



UNIVERSIDADE FEDERAL DE SANTA CATARINA  
CAMPUS ARARANGUÁ  
DEPARTAMENTO DE ENGENHARIA DA COMPUTAÇÃO  
CURSO ENGENHARIA DA COMPUTAÇÃO

Eduardo Zambotto da Silva

João Pedro Blanco

Júlio César Pereira Amaral

**Sistema de Gerenciamento de Pet Shop**

Trabalho Final

Matéria: Banco de Dados I (DEC7129)

Professor: Alexandre Leopoldo Gonçalves

Araranguá

2024

## SUMÁRIO

<b>1 INTRODUÇÃO</b>	<b>3</b>
1.1 DESCRIÇÃO DETALHADA:	4
Requisitos Específicos	5
<b>2 DESENVOLVIMENTO</b>	<b>6</b>
2.1 modelo conceitual	6
2.2 modelo lógico	6
2.3 script ddl	7
2.4 consultas	11
2.5 inteligência artificial generativa	15
<b>3 CONCLUSÃO</b>	<b>17</b>

## **1 INTRODUÇÃO**

O sistema desenvolvido no trabalho tem por objetivo propor uma solução eficiente para Pet Shops de todas as faixas de faturamento, mas tendo foco nos de pequeno e médio porte. A solução inclui um banco de dados relacional(PostgreSQL), uma aplicação para gerenciá-lo e um sistema de classificação que faz uso de inteligência artificial generativa, sendo ambos os códigos desenvolvidos na linguagem de programação python.

## 1.1 DESCRIÇÃO DETALHADA:

Uma empresa especializada no desenvolvimento de soluções, serviços e produtos para pet shops deseja desenvolver um sistema de banco de dados para otimizar suas operações. O objetivo é organizar informações sobre clientes, funcionários, animais, serviços, produtos e transações comerciais. Os seguintes requisitos devem ser levados em consideração:

1. Cada funcionário deve possuir identificador único, CPF e nome.
2. Os funcionários devem ser divididos em tipos, sendo estes tipos suas atribuições, que devem conter descrição e salário.
3. Cada cliente possui um identificador, nome, telefone e endereço.
4. Animais possuem identificador, nome, peso e estão associados a uma raça.
5. Um cliente pode possuir vários animais, mas cada animal pertence a um único cliente.
6. Note que cada animal está associado a uma raça, cada raça é identificada por um nome único e pertence a uma espécie. Espécies possuem identificador e nome.
7. Produtos que serão usados nos atendimentos possuem identificador, nome, quantidade em estoque, valor, e estão associados a um tipo e a um fornecedor. O tipo de produto é identificado por uma descrição única.
8. Fornecedores possuem identificador, nome e e-mail.
9. Serviços(atendimentos) são definidos por um identificador, descrição, valor, e estão associados a um tipo de serviço. O tipo de serviço deve possuir uma descrição única.
10. Cada compra é registrada em uma nota de compra com identificador, valor total, data e o funcionário responsável pela compra. As notas de compra incluem vários produtos, cada um com uma quantidade associada.
11. As vendas são registradas em notas de venda que possuem identificador, data, valor total e o cliente responsável pela compra.
12. Um atendimento é associado a uma nota de venda e inclui informações sobre o serviço realizado, o animal atendido, a data do atendimento, o funcionário

responsável e o valor total do atendimento. Produtos utilizados durante o atendimento são registrados com suas respectivas quantidades.

