



CENTRO UNIVERSITÁRIO UNIRUY – WYDEN

MICHEL WEBER

NATÁLIA SUE

RAQUEL MOURA

BANCO DE DADOS  
PETSHOP PATAQUEPARIU

SALVADOR

2024

MICHEL WEBER

NATÁLIA SUE

RAQUEL MOURA

## BANCO DE DADOS PETSHOP PATAQUEPARIU

Neste estudo, compartilhamos o desenvolvimento de banco de dados para um sistema de administração de um PetShop.

Trabalho desenvolvido na disciplina de Banco de Dados no Centro Universitário UniRuy.

Docente: Heleno Cardoso

SALVADOR

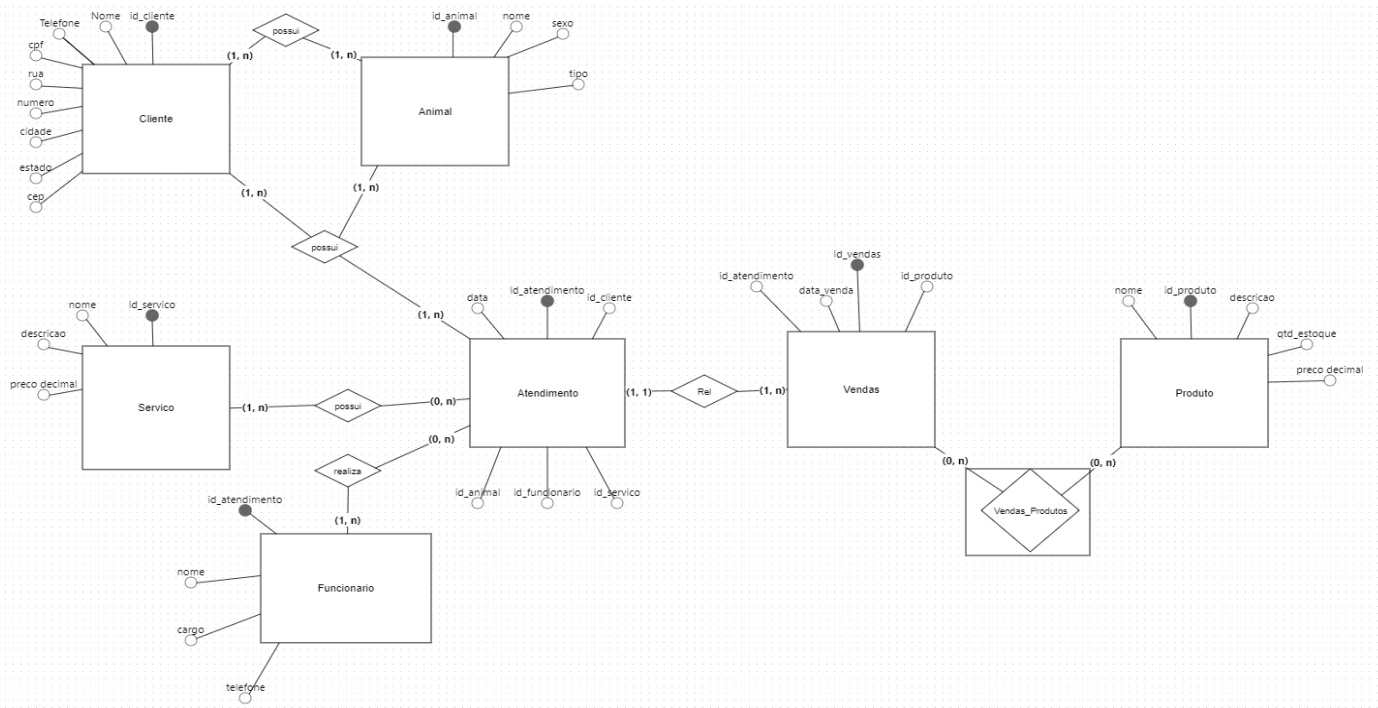
2024

# 1. INTRODUÇÃO

Neste estudo, compartilhamos o desenvolvimento de banco de dados para um sistema de administração de um PetShop. O foco principal é criar e implementar um banco de dados relacional que possa dar suporte às atividades de gestão relacionadas a animais, clientes, funcionários, produtos e serviços. Nosso objetivo é que ao término desse projeto seja possível compreender mais a fundo os processos envolvidos na modelagem conceitual, lógica e física de bancos de dados, além da execução bem-sucedida de um sistema operacional.

