais que ainda não foram utilizados em nenhuma saída de estoque:

```
127 • SELECT *
128 FROM Material
129 WHERE ID_Material NOT IN (SELECT DISTINCT ID_Material FROM SaidaEstoque);
```

Algoritmo 37 - Consulta 22 (Autor)

	ID_Material	Nome	Descricao	Unidade	Quantidade_Atual
	12	Fio Elétrico	Fio elétrico 2,5mm	metro	1000
	13	Canos de Cobre	Canos de cobre 1/2 polegada	metro	500

Tabela 22 - Consulta 22 (Autor)

23) Listar os materiais que tiveram uma quantidade de entrada superior a 100 unidades:

```
131 • SELECT *
132 FROM Material
133 WHERE ID_Material IN (SELECT ID_Material FROM EntradaEstoque GROUP BY ID_Material HAVING SUM(Quantidade) > 100);
```

Algoritmo 38 - Consulta 23 (Autor)

	ID_Material	Nome	Descricao	Unidade	Quantidade_Atual
►	4	Tijolo	Tijolo cerâmico 8 furos	unidade	5000
	5	Bloco de Concreto	Bloco de concreto 14x19x39	unidade	2000
	6	Aço CA-50	Barra de aço CA-50 12m	barra	150
	9	Telha Cerâmica	Telha cerâmica colonial	unidade	1200
	10	Argamassa	Argamassa AC-II para revestimento	saco	200
	11	Tinta Acrílica	Tinta acrílica para paredes	litro	200
	12	Fio Elétrico	Fio elétrico 2,5mm	metro	1000
	13	Canos de Cobre	Canos de cobre 1/2 polegada	metro	500
	14	Parafuso	Parafuso 3cm	unidade	10000

Tabela 23 - Consulta 23 (Autor)

24) Calcular o total de materiais que foram recebidos e utilizados, apresentando o saldo atual no estoque:

```

158 • SELECT m.Nome,
159        COALESCE(SUM(e.Quantidade), 0) AS TotalRecebido,
160        COALESCE((SELECT SUM(s.Quantidade) FROM SaidaEstoque s WHERE s.ID_Material = m.ID_Material), 0) AS TotalUtilizado,
161        m.Quantidade_Atual AS SaldoAtual
162 FROM Material m
163 LEFT JOIN EntradaEstoque e ON m.ID_Material = e.ID_Material
164 GROUP BY m.Nome, m.Quantidade_Atual;

```

Algoritmo 39 - Consulta 34 (Autor)

	Nome	TotalRecebido	TotalUtilizado	SaldoAtual
►	Cimento	100	20	100
	Areia	50	10	50
	Brita	30	5	30
	Tijolo	5000	1000	5000
	Bloco de Concreto	2000	500	2000
	Aço CA-50	150	30	150
	Cal Hidratada	80	10	80
	Tubos PVC	100	20	100

Tabela 24 - Consulta 24 (Autor)

25) Listar todas as entradas de estoque ordenadas pela quantidade recebida, em ordem decrescente:

```

215 • SELECT e.Data_Entrada, m.Nome AS Material, e.Quantidade
216 FROM EntradaEstoque e
217 JOIN Material m ON e.ID_Material = m.ID_Material
218 ORDER BY e.Quantidade DESC;

```

Algoritmo 40 - Consulta 25 (Autor)

	Data_Entrada	Material	Quantidade
►	2024-06-09	Parafuso	10000
	2024-05-09	Tijolo	5000
	2024-05-10	Bloco de Concreto	2000
	2024-05-14	Telha Cerâmica	1200
	2024-06-07	Fio Elétrico	1000
	2024-06-08	Canos de Cobre	500
	2024-06-13	Interruptor	500
	2024-06-14	Lâmpada LED	300
	2024-05-15	Argamassa	200
	2024-06-06	Tinta Acrílica	200
	2024-06-15	Fita Isolante	200
	2024-05-11	Aço CA-50	150

Tabela 25 – Consulta 25 (Autor)

26) Listar os materiais que tiveram saídas superiores a 50 unidades em uma única transação:

```
189 • SELECT m.Nome, s.Quantidade
190 FROM Material m
191 JOIN SaidaEstoque s ON m.ID_Material = s.ID_Material
192 WHERE s.Quantidade > 50;
```

Algoritmo 41 - Consulta 26 (Autor)

	Nome	Quantidade
▶	Tijolo	1000
	Bloco de Concreto	500
	Telha Cerâmica	200

Tabela 26 - Consulta 26 (Autor)

27) Calcular a quantidade total de material saída até o momento:

```
197 • SELECT m.Nome AS Material, SUM(s.Quantidade) AS TotalSaida
198 FROM Material m
199 JOIN SaidaEstoque s ON m.ID_Material = s.ID_Material
200 GROUP BY m.Nome;
```

Algoritmo 42 - Consulta 27 (Autor)

	Material	TotalSaida
▶	Cimento	20
	Areia	10
	Brita	5
	Tijolo	1000
	Bloco de Concreto	500
	Aço CA-50	30
	Cal Hidratada	10
	Tubos PVC	20
	Telha Cerâmica	200
	Argamassa	50

Tabela 27 - Consulta 27 (Autor)

28) Listar todos os materiais, suas quantidades de entradas e saídas totais:

```
206 • SELECT m.Nome AS Material,
207 SUM(e.Quantidade) AS TotalEntrada,
208 SUM(s.Quantidade) AS TotalSaida
209 FROM Material m
210 LEFT JOIN EntradaEstoque e ON m.ID_Material = e.ID_Material
211 LEFT JOIN SaidaEstoque s ON m.ID_Material = s.ID_Material
212 GROUP BY m.Nome;
```