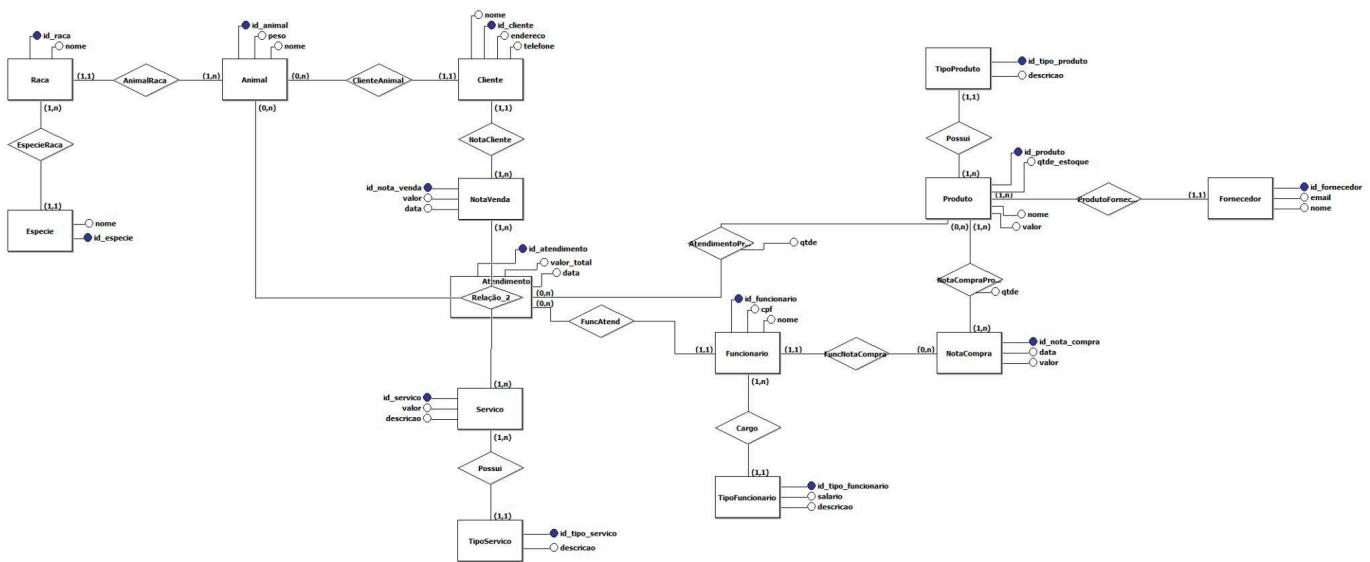
### **Requisitos Específicos**

1. Garantir unicidade para atributos importantes como CPF de funcionários, nomes de produtos, serviços e raças.
2. Todas as relações entre entidades devem ser representadas com chaves estrangeiras para garantir integridade referencial.
3. As operações do petshop devem permitir consultas sobre histórico de atendimentos, consumo de produtos, e movimentações financeiras relacionadas a vendas e compras.