A estrutura deste estudo segue a sequência: inicialmente apresentamos o modelo conceitual, onde discutimos o sistema escolhido e exibimos o diagrama Entidade-Relacionamento (ER). Em seguida, detalhamos o modelo lógico do banco de dados com a elaboração do diagrama no MySQL Workbench. Por fim, abordamos o modelo físico ao fornecer os scripts SQL para criar as tabelas e suas respectivas chaves primárias e estrangeiras.

## 2. MODELO CONCEITUAL



O sistema abordado neste projeto é um de gerenciamento de um PetShop. Sua finalidade é administrar dados relacionados a animais, clientes, funcionários, produtos e serviços oferecidos pela loja de animais. O sistema possibilitará o registro de informações sobre animais e clientes, a administração do estoque de produtos, a atribuição de serviços prestados aos animais e a organização dos dados dos funcionários.

### 3. DIAGRAMA ENTIDADE-RELACIONAMENTO (ER)

#### Relacionamentos:

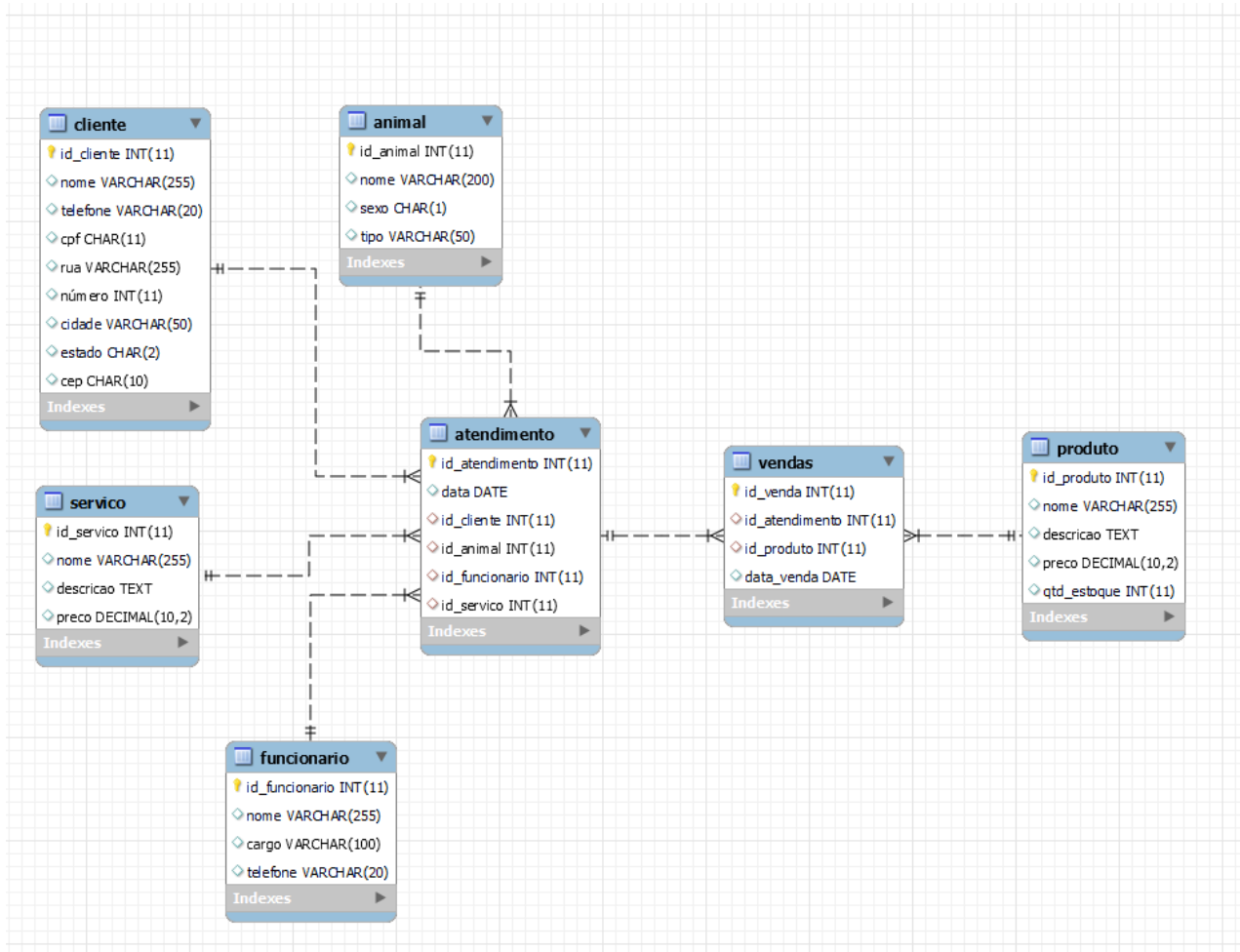
- a. Cliente - Animal: Um cliente pode ter vários animais (1).
- b. Cliente - Atendimento: Um cliente pode ter vários atendimentos (1).
- c. Animal - Atendimento: Um animal pode ter vários atendimentos (1).
- d. Serviço - Atendimento: Um serviço pode estar em vários atendimentos (1).
- e. Funcionário - Atendimento: Um funcionário pode realizar vários atendimentos (1).
- f. Atendimento - Vendas: Um atendimento pode gerar uma venda (1:1).
- g. Vendas - Produtos: Uma venda pode incluir vários produtos (N), necessitando uma tabela associativa (Vendas\_Produtos).

