INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA DE SÃO PAULO

ABRAÃO MANGE GARCIA GAMA

SISTEMA DE GERENCIAMENTO DE FARMACIA

CAMPOS DO JORDÃO
2024

RESUMO

O gerenciamento de banco de dados desempenha um papel crucial em diversos setores, sendo especialmente importante no contexto de farmácias, devido à complexidade das informações e à necessidade de controle rigoroso de estoque, vendas e relacionamento com clientes. Este projeto tem como objetivo desenvolver um sistema de gerenciamento de banco de dados para farmácias, visando otimizar os processos operacionais e garantir o cumprimento das regulamentações. A pesquisa foi realizada em farmácias locais, com o intuito de compreender as necessidades do setor e aplicar esse conhecimento na construção do sistema. A análise dos dados coletados sustentou a criação de um modelo de banco de dados eficiente, adequado às especificidades do ambiente farmacêutico.

1. INTRODUÇÃO

O gerenciamento eficiente de farmácias é fundamental para garantir o fornecimento adequado de medicamentos e produtos de saúde, além de oferecer um atendimento de qualidade aos clientes. Nesse contexto, a implementação de um sistema de banco de dados se torna uma solução estratégica para automatizar e organizar as operações diárias das farmácias, como controle de estoque, vendas, gestão de clientes e integração com fornecedores. Este projeto visa desenvolver um sistema de gerenciamento de banco de dados para farmácias, utilizando a modelagem de dados para criar uma estrutura eficiente que atenda às necessidades do setor.

1.1 Objetivos

O objetivo deste trabalho é desenvolver e implementar um sistema de gerenciamento de banco de dados específico para farmácias, visando otimizar a gestão de estoque, o controle de vendas e o relacionamento com clientes e fornecedores. A partir da análise de farmácias locais, serão identificados alguns desafios enfrentados por esses estabelecimentos e como a tecnologia pode ser utilizada para melhorar a eficiência operacional. Além disso, o projeto abordará as principais funcionalidades que um sistema desse tipo deve conter para garantir um gerenciamento adequado e conforme as exigências regulamentares.

1.2 Justificativa

O setor farmacêutico enfrenta uma série de desafios, desde o controle rigoroso de medicamentos controlados até a necessidade de atender a regulamentações governamentais. Com o aumento da demanda por produtos de saúde, a implementação de um sistema de gerenciamento de banco de dados se apresenta como uma solução necessária. A utilização de um sistema integrado de dados proporciona o controle eficiente de estoque, reduz erros operacionais e melhora a relação com os clientes. Portanto, justificar a implementação de um sistema de banco de dados em farmácias é compreender como a tecnologia pode transformar a gestão e trazer benefícios significativos em termos de eficiência, conformidade e satisfação do cliente.

1.3 Aspectos Metodológicos

Para o desenvolvimento deste projeto, a metodologia adotada inclui uma pesquisa de campo realizada em farmácias locais. Durante essa pesquisa, foi possível adquirir base para os processos de gerenciamento de farmácias. O modelo conceitual do banco de dados foi elaborado utilizando a ferramenta **draw.io**, que permitiu representar visualmente as entidades e seus relacionamentos. Para o modelo físico, foi utilizada a linguagem SQL para criar o banco de dados e suas respectivas tabelas. A coleta das regras de negócio foi realizada diretamente nos estabelecimentos, onde foram observados processos como estoque, registros de vendas e atendimento ao cliente.

2. METODOLOGIA

A metodologia de desenvolvimento do sistema baseia-se em quatro principais etapas: a pesquisa de campo, a modelagem conceitual do banco de dados e a implementação do modelo físico.

2.1 Pesquisa de Campo

A pesquisa de campo foi realizada em farmácias locais para compreender a rotina operacional e os desafios enfrentados no gerenciamento de farmácias. Durante a visita aos estabelecimentos, foi possível coletar informações sobre os processos de controle de estoque, gestão de vendas e relacionamento com fornecedores. Além disso, a interação com os funcionários e gestores ajudou a entender a base das necessidades do setor, como integrar diferentes processos em um único sistema.

2.2 Modelagem Conceitual

Com base nas informações coletadas, o modelo conceitual do banco de dados foi desenvolvido utilizando a ferramenta draw.io. Essa ferramenta permitiu criar um diagrama entidade-relacionamento que representa as entidades principais do sistema, como Clientes, Vendas, Produtos e Fornecedores, e os relacionamentos entre elas.