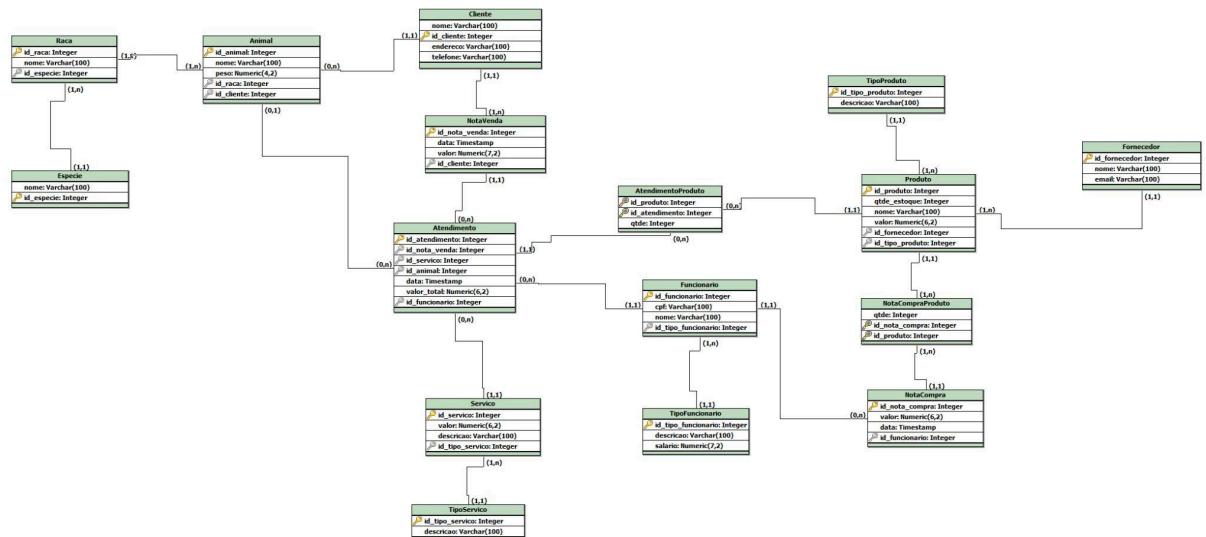
## 2 DESENVOLVIMENTO

O desenvolvimento foi dividido em algumas etapas, sendo elas, requisitos do trabalho. São estes, modelo conceitual e lógico, script DDL, consultas e representação gráfica das mesmas e uma solução que faça uso de algum recurso de inteligência artificial generativa.

### 2.1 MODELO CONCEITUAL



### 2.2 MODELO LÓGICO



## 2.3 SCRIPT DDL

```
CREATE TABLE Fornecedor (  
    id_fornecedor Integer PRIMARY KEY,  
    nome Varchar(100) NOT NULL,  
    email Varchar(100) NOT NULL  
);
```

```
CREATE TABLE TipoFuncionario (  
    id_tipo_funcionario Integer PRIMARY KEY,  
    descricao Varchar(100) NOT NULL,  
    salario Numeric(7,2) NOT NULL  
);
```

```
CREATE TABLE TipoServico (  
    id_tipo_servico Integer PRIMARY KEY,  
    descricao Varchar(100) NOT NULL  
);
```

```
CREATE TABLE TipoProduto (  
    id_tipo_produto Integer PRIMARY KEY,  
    descricao Varchar(100) NOT NULL  
);
```

```
CREATE TABLE Especie (  
    nome Varchar(100) NOT NULL,  
    id_especie Integer PRIMARY KEY  
);
```

```
CREATE TABLE Cliente (  
    nome Varchar(100) NOT NULL,  
    id_cliente Integer PRIMARY KEY,  
    endereco Varchar(100),  
    telefone Varchar(100)  
);
```

```
CREATE TABLE NotaCompra (  
    id_nota_compra Integer PRIMARY KEY,  
    valor Numeric(6,2) NOT NULL,  
    data Timestamp NOT NULL,  
    id_funcionario Integer,  
    FOREIGN KEY(id_funcionario) REFERENCES Funcionario (id_funcionario)  
);
```

```
CREATE TABLE Raca (  
    id_raca Integer PRIMARY KEY,  
    nome Varchar(100) NOT NULL,  
    id_especie Integer,  
    FOREIGN KEY(id_especie) REFERENCES Especie (id_especie)  
);
```

```
CREATE TABLE NotaVenda (  
    id_nota_venda Integer PRIMARY KEY,  
    data Timestamp NOT NULL,  
    valor Numeric(7,2) NOT NULL,  
    id_cliente Integer,  
    FOREIGN KEY(id_cliente) REFERENCES Cliente (id_cliente)  
);
```

```
CREATE TABLE Animal (  
    id_animal Integer PRIMARY KEY,  
    nome Varchar(100) NOT NULL,  
    peso Numeric(4,2) NOT NULL,  
    id_raca Integer,  
    id_cliente Integer,  
    FOREIGN KEY(id_raca) REFERENCES Raca (id_raca),  
    FOREIGN KEY(id_cliente) REFERENCES Cliente (id_cliente)  
);
```



```

CREATE TABLE Produto (
    id_produto Integer PRIMARY KEY,
    qtde_estoque Integer NOT NULL,
    nome Varchar(100) NOT NULL,
    valor Numeric(6,2) NOT NULL,
    id_fornecedor Integer,
    id_tipo_produto Integer,
    FOREIGN KEY(id_fornecedor) REFERENCES Fornecedor (id_fornecedor),
    FOREIGN KEY(id_tipo_produto) REFERENCES TipoProduto (id_tipo_produto)
);

```

```

CREATE TABLE Funcionario (
    id_funcionario Integer PRIMARY KEY,
    cpf Varchar(100) NOT NULL,
    nome Varchar(100) NOT NULL,
    id_tipo_funcionario Integer,
    FOREIGN KEY(id_tipo_funcionario) REFERENCES TipoFuncionario
(id_tipo_funcionario)
);

