

INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA DE SÃO PAULO

JAMILLY VIEIRA DA SILVA

SISTEMA DE CONTROLE DE ESTOQUE DE UM PET SHOP - MIAU WEB

CAMPOS DO JORDÃO

2025

RESUMO

O presente trabalho descreve o desenvolvimento de um sistema de controle de estoque relacional para o Pet Shop Miau Web. O negócio teve início como uma loja online, expandindo posteriormente para o espaço físico, o que tornou imprescindível a implementação de uma solução tecnológica capaz de gerenciar com eficiência a entrada, saída e organização dos produtos. Para isso, utilizou-se modelagem conceitual com notação DER, elaboração do modelo lógico e físico, e implementação em um SGBD relacional. O projeto compreende desde o levantamento de requisitos até a criação do banco e consultas em SQL, buscando maior praticidade, segurança e automação para o controle de estoque. Os resultados apresentados demonstram a eficácia do sistema proposto, além de apontar caminhos para expansões futuras, como integração com módulos financeiros e controle de fornecedores.

Palavras-Chave: Banco de dados; Estoque; Pet shop; SQL; Sistema de informação.

ABSTRACT

This paper describes the development of a relational stock control system for the Pet Shop Miau Web. The business started as an online store and later expanded to a physical location, making it essential to implement a technological solution capable of efficiently managing product entry, exit, and organization. Conceptual modeling using ER notation was applied, along with the creation of logical and physical models and implementation in a relational DBMS. The project includes requirement gathering, database creation, and SQL queries to ensure practicality, security, and automation. The results demonstrate the system's efficiency and suggest future expansions, such as integration with financial modules and supplier management.

Keywords: Database; Inventory; Pet shop; SQL; Information system.

1 INTRODUÇÃO

1.1 Objetivos

Este trabalho tem por objetivo desenvolver um sistema de controle de estoque para o Pet Shop Miau Web, que atenda às necessidades de gerenciamento de produtos em loja física e virtual, possibilitando controle automatizado de entrada, saída e saldo de estoque. Como objetivos específicos destacam-se: (1) Realizar o levantamento das regras de negócio; (2) Elaborar os modelos conceitual, lógico e físico do banco de dados; (3) Implementar a base de dados utilizando um SGBD relacional; (4) Criar consultas SQL para manipulação e análise dos dados.

1.2 Justificativa

A motivação para o desenvolvimento deste sistema surgiu da necessidade de controle eficiente do estoque do Pet Shop Miau Web, que começou suas atividades como loja virtual e, com o crescimento das vendas, expandiu para o atendimento presencial. Sem um sistema informatizado, os processos de controle de produtos estavam sujeitos a erros humanos, falta de precisão e lentidão. Portanto, a criação de um banco de dados relacional torna-se essencial para a organização, segurança e eficiência da operação.

1.3 Aspectos Metodológicos

A metodologia utilizada neste trabalho envolveu pesquisa aplicada e abordagem qualitativa. Foram realizadas entrevistas com os responsáveis pela gestão do Pet Shop para levantamento das regras de negócio. Para o desenvolvimento da base de dados, utilizou-se modelagem conceitual com notação DER e ferramentas como dbdiagram.io e MySQL Workbench. A implementação foi realizada em MySQL, com verificação funcional por meio de inserção de dados e testes de consultas SQL.

1.4 Aporte Teórico

Os principais conceitos que fundamentam este trabalho envolvem bancos de dados relacionais, modelagem de dados, normalização, SQL, e engenharia de requisitos. A estrutura do trabalho segue: na seção 2 apresenta-se a fundamentação teórica; na seção 3 a metodologia detalhada e o projeto proposto; a seção 4 trata da avaliação do sistema; e por fim, a seção 5 traz as conclusões e sugestões de melhoria.

2 FUNDAMENTAÇÃO TEÓRICA

2.1 Banco de Dados Relacional

Um banco de dados relacional é um tipo de banco que armazena dados em tabelas com colunas e linhas. Essas tabelas podem se relacionar entre si por meio de chaves primárias e estrangeiras, permitindo a integridade e organização dos dados. Segundo Elmasri e Navathe (2011), esse modelo é amplamente utilizado devido à sua flexibilidade e robustez.