#### Explicação das Cardinalidades:

- 1. **Cliente - Animal (1)**
  - 1. **Explicação:** Um cliente pode ter vários animais, mas cada animal pertence a um único cliente.
- 2. **Cliente - Atendimento (1)**
  - 1. **Explicação:** Um cliente pode ter vários atendimentos, mas cada atendimento está associado a um único cliente.
- 3. **Animal - Atendimento (1)**
  - 1. **Explicação:** Um animal pode ter vários atendimentos, mas cada atendimento está associado a um único animal.
- 4. **Serviço - Atendimento (1)**
  - 1. **Explicação:** Um serviço pode ser prestado em vários atendimentos, mas cada atendimento está associado a um único serviço.
- 5. **Funcionário - Atendimento (1)**
  - 1. **Explicação:** Um funcionário pode realizar vários atendimentos, mas cada atendimento é realizado por um único funcionário.
- 6. **Atendimento - Vendas (1:1)**
  - 1. **Explicação:** Cada atendimento pode gerar uma venda, e cada venda está associada a um único atendimento.
- 7. **Vendas - Produtos (N)**
  - 1. **Explicação:** Uma venda pode incluir vários produtos e um produto pode estar presente em várias vendas. Este relacionamento é modelado por uma tabela associativa que registra cada instância de um produto em uma venda.

## 4. MODELO LÓGICO

O Modelo lógico criado para o banco de dados de um petshop foi pensado para organizar as informações de forma eficiente e garantir a integridade dos dados. Esse modelo reflete as operações diárias do petshop, facilitando o gerenciamento de clientes, animais, atendimentos, funcionários, produtos e serviços. Cada tabela foi criada pensando nos detalhes importantes e definir claramente as relações entre os diferentes elementos, permitindo um acesso rápido e preciso às informações e atendendo às necessidades do negócio.