2.3 Implementação do Modelo Físico

A implementação do modelo físico foi realizada utilizando SQL, criando as tabelas e os relacionamentos conforme o modelo conceitual. A ferramenta SQL Server Management Studio 19 possibilitou a criação das estruturas necessárias para armazenar os dados de forma eficiente e organizada. O banco de dados foi projetado para garantir a integridade dos dados, o desempenho nas consultas e a facilidade de manutenção. Durante essa fase, foram realizadas as inserções de dados fictícios nas tabelas, com base nas informações coletadas na pesquisa de campo.

3. RESULTADOS OBTIDOS

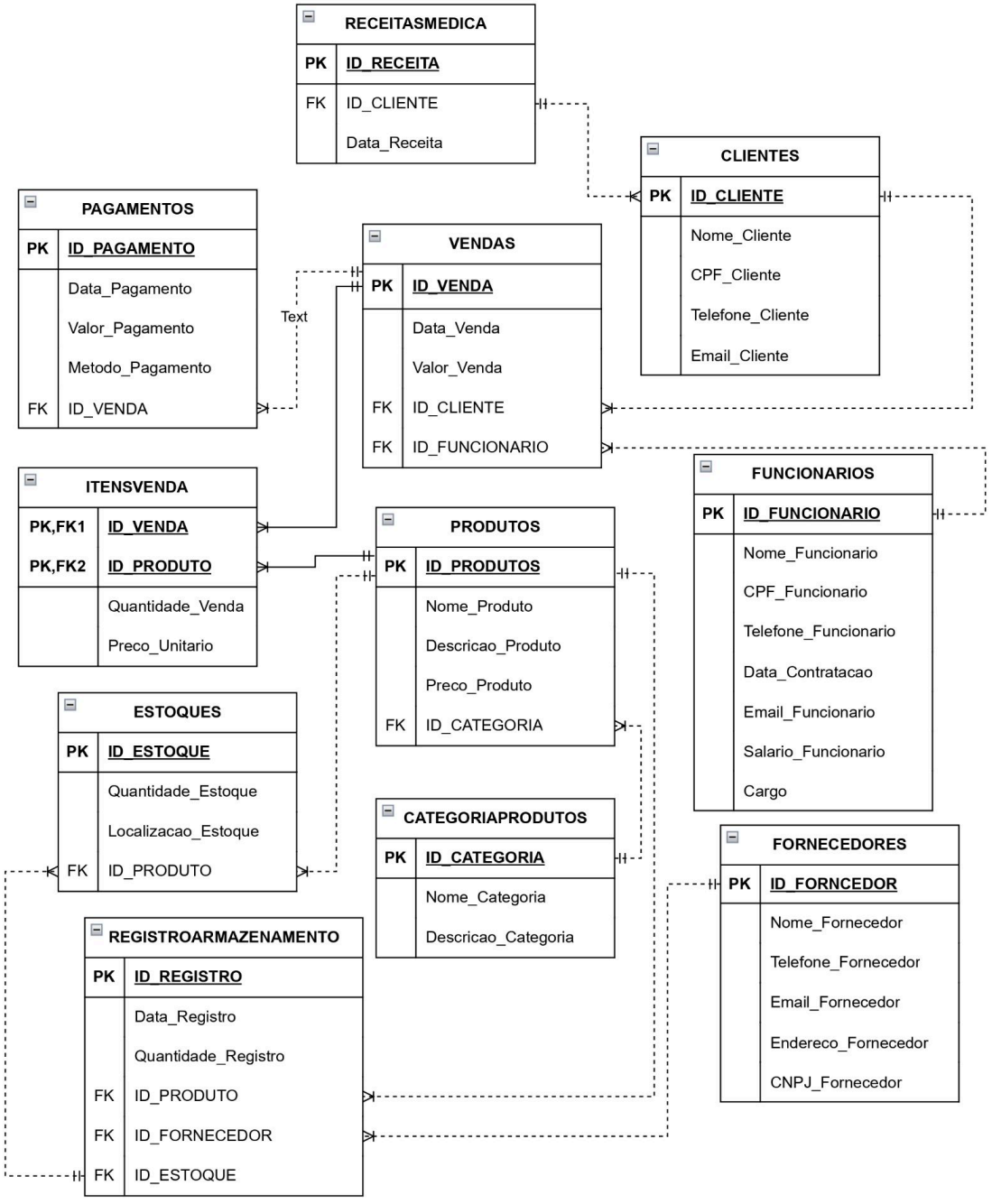
Os resultados obtidos são a apresentação dos dois modelos principais desenvolvidos durante o projeto: o modelo conceitual e o modelo físico.