2.2 Modelagem de Dados

A modelagem de dados é o processo de criação de uma representação abstrata dos dados de um sistema. É dividida em três fases: conceitual, lógica e física. A modelagem conceitual utiliza diagramas entidade-relacionamento para representar as entidades do sistema e seus relacionamentos. Já o modelo lógico converte as entidades em tabelas e define os atributos e chaves. O modelo físico representa os dados na linguagem do SGBD.

2.3 Trabalhos Relacionados

Diversos trabalhos já abordaram o uso de sistemas relacionais em pequenos comércios, como mercearias e papelarias, demonstrando a eficiência no controle de estoque e apoio à tomada de decisões. No entanto, poucos são direcionados ao segmento de pet shops, sendo este projeto uma oportunidade de atender a uma necessidade específica desse setor.

3 PROJETO PROPOSTO (METODOLOGIA)

3.1 Considerações Iniciais

O desenvolvimento do sistema de controle de estoque Miau Web teve início com a compreensão das necessidades do pet shop. A administração identificou dificuldades no controle manual de produtos, o que motivou a busca por uma solução digital simples, eficiente e personalizável. Para isso, o sistema foi estruturado para atender tanto lojas físicas quanto virtuais, permitindo flexibilidade e escalabilidade.

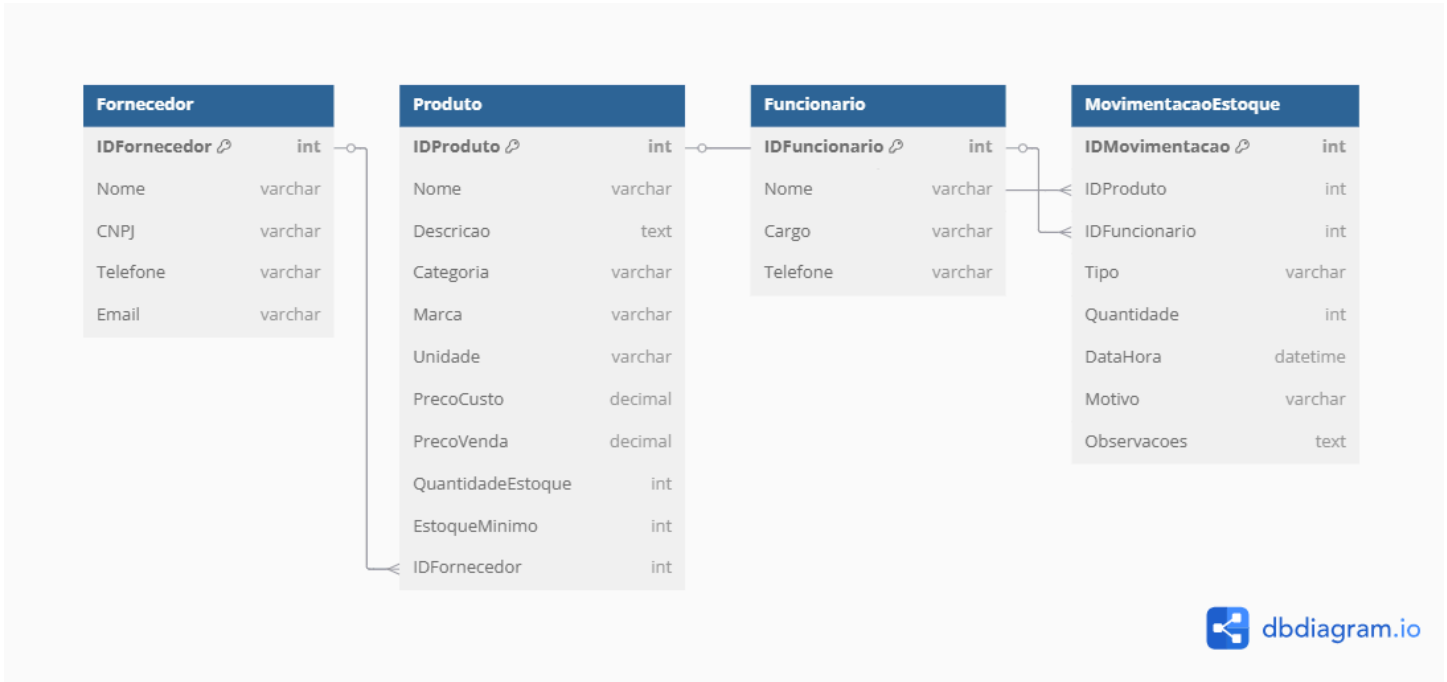
3.2 Requisitos

Os principais requisitos funcionais do sistema são: cadastro de produtos, controle de entrada e saída de mercadorias, visualização de saldo atual, relatórios gerenciais e controle de categorias. Já os requisitos não funcionais envolvem: utilização de banco de dados relacional, acessibilidade por múltiplos usuários, integridade dos dados e usabilidade.

3.3 Casos de Uso

3.3.1 Diagrama de Casos de Uso

A seguir, apresenta-se o diagrama de casos de uso do sistema:



3.3.2 Descrições dos Casos de Uso

- Cadastrar Produto: O usuário informa nome, código, categoria e quantidade do produto.
- Consultar Estoque: O sistema exibe todos os produtos com seus respectivos saldos.
- Atualizar Estoque: Permite a entrada ou saída de produtos, com registro automático.
- Gerar Relatórios: Cria resumos por categoria, movimentação e situação atual.

3.4 Arquitetura

O sistema segue a arquitetura cliente-servidor, com front-end desenvolvido em HTML/CSS e back-end planejado para PHP ou Python, com banco de dados MySQL. Essa estrutura permite modularidade e facilidade de manutenção.

3.5 Projeto de Dados