O modelo lógico detalha como os dados são organizados em um banco de dados relacional. Ele inclui tabelas, colunas, tipos de dados e as relações entre essas tabelas. Abaixo, encontra-se uma descrição das tabelas principais e seus relacionamentos.

## Tabelas Principais e Suas Colunas:

### a. Cliente

- **ID\_Cliente:** Identificador único do cliente (Primary Key)
- **Nome:** Nome do cliente
- **Endereço:** Endereço do cliente
- **Telefone:** Telefone do cliente

### b. Animal

- **ID\_Animal:** Identificador único do animal (Primary Key)
- **Nome:** Nome do animal
- **Tipo:** Espécie do animal (cachorro, gato, etc.)
- **Raça:** Raça do animal
- **Idade:** Idade do animal

### c. Atendimento

- **ID\_Atendimento:** Identificador único do atendimento (Primary Key)
- **Data:** Data do atendimento
- **Hora:** Hora do atendimento
- **ID\_Cliente:** Identificador do cliente (Foreign Key referenciando Cliente)
- **ID\_Animal:** Identificador do animal (Foreign Key referenciando Animal)
- **ID\_Serviço:** Identificador do serviço prestado (Foreign Key referenciando Serviço)
- **ID\_Funcionário:** Identificador do funcionário que realizou o atendimento (Foreign Key referenciando Funcionário)

### d. Serviço

- **ID\_Serviço:** Identificador único do serviço (Primary Key)
- **Descrição:** Descrição do serviço
- **Preço:** Preço do serviço

### e. Funcionário

- **ID\_Funcionário:** Identificador único do funcionário (Primary Key)
- **Nome:** Nome do funcionário
- **Cargo:** Cargo do funcionário
- **Telefone:** Telefone do funcionário

### f. Vendas

- **ID\_Venda:** Identificador único da venda (Primary Key)
- **Data:** Data da venda
- **ID\_Atendimento:** Identificador do atendimento associado (Foreign Key referenciando Atendimento)

### g. Produtos

- **ID\_Produto:** Identificador único do produto (Primary Key)
- **Nome:** Nome do produto
- **Preço:** Preço do produto
- **Quantidade\_em\_Estoque:** Quantidade em estoque do produto

### h. Vendas\_Produtos

- **ID\_Venda:** Identificador da venda (Foreign Key referenciando Vendas)

- **ID\_Produto:** Identificador do produto (Foreign Key referenciando Produtos)
- **Quantidade:** Quantidade do produto na venda
- **Preço\_Venda:** Preço do produto na venda



## 5. MODELO FÍSICO

Apresentamos os comandos SQL responsáveis pela criação das tabelas no banco de dados, incluindo as definições de chaves primárias e estrangeiras.

O modelo físico define a implementação real do banco de dados em um sistema de gerenciamento de banco de dados (SGBD). Ele especifica a estrutura detalhada das tabelas, índices, chaves, tipos de dados e outras propriedades físicas que garantem o desempenho e a integridade dos dados.

### Explicação dos Creates

As linhas de comando a seguir SQL criam a estrutura das tabelas do banco de dados. Cada tabela tem uma chave primária única, que serve para identificar cada registro de forma exclusiva.

As chaves estrangeiras são usadas para estabelecer relações entre as tabelas, conforme definido nos modelos conceitual e lógico.

Por exemplo, a tabela Atendimento tem chaves estrangeiras que referenciam as tabelas Cliente, Animal, Funcionário, Produto e Serviço.

Isso significa que cada atendimento está ligado a um cliente, um animal, um funcionário, um produto e um serviço específicos.

#### Chave Primária (Primary Key - PK)

Uma chave primária é um campo ou conjunto de campos que identifica de maneira única cada registro em uma tabela de banco de dados.