3.1 Regras de Negócios

1. Relação: Um cliente pode realizar várias vendas.
2. Relação: Um funcionário pode registrar várias vendas.
3. Relação: Cada produto pertence a uma categoria.
4. Relação: Um produto pode estar em várias vendas.
5. Relação: Uma venda pode conter vários produtos.
6. Relação: Um fornecedor pode registrar várias entradas no armazenamento.
7. Relação: Um produto pode ter múltiplos registros de armazenamento.
8. Relação: Um estoque pode estar associado a múltiplos registros.
9. Relação: Uma venda pode ter um ou mais pagamentos.
10. Relação: Um cliente pode ter várias receitas médicas.

3.1 Modelo Conceitual

O modelo conceitual do banco de dados foi elaborado utilizando a ferramenta draw.io. O diagrama entidade-relacionamento resultante apresentou as principais entidades do sistema, como Clientes, Produtos, Vendas, Fornecedores e Estoque, e os relacionamentos entre elas. O modelo conceitual também incluiu atributos importantes para cada entidade, como o nome do cliente, a quantidade de produto em estoque, o preço de venda,



3.2 Modelo Físico

A implementação do modelo físico foi feita utilizando SQL, e o código de criação das tabelas foi gerado com base no modelo conceitual. O banco de dados foi estruturado com as tabelas adequadas e com os tipos de dados apropriados para armazenar as informações de forma eficiente. O código SQL para criação do banco de dados inclui a definição das tabelas, as chaves primárias e estrangeiras, e as restrições necessárias para garantir a integridade dos dados.

```
CREATE DATABASE GERENCIAMENTO_FARMACIA;
GO


USE GERENCIAMENTO_FARMACIA;
GO


CREATE TABLE CLIENTES (
    ID_CLIENTE INT PRIMARY KEY,
    Nome_Cliente NVARCHAR(100) NOT NULL,
    CPF_Cliente CHAR(11) UNIQUE NOT NULL,
    Telefone_Cliente NVARCHAR(15),
    Email_Cliente NVARCHAR(100)
);
GO


CREATE TABLE FUNCIONARIOS (
    ID_FUNCIONARIO INT PRIMARY KEY,
    Nome_Funcionario NVARCHAR(100),
    CPF_Funcionario CHAR(11) UNIQUE NOT NULL,
    Telefone_Funcionario NVARCHAR(15),
    Data_Contratacao DATE,
    Salario DECIMAL(10, 2),
    Email_Funcionario NVARCHAR(100),
```

```
    Cargo NVARCHAR(50)
);
GO

CREATE TABLE CATEGORIAPRODUTOS (
    ID_CATEGORIA INT PRIMARY KEY,
    Nome_Categoria NVARCHAR(50) NOT NULL,