Algoritmo 43 - Consulta 28 (Autor)

	Material	TotalEntrada	TotalSaida
►	Cimento	100	20
	Areia	50	10
	Brita	30	5
	Tijolo	5000	1000
	Bloco de Concreto	2000	500
	Aço CA-50	150	30
	Cal Hidratada	80	10
	Tubos PVC	100	20
	Telha Cerâmica	1200	200
	Argamassa	200	50

Tabela 28 - Consulta 28 (Autor)

29) Listar todos os materiais e o fornecedor que mais forneceu cada um deles:

```

220 • SELECT m.Nome AS Material, f.Nome AS Fornecedor, SUM(e.Quantidade) AS QuantidadeTotal
221 FROM Material m
222 JOIN EntradaEstoque e ON m.ID_Material = e.ID_Material
223 JOIN Pedido p ON e.ID_Pedido = p.ID_Pedido
224 JOIN Fornecedor f ON p.ID_Fornecedor = f.ID_Fornecedor
225 GROUP BY m.ID_Material, f.ID_Fornecedor
226 HAVING SUM(e.Quantidade) = (
227     SELECT MAX(Total)
228     FROM (
229         SELECT SUM(e2.Quantidade) AS Total
230         FROM EntradaEstoque e2
231         JOIN Pedido p2 ON e2.ID_Pedido = p2.ID_Pedido
232         WHERE e2.ID_Material = m.ID_Material
233         GROUP BY p2.ID_Fornecedor
234     ) AS SubQuery
235 );

```

Algoritmo 44 - Consulta 29 (Autor)

	Material	Fornecedor	QuantidadeTotal
►	Cimento	Ferragens	100
	Aço CA-50	Ferragens	150
	Tinta Acrílica	Ferragens	200
	Janela de Alumínio	Ferragens	30
	Areia	Ferramentas	50
	Cal Hidratada	Ferramentas	80
	Fio Elétrico	Ferramentas	1000
	Piso Cerâmico	Ferramentas	150
	Brita	Gaúcho Materiais	30
	Tubos PVC	Gaúcho Materiais	100
	Canos de Cobre	Gaúcho Materiais	500
	Interruptor	Gaúcho Materiais	500
	Tijolo	Deposito Constru...	5000
	Telha Cerâmica	Deposito Constru...	1200
	Parafuso	Deposito Constru...	10000
	Lâmpada LED	Deposito Constru...	300
	Bloco de Concreto	Fornecendo Sua ...	2000

Tabela 29- Consulta 29 (Autor)

30) Listar os materiais que ainda não tiveram saída no estoque:

```
238 ● SELECT m.Nome AS Material
239 FROM Material m
240 LEFT JOIN SaidaEstoque s ON m.ID_Material = s.ID_Material
241 WHERE s.ID_Saida IS NULL;
```

Algoritmo 45 - Consulta 30 (Autor)

	Material
►	Tinta Acrílica
	Fio Elétrico
	Canos de Cobre
	Parafuso
	Porta de Madeira
	Janela de Alumínio
	Piso Cerâmico
	Interruptor
	Lâmpada LED
	Fita Isolante

Tabela 30 - Consulta 30 (Autor).

4. CONCLUSÃO

O desenvolvimento do sistema de controle de estoque para o canteiro de obras, utilizando um modelo relacional de banco de dados SQL, provou ser uma solução eficaz para a gestão dos materiais necessários à construção civil. A metodologia aplicada, que incluiu a coleta de requisitos, modelagem conceitual, modelagem lógica, implementação física do banco de dados e desenvolvimento de consultas SQL, resultou em um sistema robusto e funcional. Este sistema é capaz de registrar com precisão as entradas e saídas de materiais, manter o controle sobre os estoques e fornecer informações relevantes através de consultas SQL bem definidas.

A validação do sistema confirmou sua eficiência e precisão, contribuindo para a otimização dos processos de gestão de estoque no canteiro de obras. A documentação detalhada do processo de desenvolvimento assegura que o sistema possa ser mantido e aprimorado no futuro. Sugestões para melhorias futuras incluem a integração com outros sistemas de gestão utilizados na construção civil e a automação de processos, como a atualização automática dos estoques.

O sistema alcançou seus objetivos, proporcionando uma gestão de estoque mais eficiente e precisa, melhorando significativamente os processos no canteiro de obras.

REFERÊNCIAS

CONNOLLY, T. M.; BEGG, C. E. **Bancos de Dados: Uma Abordagem Prática**. Pearson, 2015.

CORONEL, Carlos; ROB, Peter. **Sistema de Banco de Dados: Projeto, Implementação e Gerenciamento**. Volume 4, 6ª edição. São Paulo: Atlas, 2009.

ELMASRI, R.; NAVATHE, S. B. **Fundamentos de Sistemas de Bancos de Dados**, Pearson, 2019.

HEUSER, Carlos Alberto. **Projeto de Banco de Dados**. Porto Alegre: Bookman, 2009.

LAUDON, K. C.; LAUDON, J. P. **Sistemas de Informação Gerenciais**. Pearson, 2019.

RAMAKRISHNAN, R.; GEHRKE, J. **Sistemas de Gerenciamento de Bancos de Dados**. McGraw-Hill, 2002.

SILBERSCHATZ, A.; KORTH, H. F.; SUDARSHAN, S. **Sistema de Banco de Dados**. Campus, 2019.