```

```

CREATE TABLE Servico (
    id_servico Integer PRIMARY KEY,
    valor Numeric(6,2),
    descricao Varchar(100) NOT NULL,
    id_tipo_servico Integer,
    FOREIGN KEY(id_tipo_servico) REFERENCES TipoServico (id_tipo_servico)
);

```

```

CREATE TABLE NotaCompraProduto (
    qtde Integer NOT NULL,
    id_nota_compra Integer,
    id_produto Integer,
    PRIMARY KEY(id_nota_compra, id_produto),
    FOREIGN KEY(id_nota_compra) REFERENCES NotaCompra (id_nota_compra),

```

```

FOREIGN KEY(id_produto) REFERENCES Produto (id_produto)
);

```

```

CREATE TABLE Atendimento (
    id_atendimento Integer PRIMARY KEY,
    id_nota_venda Integer,
    id_servico Integer,
    id_animal Integer,
    data Timestamp NOT NULL,
    valor_total Numeric(6,2) NOT NULL,
    id_funcionario Integer,
    FOREIGN KEY(id_nota_venda) REFERENCES NotaVenda (id_nota_venda),
    FOREIGN KEY(id_servico) REFERENCES Servico (id_servico),
    FOREIGN KEY(id_animal) REFERENCES Animal (id_animal),
    FOREIGN KEY(id_funcionario) REFERENCES Funcionario (id_funcionario)
);

```

```

CREATE TABLE AtendimentoProduto (
    qtde Integer NOT NULL,
    id_produto Integer,
    id_atendimento Integer,
    PRIMARY KEY(id_produto, id_atendimento),
    FOREIGN KEY(id_produto) REFERENCES Produto (id_produto),
    FOREIGN KEY(id_atendimento) REFERENCES Atendimento (id_atendimento)
);

```

```

ALTER TABLE NotaCompra ADD FOREIGN KEY(id_funcionario) REFERENCES
Funcionario (id_funcionario);

```

```

ALTER TABLE Raca
ADD CONSTRAINT raca_unico UNIQUE (nome);

```

```

ALTER TABLE Especie
ADD CONSTRAINT especie_unico UNIQUE (nome);

```

```
ALTER TABLE TipoProduto
ADD CONSTRAINT tipo_produto_unico UNIQUE (descricao);
```

```
ALTER TABLE Produto
ADD CONSTRAINT produto_unico UNIQUE (nome);
```

```
ALTER TABLE TipoServico
ADD CONSTRAINT tipo_servico_unico UNIQUE (descricao);
```

```
ALTER TABLE Servico
ADD CONSTRAINT servico_unico UNIQUE (descricao);
```

```
ALTER TABLE TipoFuncionario
ADD CONSTRAINT tipo_funcionario_unico UNIQUE (descricao);
```

```
ALTER TABLE Funcionario
ADD CONSTRAINT funcionario_unico UNIQUE (cpf);
```

```
ALTER TABLE Atendimento
ADD CONSTRAINT atendimento_unico UNIQUE (id_nota_venda, id_servico,
id_animal, id_funcionario);
```

```
ALTER TABLE Atendimento
DROP CONSTRAINT atendimento_unico;
```