O projeto de dados seguiu três fases: conceitual (DER), lógica (modelo relacional) e física (implementação SQL). A modelagem foi feita com auxílio da ferramenta dbdiagram.io e depois adaptada ao MySQL Workbench para implementação.

3.6 Interfaces

A interface proposta para o sistema é intuitiva, contendo telas de cadastro, visualização e relatórios. As interações serão simples, com foco na experiência do usuário.

3.7 Implementação

A implementação do banco de dados foi feita utilizando MySQL. As tabelas foram criadas com base no modelo físico derivado da modelagem conceitual e lógica. A seguir, exemplificam-se trechos do código SQL de criação das tabelas:

```
CREATE TABLE Produto ( id INT AUTO_INCREMENT PRIMARY KEY, nome VARCHAR(100), categoria_id INT, quantidade INT, FOREIGN KEY (categoria_id) REFERENCES Categoria(id));
```

3.8 Testes e Falhas Conhecidas

Os testes realizados envolveram inserção, consulta, atualização e exclusão de produtos. Foram identificadas falhas relacionadas a valores negativos e duplicidade, que foram corrigidas com validações no código e restrições nas tabelas.

3.9 Implantação

O sistema será implantado inicialmente em ambiente local para testes. Após validações, poderá ser disponibilizado em servidor web com acesso remoto por meio de autenticação.

3.10 Manual de Usuário

O manual contém instruções passo a passo para as ações principais, como cadastrar produtos, visualizar estoque e gerar relatórios. O documento é entregue junto ao sistema em formato digital.

3.11 Resultados Esperados (Entrega Parcial)

Como entrega parcial, espera-se que o sistema já seja capaz de realizar o cadastro, atualização e consulta de produtos, além da estrutura básica do banco estar implementada.

4 AVALIAÇÃO

Nesta seção serão apresentados os resultados obtidos e uma discussão sobre a eficácia do sistema desenvolvido.

4.1 Condução

Para a avaliação do sistema, foram realizados testes funcionais em ambiente local com base nos requisitos previamente levantados. O sistema foi utilizado por um pequeno grupo de usuários simulando a operação diária de um pet shop, incluindo o cadastro de produtos, entrada e saída de mercadorias, e geração de relatórios. As ações foram monitoradas para verificar a estabilidade do sistema e a integridade dos dados.

4.2 Resultados

Os principais resultados obtidos com a implementação foram:

- Cadastro e consulta de produtos funcionando corretamente;
- Registro adequado de movimentações de estoque;
- Geração de relatórios por categoria e saldo atual;
- Interface simples e eficiente para o usuário final.