#### Chave Estrangeira (Foreign Key - FK)

Uma chave estrangeira é um campo ou conjunto de campos em uma tabela que cria um vínculo com a chave primária de outra tabela, estabelecendo uma relação entre as duas tabelas. Duas características:

**Referencial:** A chave estrangeira aponta para a chave primária de outra tabela, estabelecendo uma relação entre as tabelas.

**Integridade:** Assegura que os valores na chave estrangeira correspondam a valores válidos na tabela referenciada, mantendo a integridade referencial dos dados.

As informações geradas por este projeto possibilitam o desenvolvimento de estratégias mais refinadas em diversos setores, garantindo melhores resultados financeiros para as empresas.

## **Índices e Chaves**

- **Primary Keys:** Garantem a unicidade de cada registro dentro de uma tabela.
- **Foreign Keys:** Mantêm a integridade referencial entre tabelas.
- **Índices:** Criados em colunas frequentemente pesquisadas para melhorar o desempenho das consultas.

## **Tipos de Dados**

- **INT:** Utilizado para identificadores e quantidades.
- **VARCHAR:** Utilizado para armazenar textos como nomes, endereços e descrições.
- **DATE:** Utilizado para armazenar datas.
- **TIME:** Utilizado para armazenar horas.
- **DECIMAL:** Utilizado para armazenar valores monetários com precisão.

## **Integridade e Constrangimentos**

- **Chaves Primárias (Primary Keys):** Garantem que cada registro é único.
- **Chaves Estrangeiras (Foreign Keys):** Enforce constraints to maintain referential integrity between tables.
- **Not Null Constraints:** Garantem que certos campos devem conter valores válidos, não permitindo valores nulos.

## **Performance e Otimização**

- **Índices:** Criados para colunas frequentemente utilizadas em condições de pesquisa e junções (JOINS).
- **Normalização:** Minimizando redundâncias e garantindo a consistência dos dados.
- **Particionamento de Tabelas:** Pode ser utilizado em tabelas grandes para melhorar o desempenho.

## Tabelas e Esquemas

1. **Cliente**
  - **ID\_Cliente:** INT, Primary Key, Auto Increment
  - **Nome:** VARCHAR(100), NOT NULL
  - **Endereço:** VARCHAR(255), NOT NULL
  - **Telefone:** VARCHAR(15), NOT NULL ☺ ☹ ☺ ☹ ☺ ☹ ☺
2. **Animal**
  - **ID\_Animal:** INT, Primary Key, Auto Increment
  - **Nome:** VARCHAR(100), NOT NULL
  - **Espécie:** VARCHAR(50), NOT NULL
  - **Raça:** VARCHAR(50)
  - **Idade:** INT
  - **ID\_Cliente:** INT, Foreign Key, NOT NULL
3. **Atendimento**
  - **ID\_Atendimento:** INT, Primary Key, Auto Increment
  - **Data:** DATE, NOT NULL
  - **Hora:** TIME, NOT NULL
  - **ID\_Cliente:** INT, Foreign Key, NOT NULL
  - **ID\_Animal:** INT, Foreign Key, NOT NULL
  - **ID\_Serviço:** INT, Foreign Key, NOT NULL
  - **ID\_Funcionário:** INT, Foreign Key, NOT NULL
4. **Serviço**
  - **ID\_Serviço:** INT, Primary Key, Auto Increment
  - **Descrição:** VARCHAR(255), NOT NULL
  - **Preço:** DECIMAL(10, 2), NOT NULL
5. **Funcionário**
  - **ID\_Funcionário:** INT, Primary Key, Auto Increment
  - **Nome:** VARCHAR(100), NOT NULL
  - **Cargo:** VARCHAR(50), NOT NULL
  - **Telefone:** VARCHAR(15), NOT NULL
6. **Vendas**
  - **ID\_Venda:** INT, Primary Key, Auto Increment
  - **Data:** DATE, NOT NULL
  - **ID\_Atendimento:** INT, Foreign Key, Unique, NOT NULL
7. **Produtos**
  - **ID\_Produto:** INT, Primary Key, Auto Increment
  - **Nome:** VARCHAR(100), NOT NULL
  - **Preço:** DECIMAL(10, 2), NOT NULL
  - **Quantidade\_em\_Estoque:** INT, NOT NULL
8. **Vendas\_Produtos**
  - **ID\_Venda:** INT, Foreign Key, NOT NULL
  - **ID\_Produto:** INT, Foreign Key, NOT NULL
  - **Quantidade:** INT, NOT NULL
  - **Preço\_Venda:** DECIMAL(10, 2) -- opcional
  - **Primary Key (ID\_Venda, ID\_Produto)**