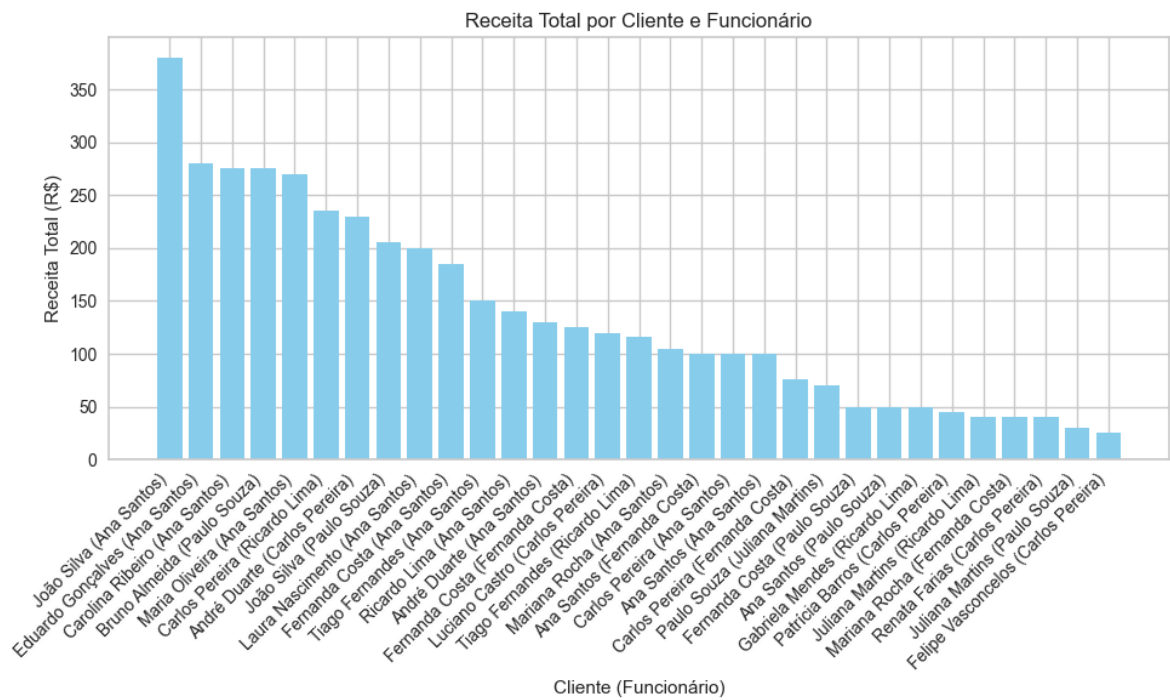
## 2.4 CONSULTAS

### 2.4.1

**Descrição da consulta:** Receita Total por Cliente e Funcionário.

Nome do Cliente	Nome do Funcionário	Receita Total (R\$)
João Silva	Ana Santos	380.00
Eduardo Gonçalves	Ana Santos	280.00
Carolina Ribeiro	Ana Santos	275.00