    Descricao_Categoria NVARCHAR(500)
);
GO

CREATE TABLE PRODUTOS (
    ID_PRODUTOS INT PRIMARY KEY,
    Nome_Produto NVARCHAR(100) NOT NULL,
    Descricao_Produto NVARCHAR(500),
    Preco_Produto DECIMAL(10, 2),
    ID_CATEGORIA INT,
    FOREIGN KEY (ID_CATEGORIA) REFERENCES
CATEGORIAPRODUTOS(ID_CATEGORIA)
);
GO

CREATE TABLE FORNECEDORES (
    ID_FORNCEDOR INT PRIMARY KEY,
```

```
    Nome_Fornecedor NVARCHAR(100) NOT NULL,
    Telefone_Fornecedor NVARCHAR(15),
    Email_Fornecedor NVARCHAR(100),
    Endereco_Fornecedor NVARCHAR(200),
    CNPJ_Fornecedor CHAR(14) UNIQUE NOT NULL
);
GO



-- Tabela REGISTROARMAZENAMENTO
CREATE TABLE REGISTROARMAZENAMENTO (
    ID_REGISTRO INT PRIMARY KEY,
    Data_Registro DATE,
    Quantidade_Registro INT,
    ID_PRODUTO INT,
    ID_FORNECEDOR INT,
    ID_ESTOQUE INT,
    FOREIGN KEY (ID_PRODUTO) REFERENCES PRODUTOS(ID_PRODUTOS),
    FOREIGN KEY (ID_FORNECEDOR) REFERENCES
FORNECEDORES(ID_FORNCEDOR),
    FOREIGN KEY (ID_ESTOQUE) REFERENCES ESTOQUES(ID_ESTOQUE)
);
GO



CREATE TABLE ESTOQUES (
    ID_ESTOQUE INT PRIMARY KEY,
```

```
    Quantidade_Estoque INT,
    Localizacao_Estoque NVARCHAR(100),
    ID_PRODUTO INT,
    FOREIGN KEY (ID_PRODUTO) REFERENCES PRODUTOS(ID_PRODUTOS)
);
GO


CREATE TABLE PAGAMENTOS (
    ID_PAGAMENTO INT PRIMARY KEY,
    Data_Pagamento DATE,
    Valor_Pagamento DECIMAL(10, 2),
    Metodo_Pagamento NVARCHAR(50),
    ID_VENDA INT,
    FOREIGN KEY (ID_VENDA) REFERENCES VENDAS(ID_VENDA)
);
GO


CREATE TABLE VENDAS (
    ID_VENDA INT PRIMARY KEY,
    Data_Venda DATE,
    Valor_Venda DECIMAL(10, 2),
    ID_CLIENTE INT,
    ID_FUNCIONARIO INT,
    FOREIGN KEY (ID_CLIENTE) REFERENCES CLIENTES(ID_CLIENTE),
```

```
    FOREIGN KEY (ID_FUNCIONARIO) REFERENCES
FUNCIONARIOS(ID_FUNCIONARIO)

);

GO



CREATE TABLE ITENSVENDA (

    ID_VENDA INT,

    ID_PRODUTO INT,

    Quantidade_Venda INT,

    Preco_Unitario DECIMAL(10, 2),

    PRIMARY KEY (ID_VENDA, ID_PRODUTO),

    FOREIGN KEY (ID_VENDA) REFERENCES VENDAS(ID_VENDA),

    FOREIGN KEY (ID_PRODUTO) REFERENCES PRODUTOS(ID_PRODUTOS)

);

GO



CREATE TABLE RECEITASMEDICA (

    ID_RECEITA INT PRIMARY KEY,

    ID_CLIENTE INT,

    Data_Receita DATE,

    FOREIGN KEY (ID_CLIENTE) REFERENCES CLIENTES(ID_CLIENTE)

);

GO
```