## Código SQL

```
CREATE DATABASE PataQuePariu;
```

```
USE PataQuePariu;
```

```
CREATE TABLE Cliente (  
    id_cliente INT PRIMARY KEY AUTO_INCREMENT,  
    nome VARCHAR(255),  
    telefone VARCHAR(20),  
    cpf CHAR(11),  
    rua VARCHAR(255),  
    número INT,  
    cidade VARCHAR(50),  
    estado CHAR(2),  
    cep CHAR(10)  
);
```

```
CREATE TABLE Animal (  
    id_animal INT PRIMARY KEY AUTO_INCREMENT,  
    nome VARCHAR(200),  
    sexo CHAR(1),  
    tipo VARCHAR(50)  
);
```

```
CREATE TABLE Funcionario (  
    id_funcionario INT PRIMARY KEY AUTO_INCREMENT,  
    nome VARCHAR(255),  
    cargo VARCHAR(100),  
    telefone VARCHAR(20)  
);
```

```
CREATE TABLE Servico (  
    id_servico INT PRIMARY KEY AUTO_INCREMENT,  
    nome VARCHAR(255),  
    descricao TEXT,  
    preco DECIMAL(10, 2)  
);
```

```
CREATE TABLE Produto (  
    id_produto INT PRIMARY KEY AUTO_INCREMENT,  
    nome VARCHAR(255),  
    descricao TEXT,  
    preco DECIMAL(10, 2),  
    qtd_estoque INT  
);
```

```
CREATE TABLE Atendimento (  
    id_atendimento INT PRIMARY KEY AUTO_INCREMENT,  
    data DATE,
```

```
id_cliente INT,  
id_animal INT,  
id_funcionario INT,  
id_servico INT,  
FOREIGN KEY (id_cliente) REFERENCES Cliente(id_cliente),  
FOREIGN KEY (id_animal) REFERENCES Animal(id_animal),  
FOREIGN KEY (id_funcionario) REFERENCES Funcionario(id_funcionario),  
FOREIGN KEY (id_servico) REFERENCES Servico(id_servico)  
);  
  
CREATE TABLE Vendas (  
    id_venda INT PRIMARY KEY AUTO_INCREMENT,  
    id_atendimento INT,  
    id_produto INT, -- Relacionamento com Produto  
    data_venda DATE,  
    FOREIGN KEY (id_atendimento) REFERENCES Atendimento(id_atendimento),  
    FOREIGN KEY (id_produto) REFERENCES Produto(id_produto)  
);
```

## 6. DIAGRAMA ENTIDADE-RELACIONAMENTO (ER)

### Entidade: Cliente

- Atributo Primário:
- id\_cliente (INT, PRIMARY KEY)
- Atributos Secundários:
- nome (VARCHAR(100))
- endereco (VARCHAR(200))
- telefone (VARCHAR(20))

### Entidade: Animal

- Atributo Primário:
- id\_animal (INT, PRIMARY KEY)
- Atributos Secundários:
- nome (VARCHAR(100))
- especie (VARCHAR(50))
- raca (VARCHAR(50))
- data\_nascimento (DATE)
- id\_cliente (INT, FOREIGN KEY referenciando Cliente(id\_cliente))