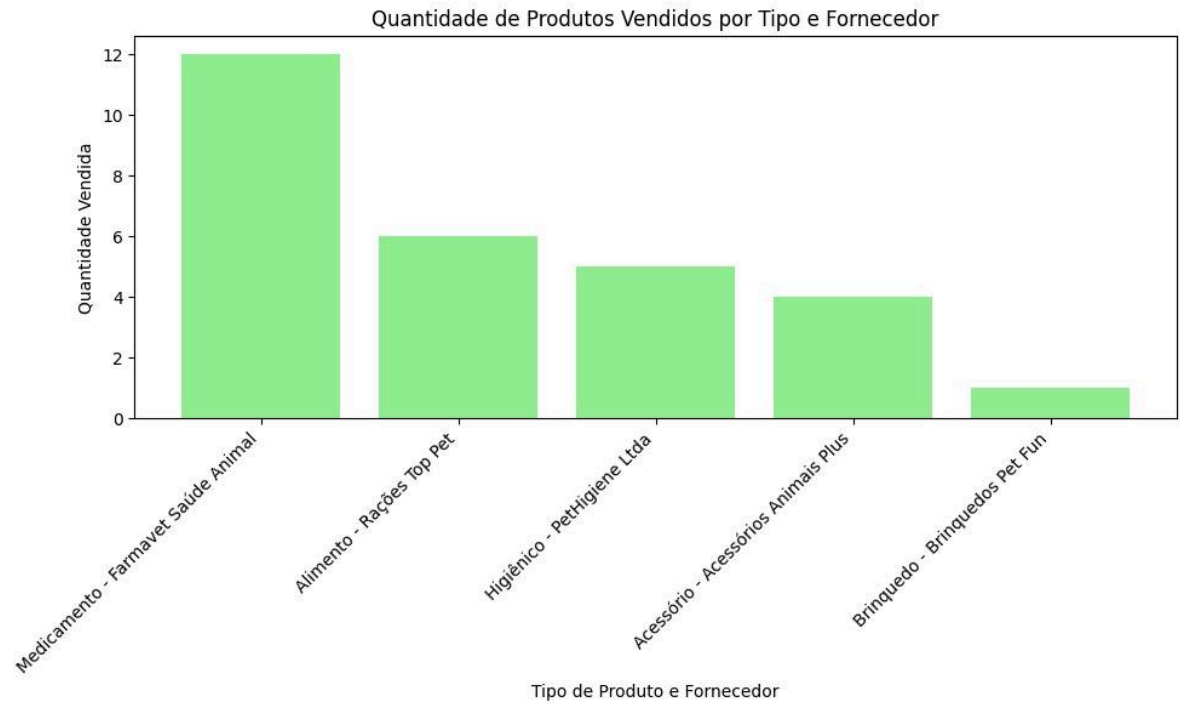
Bruno Almeida	Paulo Souza	275.00
Maria Oliveira	Ana Santos	270.00
Carlos Pereira	Ricardo Lima	235.00
André Duarte	Carlos Pereira	230.00
João Silva	Paulo Souza	205.90
Laura Nascimento	Ana Santos	200.00
Fernanda Costa	Ana Santos	185.00
Tiago Fernandes	Ana Santos	150.00
Ricardo Lima	Ana Santos	140.00
André Duarte	Ana Santos	130.00
Fernanda Costa	Fernanda Costa	125.00
Luciano Castro	Carlos Pereira	120.00
Tiago Fernandes	Ricardo Lima	115.90
Mariana Rocha	Ana Santos	105.00
Ana Santos	Fernanda Costa	100.00
Carlos Pereira	Ana Santos	100.00
Ana Santos	Ana Santos	100.00
Carlos Pereira	Fernanda Costa	75.90
Paulo Souza	Juliana Martins	70.00