3.3 Consultas Realizadas

A seguir, são apresentadas 30 consultas SQL que podem ser realizadas no banco de dados desenvolvido, cobrindo diversos tipos, para os mais variados tipos de necessidades.

```
-- 1. Verifica clientes que não possuem e-mail cadastrado

SELECT * FROM CLIENTES WHERE Email_Cliente IS NULL;

GO


-- 2. Exibe o nome dos clientes em ordem alfabética

SELECT Nome_Cliente FROM CLIENTES ORDER BY Nome_Cliente;

GO


-- 3. Filtrar clientes com telefone começando com "11"

SELECT * FROM CLIENTES WHERE Telefone_Cliente LIKE '11%';

GO


-- 4. Produtos com preços acima de R$100,00

SELECT * FROM PRODUTOS WHERE Preco_Produto > 100;

GO


-- 5. Lista produtos por categorias específicas

SELECT Nome_Produto FROM PRODUTOS WHERE ID_CATEGORIA = 1;

GO


-- 6. Ordena produtos pelo preço de forma crescente

SELECT Nome_Produto, Preco_Produto FROM PRODUTOS ORDER BY Preco_Produto;

GO
```

```
-- 7. Quantidade total de produtos em estoque
SELECT SUM(Quantidade_Estoque) AS Total_Estoque FROM ESTOQUES;
GO

-- 8. Preço médio dos produtos
SELECT AVG(Preco_Produto) AS Preco_Medio FROM PRODUTOS;
GO

-- 10. Total de vendas realizadas por funcionário específico
SELECT COUNT(*) AS Total_Vendas
FROM VENDAS
WHERE ID_FUNCIONARIO = 3;
GO

-- 11. Valor total de vendas realizadas
SELECT SUM(Valor_Venda) AS Total_Vendas FROM VENDAS;
GO

-- 12. Encontrar funcionários com maior salário
SELECT TOP 1 * FROM FUNCIONARIOS ORDER BY Salario DESC;
GO

-- 12. Listar funcionários que foram contratados antes de 2020
SELECT * FROM FUNCIONARIOS WHERE Data_Contratacao < '2020-01-01';
GO
```

```
-- 13. Produtos vendidos em uma venda específica
SELECT p.Nome_Produto, i.Quantidade_Venda
FROM ITENSVENDA i
INNER JOIN PRODUTOS p ON i.ID_PRODUTO = p.ID_PRODUTOS
WHERE i.ID_VENDA = 10;
GO

-- 14. Receitas médicas associadas a um cliente específico
SELECT * FROM RECEITASMEDICA WHERE ID_CLIENTE = 4;
GO

-- 15. Obter o total de vendas realizadas por cada funcionarios e ordená-las
-- do maior para o menor
SELECT f.Nome_Funcionario, COUNT(v.ID_VENDA) AS Total_Vendas
FROM FUNCIONARIOS f
LEFT JOIN VENDAS v ON f.ID_FUNCIONARIO = v.ID_FUNCIONARIO
GROUP BY f.Nome_Funcionario
ORDER BY Total_Vendas DESC;
GO

-- 16. Clientes que fizera, compras acima de R$500,00
SELECT c.Nome_Cliente, v.Valor_Venda
FROM CLIENTES c
INNER JOIN VENDAS v ON c.ID_CLIENTE = v.ID_CLIENTE
WHERE v.Valor_Venda > 500;
GO
```

```
-- 17. Receitas médicas criadas nos últimos 30 dias
SELECT *
FROM RECEITASMEDICA
WHERE Data_Receita >= DATEADD(DAY, -30, GETDATE());
GO

-- 18. Métodos de pagamento mais utilizados
SELECT Metodo_Pagamento, COUNT(*) AS Total
FROM PAGAMENTOS
GROUP BY Metodo_Pagamento
ORDER BY Total DESC;
GO

-- 19. Lucro total por produto, considerando a quantidade
-- vendida e preço unitário
SELECT p.Nome_Produto, SUM(i.Quantidade_Venda * i.Preco_Unitario) AS Lucro_Total
FROM ITENSVENDA i
INNER JOIN PRODUTOS p ON i.ID_PRODUTO = p.ID_PRODUTOS
GROUP BY p.Nome_Produto
ORDER BY Lucro_Total DESC;
GO

-- 20. Funcionarios que não realizaram vendas
SELECT f.Nome_Funcionario
FROM FUNCIONARIOS f
```

```
LEFT JOIN VENDAS v ON f.ID_FUNCIONARIO = v.ID_FUNCIONARIO

WHERE v.ID_FUNCIONARIO IS NULL;

GO


-- 21. Categoria com o maior números de produtos

SELECT c.Nome_Categoria, COUNT(p.ID_PRODUTOS) AS Total_Produtos

FROM CATEGORIAPRODUTOS c

INNER JOIN PRODUTOS p ON c.ID_CATEGORIA = p.ID_CATEGORIA

GROUP BY c.Nome_Categoria

ORDER BY Total_Produtos DESC;

GO


-- 21. Histórico de vendas por cliente, nome e data

SELECT c.Nome_Cliente, v.Data_Venda, v.Valor_Venda