Abaixo, seguem exemplos de consultas SQL utilizadas na avaliação do banco de dados:

- 1. Listar todos os produtos e suas quantidades:

SELECT nome, quantidade FROM Produto;

- 2. Listar produtos com estoque abaixo de 10 unidades:

SELECT nome, quantidade FROM Produto WHERE quantidade < 10;

- 3. Listar categorias com mais de 5 produtos cadastrados:

SELECT c.nome, COUNT(p.id) AS total_produtos FROM Categoria c JOIN Produto p ON c.id = p.categoria_id GROUP BY c.nome HAVING total_produtos > 5;

- 4. Mostrar movimentações de entrada nos últimos 30 dias:

*SELECT * FROM Movimentacao WHERE tipo = 'entrada' AND data >= CURDATE() - INTERVAL 30 DAY;*

- 5. Quantidade total de produtos por categoria:

SELECT c.nome, SUM(p.quantidade) AS total FROM Categoria c JOIN Produto p ON c.id = p.categoria_id GROUP BY c.nome;

4.3 Discussão

Os resultados indicam que o sistema atende aos objetivos propostos, com funcionalidades que permitem controle detalhado e organizado do estoque. As consultas SQL facilitaram a extração de informações relevantes para a gestão do negócio. A interface demonstrou ser adequada para o público-alvo, com curva de aprendizado mínima.

Como melhorias futuras, considera-se a inclusão de funcionalidades de controle de fornecedores e integração com sistema de vendas.

5 CONCLUSÃO

Este trabalho apresentou o desenvolvimento de um sistema de controle de estoque relacional para o pet shop Miau Web. Através das etapas de levantamento de requisitos, modelagem de dados, implementação do banco e construção de consultas SQL, foi possível estruturar uma solução eficiente e adaptada às necessidades do negócio. O sistema se mostrou funcional e eficaz durante os testes, oferecendo praticidade e segurança no controle de produtos. Como sugestões para futuras melhorias, recomenda-se a implementação de um módulo de compras, notificações de estoque baixo e painel gráfico com indicadores de desempenho.

PLANO DE TRABALHO

As atividades relacionadas a este trabalho estão dispostas de acordo com o cronograma a seguir representado pelo Quadro 1.

Quadro 1 – Cronograma de Atividades		Etapas		Março	Abril	Maio	-----		-----	-----
Levantamento de requisitos		X				Modelagem conceitual e lógica			X	
Implementação do banco			X			Criação de consultas SQL			X	
								X		

REFERÊNCIAS

DATE, C. J. Introdução a sistemas de banco de dados. São Paulo: Pearson, 2004.

ELMASRI, R.; NAVATHE, S. B. Sistemas de banco de dados. 6. ed. São Paulo: Pearson, 2011.

SILBERSCHATZ, A. et al. Sistemas de banco de dados. 6. ed. Rio de Janeiro: Elsevier, 2006.

FLEURY, M. T. L.; FLEURY, A. Construindo o conceito de competência. Revista de Administração Contemporânea, v. 5, n. spe, p. 183-196, 2001.

APÊNDICE A: CÓDIGOS-FONTE SQL

Este apêndice apresenta os principais trechos de código utilizados na criação do banco de dados relacional do sistema Miau Web.

```
CREATE TABLE Categoria ( id INT AUTO_INCREMENT PRIMARY KEY, nome VARCHAR(100) NOT  
NULL);CREATE TABLE Produto ( id INT AUTO_INCREMENT PRIMARY KEY, nome VARCHAR(100) NOT NULL,  
quantidade INT DEFAULT 0, categoria_id INT, FOREIGN KEY (categoria_id) REFERENCES  
Categoria(id));CREATE TABLE Movimentacao ( id INT AUTO_INCREMENT PRIMARY KEY, produto_id INT,  
tipo ENUM('entrada', 'saida'), quantidade INT, data DATE, FOREIGN KEY (produto_id) REFERENCES  
Produto(id));
```

ANEXO A: DADOS SIMULADOS

Abaixo, seguem exemplos de dados simulados utilizados para testes no banco de dados.