Fernanda Costa	Paulo Souza	50.00
Ana Santos	Paulo Souza	50.00
Gabriela Mendes	Ricardo Lima	50.00
Patrícia Barros	Carlos Pereira	45.00
Juliana Martins	Ricardo Lima	40.00
Mariana Rocha	Fernanda Costa	40.00
Renata Farias	Carlos Pereira	40.00
Juliana Martins	Paulo Souza	30.00
Felipe Vasconcelos	Carlos Pereira	25.00



2.4.2

**Descrição consulta:** Quantidade de Produtos Vendidos por Tipo e Fornecedor.

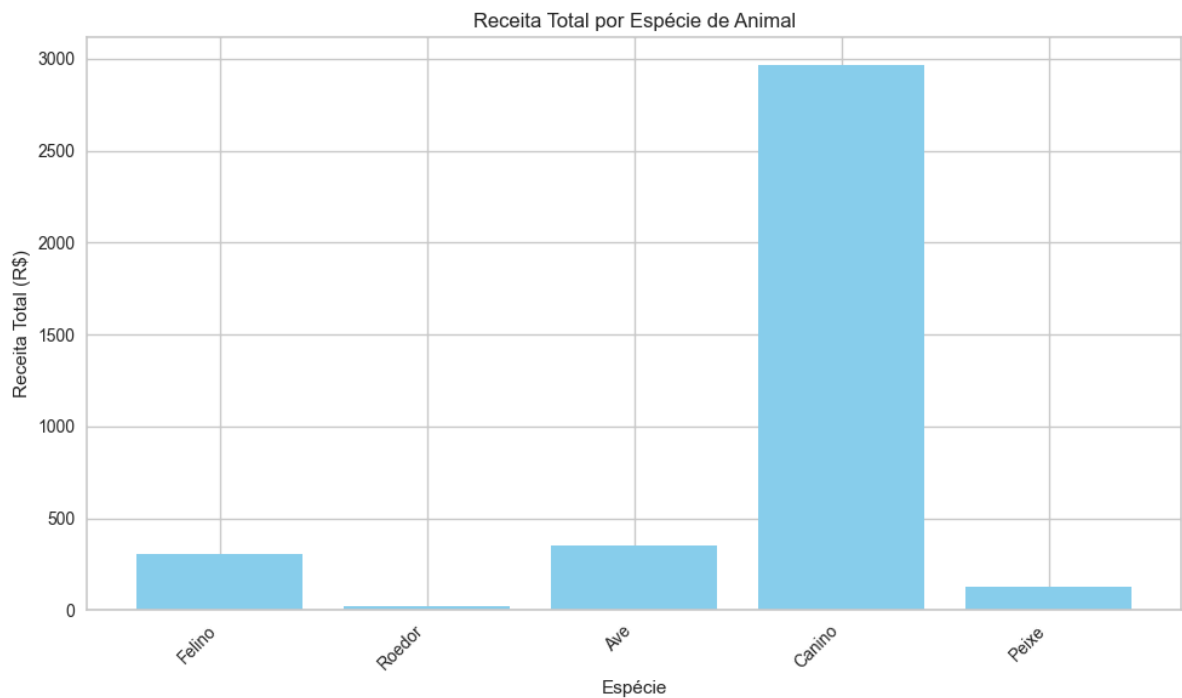
Tipo de Produto	Nome do Fornecedor	Total Vendido
Medicamento	Farmavet Saúde Animal	12
Alimento	Rações Top Pet	6
Higiênico	PetHigiene Ltda	5
Acessório	Acessórios Animais Plus	4
Brinquedo	Brinquedos Pet Fun	1