FROM CLIENTES c

INNER JOIN VENDAS v ON c.ID_CLIENTE = v.ID_CLIENTE

ORDER BY v.Data_Venda DESC;

GO


-- 22. Obter o nome e o telefone dos fornecedores que

-- forneceram mais de 10 produtos

SELECT f.Nome_Fornecedor, f.Telefone_Fornecedor, COUNT(r.ID_PRODUTO) AS
Produtos_Fornecidos

FROM FORNECEDORES f

INNER JOIN REGISTROARMAZENAMENTO r ON f.ID_FORNCEDOR =
r.ID_FORNECEDOR

GROUP BY f.Nome_Fornecedor, f.Telefone_Fornecedor
```

```
HAVING COUNT(r.ID_PRODUTO) > 10;
GO

-- 23. Calcular a receita total gerada por categoria de produto
SELECT c.Nome_Categoria, SUM(i.Quantidade_Venda * i.Preco_Unitario) AS Receita_Total
FROM CATEGORIAPRODUTOS c
INNER JOIN PRODUTOS p ON c.ID_CATEGORIA = p.ID_CATEGORIA
INNER JOIN ITENSVENDA i ON p.ID_PRODUTOS = i.ID_PRODUTO
GROUP BY c.Nome_Categoria;
GO

-- 24. Listar os três produtos mais caros e suas categorias
SELECT TOP 3 p.Nome_Produto, p.Preco_Produto, c.Nome_Categoria
FROM PRODUTOS p
INNER JOIN CATEGORIAPRODUTOS c ON p.ID_CATEGORIA = c.ID_CATEGORIA
ORDER BY p.Preco_Produto DESC;
GO

-- 25. Identificar os funcionários que realizaram vendas acima da média geral
SELECT f.Nome_Funcionario, SUM(v.Valor_Venda) AS Valor_Total
FROM FUNCIONARIOS f
INNER JOIN VENDAS v ON f.ID_FUNCIONARIO = v.ID_FUNCIONARIO
GROUP BY f.Nome_Funcionario
HAVING SUM(v.Valor_Venda) > (
    SELECT AVG(Valor_Venda) FROM VENDAS
);
```

```
GO

-- 26. Produtos com estoque abaixo de 10 unidades
SELECT p.Nome_Produto, e.Quantidade_Estoque
FROM PRODUTOS p
INNER JOIN ESTOQUES e ON p.ID_PRODUTOS = e.ID_PRODUTO
WHERE e.Quantidade_Estoque < 10;
GO

-- 27. Produtos com preços entre R$50,00 e R$150,00 e suas descrições
SELECT Nome_Produto, Descricao_Produto, Preco_Produto
FROM PRODUTOS
WHERE Preco_Produto BETWEEN 50 AND 150;
GO

-- 28. Números de receitas médicas emitidas por clientes
SELECT c.Nome_Cliente, COUNT(r.ID_RECEITA) AS Total_Receitas
FROM CLIENTES c
LEFT JOIN RECEITASMEDICA r ON c.ID_CLIENTE = r.ID_CLIENTE
GROUP BY c.Nome_Cliente;
GO

-- 29. Listar os funcionários com mais de 2 anos de contratação
SELECT Nome_Funcionario, Data_Contratacao
FROM FUNCIONARIOS
WHERE DATEDIFF(YEAR, Data_Contratacao, GETDATE()) > 2;
```

```
GO


-- 30. Obter a média de preço dos produtos em cada categoria

SELECT c.Nome_Categoria, AVG(p.Preco_Produto) AS Preco_Medio

FROM CATEGORIAPRODUTOS c

INNER JOIN PRODUTOS p ON c.ID_CATEGORIA = p.ID_CATEGORIA

GROUP BY c.Nome_Categoria;

GO
```

4. CONCLUSÃO

A implementação de um sistema de banco de dados para farmácias se revelou essencial para garantir a eficiência operacional e o cumprimento das regulamentações. A pesquisa de campo realizada nas farmácias locais destacou a importância de um sistema integrado que controle de forma precisa o estoque, as vendas e o relacionamento com os clientes. A modelagem do banco de dados, utilizando a ferramenta draw.io para o modelo conceitual e SQL para o modelo físico, proporcionou uma solução eficiente para os desafios enfrentados pelo setor farmacêutico. O sistema proposto tem o potencial de otimizar processos, reduzir custos e melhorar a precisão das operações nas farmácias.

REFERÊNCIAS BIBLIOGRÁFICAS

- Rob, Peter; Coronel, Carlos. Sistemas de banco de dados. Tradução da 8ª edição. São Paulo: Cengage Learning, 2010.. São Paulo: Cengage Learning, 2010.
- Draw.io (Diagrams.net) – Ferramenta para modelagem de diagramas. https://app.diagrams.net/?src=about
- SQL Server Management Studio 19 (SSMS 19). Ferramenta oficial para gerência de bancos de dados SQL Server. Disponível em. https://learn.microsoft.com/sql/ssms