### Entidade: Funcionário

- Atributo Primário:
- id\_funcionario (INT, PRIMARY KEY)
- Atributos Secundários:
- nome (VARCHAR(100))
- cargo (VARCHAR(50))
- data\_contratacao (DATE)

### Entidade: Produto

- Atributo Primário:
- id\_produto (INT, PRIMARY KEY)
- Atributos Secundários:
- nome (VARCHAR(100))
- descricao (TEXT)
- preco (DECIMAL(10, 2))
- quantidade\_estoque (INT)

### Entidade: Serviço

- Atributo Primário:
- id\_servico (INT, PRIMARY KEY)
- Atributos Secundários:
- nome (VARCHAR(100))
- descricao (TEXT)
- preco (DECIMAL(10, 2))

### Entidade: Atendimento

- Atributo Primário:
- id\_atendimento (INT, PRIMARY KEY)
- Atributos Secundários:

- data (DATE)
- id\_cliente (INT, FOREIGN KEY referenciando Cliente(id\_cliente))
- id\_animal (INT, FOREIGN KEY referenciando Animal(id\_animal))
- id\_funcionario (INT, FOREIGN KEY referenciando Funcionário(id\_funcionario))
- id\_produto (INT, FOREIGN KEY referenciando Produto(id\_produto))
- id\_servico (INT, FOREIGN KEY referenciando Serviço(id\_servico))

## Explicação das Cardinalidades:

As cardinalidades no diagrama ER são explicadas da seguinte forma:

- Cliente (1,N) Animal: Um cliente pode possuir vários animais, mas um animal pertence a um único cliente.
- Cliente (1,N) Atendimento: Um cliente pode solicitar vários atendimentos, mas cada atendimento está associado a um único cliente.
- Animal (1,N) Atendimento: Um animal pode receber vários atendimentos, mas cada atendimento envolve apenas um animal.
- Funcionário (1,N) Atendimento: Um funcionário pode realizar vários atendimentos, mas cada atendimento é realizado por um único funcionário.
- Produto (1,N) Atendimento: Um produto pode ser utilizado em vários atendimentos, mas cada atendimento pode utilizar vários produtos.
- Serviço (1,N) Atendimento: Um serviço pode ser prestado em vários atendimentos, mas cada atendimento pode incluir vários serviços.

## 7. CONCLUSÃO

Este projeto nos proporcionou uma valiosa experiência na manipulação e desenvolvimento de bancos de dados para diversos segmentos. Com isso, conseguimos abstrair os processos necessários para a execução eficiente das atividades diárias em um petshop, modelo que pode ser adaptado para outros tipos de negócios.

O modelo de banco de dados, cuidadosamente desenvolvido, pode ser integrado a vários sistemas. A implementação de backups diários no petshop contribui para um controle eficiente dos atendimentos, gerenciamento de estoque e permite a realização de análises detalhadas dos perfis dos clientes, otimizando todo o processo da empresa.

As informações geradas por este projeto possibilitam o desenvolvimento de estratégias mais refinadas em diversos setores, garantindo melhores resultados financeiros para as empresas.





## 8. REFERÊNCIAS

Documentos do SQL, 2021 Tabelas. Disponível em:<<https://learn.microsoft.com/pt-br/sql/relationaldatabases/tables/tables?view=sql-server-ver15>>. Acesso em: 01 de maio de 2024

Oracle Brasil, 2021 Banco de dados definido. Disponível em:<<https://www.oracle.com/br/database/what-is-database/#link1>>. Acesso em: 01 de maio de 2024

Qual é a diferença entre um banco de dados e uma planilha? Disponível em:<<https://www.oracle.com/br/database/what-is-database/#link4>>. Acesso em: 01 de maio de 2024

Bruno Rafael, 2019 SQL, NoSQL, NewSQL: Qual banco de dados usar? Disponível em:<<https://blog.geekhunter.com.br/sql-nosql-newsql-qual-banco-de-dados-usar/>>. Acesso em: 01 de maio de 2024