2.4.3

**Descrição consulta:** Receita Total por Espécie de Animal.

Espécie	Receita Total (R\$)
Felino	305.00
Roedor	20.00
Ave	355.00
Canino	2967.70
Peixe	130.00



## 2.5 INTELIGÊNCIA ARTIFICIAL GENERATIVA

O sistema implementa um algoritmo k means, a fim de gerar grupos de uma dada tabela, com o objetivo de tomada de decisão. Neste caso, a escolhida foi o atendimento. A metodologia visava encontrar padrões em grupos de atendimento, visando encontrar o tipo de serviço prestado que era mais rentável para o pet shop.

O processo consistia de algumas etapas, a primeira delas, converter nossos dados em um vetor denso, convertendo cada tupla em um vetor denso, e, posteriormente, formando um espaço n dimensional destes vetores gerados com a biblioteca sentence transformers. Tendo nosso novo conjunto de dados, foi feito o uso da biblioteca yellowbrick, para definir o número de clusters(4, no nosso caso).

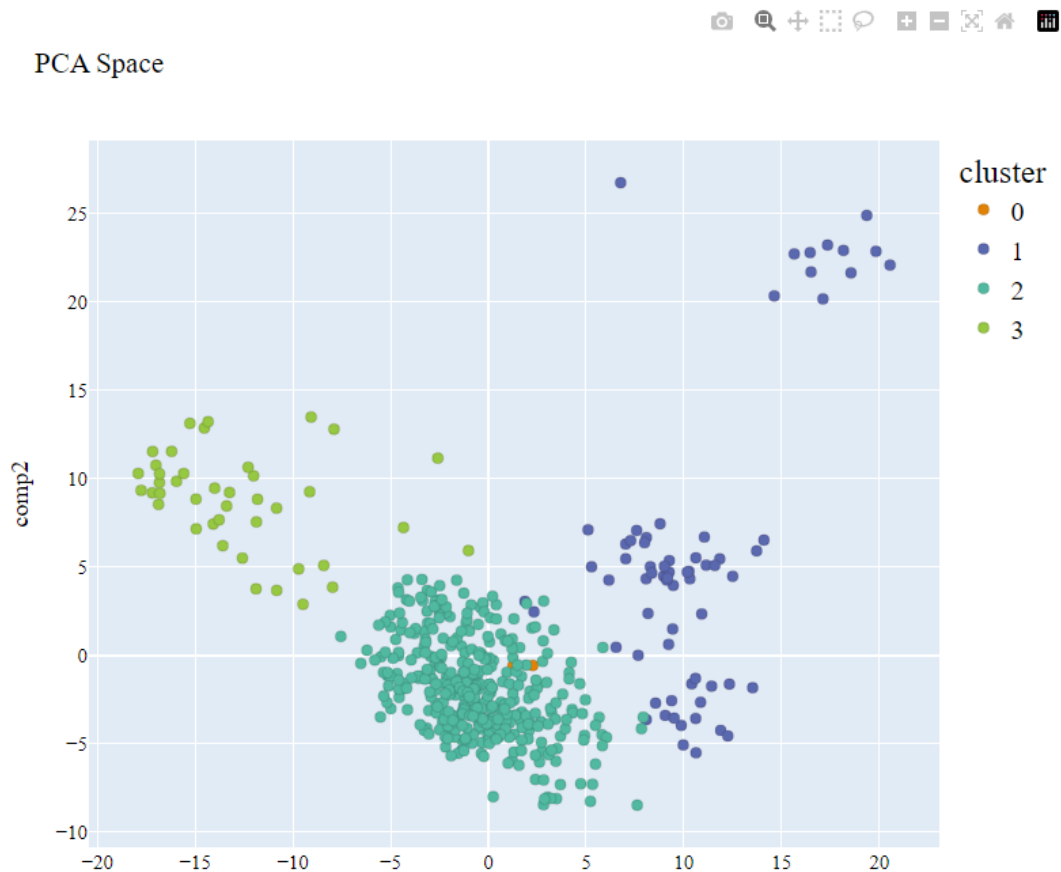


Imagem dos cluster mostrados em 2D(com ajuda de PCA)

Por fim, foi feito uso do gemini(LLM do Google) para identificar quais foram os critérios que o algoritmo k means adotou para gerar os grupos de forma não supervisionada. E este foi o resultado:

#### Inferências e Justificativas:

- **valor\_total\_atendimento:** A influência desta variável é óbvia. O k-means busca minimizar a variância intra-cluster, ou seja, agrupa pontos próximos em termos de distância euclidiana. Dados com valores totais de atendimento similares tenderão a ser agrupados no mesmo cluster.
- **descricao\_servico:** A variável categórica "descricao\_servico" representa diferentes tipos de serviços oferecidos. Se o petshop oferece serviços com preços e durações significativamente diferentes, o k-means provavelmente criou clusters que refletem a natureza desses serviços. Por exemplo, serviços de banho e tosa provavelmente formariam um cluster diferente de consultas veterinárias de emergência, dado o custo e o tempo envolvidos.



**Faixas dos Clusters:**

- Cluster 1 (serviços de alta complexidade/custo): Consultas veterinárias de emergência, cirurgias, internações (se existirem no dataset). Valor\_total\_atendimento alto.
- Cluster 2 (serviços diversos e custos intermediários): Uma mistura de serviços, como consultas gerais, exames (sangue, parasitológicos), aplicação de vacinas, etc. Valor\_total\_atendimento intermediário.
- Cluster 3 (serviços de menor custo/simples): Banhos, tosas higiênicas, aparação de unhas e venda de produtos. Valor\_total\_atendimento baixo.
- Cluster 0 (diárias): Diárias básicas ou com recreação. O valor\_total\_atendimento pode variar de acordo com a duração, porém seria um grupo separado devido à natureza do serviço.

OBS: É importante salientar que isto é apenas parte do output do LLM, e ele não deu 100% de certeza na sua resposta, tratando-a como hipotética, apesar da mesma ter sido baseada nos dados fornecidos.

**3 CONCLUSÃO**

O sistema proposto visa atender a demanda de pet shops de pequeno e médio porte, e mostra-se uma solução promissora para otimizar a gestão do público alvo, haja visto a abrangência do sistema e sua escalabilidade. Apesar do público alvo, a solução é robusta, sendo baseada em um modelo relacional, e respeitando as normas padrões, e a interface em Python torna seu uso mais intuitivo.

Além disso, a integração de inteligência artificial generativa traz um diferencial inovador, permitindo a análise e classificação de dados de forma inteligente, o que pode auxiliar os gestores na tomada de decisões estratégicas.

Com essa solução, espera-se potencializar o sucesso operacional e financeiro dos pet shops, ajudando-os a alcançar novos patamares de excelência em seus negócios.