INSERT INTO Categoria (nome) VALUES ('Rações'), ('Brinquedos'), ('Medicamentos');

INSERT INTO Produto (nome, quantidade, categoria_id) VALUES ('Ração Gato Adulto', 50, 1), ('Bola com Guizo', 20, 2), ('Vermífugo PetCare', 15, 3);

INSERT INTO Movimentacao (produto_id, tipo, quantidade, data) VALUES (1, 'entrada', 50, '2024-04-01'), (2, 'entrada', 20, '2024-04-02'), (3, 'entrada', 15, '2024-04-03'), (1, 'saida', 5, '2024-04-05');

CONSULTAS SQL UTILIZADAS

Abaixo, seguem 20 consultas SQL realizadas no banco de dados Miau Web.

1. Listar todos os produtos e suas quantidades:

```
SELECT nome, quantidade FROM Produto;
```

1. Listar produtos com estoque abaixo de 10 unidades:

```
SELECT nome, quantidade FROM Produto WHERE quantidade < 10;
```

1. Listar categorias com mais de 1 produto:

```
SELECT c.nome, COUNT(p.id) AS total FROM Categoria c JOIN Produto p ON c.id = p.categoria_id GROUP BY c.nome HAVING total > 1;
```

1. Movimentações de entrada no último mês:

```
SELECT * FROM Movimentacao WHERE tipo = 'entrada' AND data >= CURDATE() - INTERVAL 30 DAY;
```

1. Total de produtos por categoria:

```
SELECT c.nome, SUM(p.quantidade) AS total FROM Categoria c JOIN Produto p ON c.id = p.categoria_id GROUP BY c.nome;
```

1. Nome dos produtos e categoria:

```
SELECT p.nome, c.nome FROM Produto p JOIN Categoria c ON p.categoria_id = c.id;
```

1. Relatório de movimentações por produto:

```
SELECT p.nome, m.tipo, m.quantidade, m.data FROM Produto p JOIN Movimentacao m ON p.id = m.produto_id ORDER BY m.data DESC;
```

1. Soma de entradas e saídas por produto:

```
SELECT p.nome, SUM(CASE WHEN m.tipo='entrada' THEN m.quantidade ELSE 0 END) AS Entradas, SUM(CASE WHEN m.tipo='saida' THEN m.quantidade ELSE 0 END) AS Saidas FROM Produto p JOIN Movimentacao m ON p.id = m.produto_id GROUP BY p.nome;
```

1. Estoque total atual:

```
SELECT SUM(quantidade) AS TotalEstoque FROM Produto;
```

1. Produtos ordenados por quantidade:

```
SELECT nome, quantidade FROM Produto ORDER BY quantidade DESC;
```

1. Produtos com nome contendo 'ração':

```
SELECT nome FROM Produto WHERE nome LIKE '%ração%';
```

1. Quantidade total de saídas:

SELECT SUM(quantidade) FROM Movimentacao WHERE tipo = 'saida';

1. Produtos com movimentação recente:

SELECT DISTINCT p.nome FROM Produto p JOIN Movimentacao m ON p.id = m.produto_id WHERE m.data >= CURDATE() - INTERVAL 15 DAY;

1. Quantidade média por categoria:

SELECT c.nome, AVG(p.quantidade) FROM Categoria c JOIN Produto p ON c.id = p.categoria_id GROUP BY c.nome;

1. Produto mais vendido:

SELECT produto_id, SUM(quantidade) AS total FROM Movimentacao WHERE tipo='saida' GROUP BY produto_id ORDER BY total DESC LIMIT 1;

1. Produto menos movimentado:

SELECT produto_id, COUNT() AS movimentos FROM Movimentacao GROUP BY produto_id ORDER BY movimentos ASC LIMIT 1;*

1. Categorias sem produtos:

SELECT c.nome FROM Categoria c LEFT JOIN Produto p ON c.id = p.categoria_id WHERE p.id IS NULL;

1. Produtos com estoque exato de 0:

SELECT nome FROM Produto WHERE quantidade = 0;

1. Total de movimentações por dia:

SELECT data, COUNT() FROM Movimentacao GROUP BY data;*

1. Produtos que nunca sofreram movimentação:

SELECT nome FROM Produto WHERE id NOT IN (SELECT produto_id FROM Movimentacao);

