



Universidade Federal De Mato Grosso Do Sul  
Campus de Três Lagoas  
Bacharelado em Sistemas de Informação  
Banco de Dados

## **Relatório Técnico**

**Tema: Sistemas para Gerenciamento de ONGs.**

**Título: Patas Urbanas - Projeto de Banco de Dados para uma clínica.**

**Professora: Ivone Penque Matsuno Yugoshi**

**Breno Hideki Utimura**

**Gabriel Fiorentino Bressane**

**Três Lagoas - MS**

**2025**

## Sumário

<b>1. Introdução.....</b>	<b>4</b>
<b>1.1 Estudo de Caso.....</b>	<b>4</b>
<b>2. Esquema Conceitual.....</b>	<b>6</b>
2.1 Diagrama Entidade-Relacionamento.....	6
2.1 Diagrama Relacional.....	7
<b>3. Esquema Lógico.....</b>	<b>7</b>
3.1 Mapeamento da ONG Patas Urbanas.....	7
3.2 Domínio.....	8
Tabela Adoção.....	8
Tabela Adotante.....	9
Tabela Ajuda animal.....	9
Tabela Alimento.....	9
Tabela Animal.....	9
Tabela Clínica Veterinária.....	10
Tabela Consulta Animal.....	10
Tabela Dinheiro.....	11
Tabela EmailClinica.....	12
Tabela EmailPessoa.....	12
Tabela Funcionário.....	12
Tabela Gerencia.....	13
Tabela Objeto.....	13
Tabela Pessoa.....	13
Tabela Recurso.....	14
Tabela TelefoneClinica.....	14
Tabela TelefonePessoa.....	15
Tabela Vacina.....	15
Tabela Vacinacao.....	15
Tabela Veterinario.....	16
Tabela VeterinarioTrabalha.....	16
Tabela Voluntário.....	17
<b>4. Scripts de Criação do Banco de Dados e Tabelas.....</b>	<b>17</b>
4.1 Criando DataBase.....	17
4.2 Criando Schema.....	17
4.3 Script Criação Tabelas.....	17
Tabela Adocao.....	18
Tabela Adotante.....	18
Tabela AjudaAnimal.....	18
Tabela Alimento.....	18
Tabela Animal.....	19
Tabela ClinicaVeterinaria.....	19

Tabela ConsultaAnimal.....	19
Tabela Dinheiro.....	20
Tabela EmailClinica.....	20
Tabela EmailPessoa.....	20
Tabela Funcionario.....	21
Tabela Gerencia.....	21
Tabela Objeto.....	21
Tabela Pessoa.....	22
Tabela Recurso.....	22
Tabela TelefonePessoa.....	22
Tabela TelefoneClinica.....	23
Tabela Vacina.....	23
Tabela Voluntario.....	23
Tabela Vacinacao.....	24
Tabela Veterinario.....	24
Tabela VeterinarioTrabalha.....	24
<b>5. Scripts de Inserção, Atualização e Remoção do Banco de Dados.....</b>	<b>24</b>
5.1 Inserção.....	25
5.2 Atualização.....	25
5.3 Remoção.....	26
<b>6. Consultas.....</b>	<b>27</b>
6.1 Exibir os nomes e CPFs dos voluntários:.....	27
6.2 Listar todos os animais adotados, com nome do animal, nome do adotante e data de adoção:.....	27
6.3 Mostrar o nome dos funcionários e o total de recursos que cada um gerencia:.....	28
6.4 Listar todos os animais que já receberam vacinação, indicando a vacina e a data:.....	28
6.5 Quantidade de animais consultados por cada veterinário.....	29
6.6 Total de dinheiro arrecadado no ano de 2025:.....	29
6.7 Mostrar o nome da clínica e quantidade distinta de animais já atendidos nela (consultas realizadas):.....	30
6.8 Animais que nunca foram adotados:.....	30
6.9 Quantidade de supervisionados por supervisor.....	31
6.10 Total de funcionários distintos, quantos estão trabalhando e quantos estão parados:.....	31

## 1. Introdução

As ONGs exercem um papel fundamental na sociedade, agindo em diversas áreas como assistência social, meio ambiente, educação, saúde e defesa dos direitos humanos. Apesar da importância do seu trabalho, muitas dessas organizações ainda enfrentam limitações tecnológicas que dificultam a gestão adequada de informações, recursos e pessoas envolvidas em suas atividades.

Criar um sistema de banco de dados voltado para o gerenciamento de ONGs permite estruturar e armazenar dados essenciais. Essa estrutura facilita o controle interno, auxilia na geração de relatórios e indicadores, e torna a tomada de decisões mais eficiente. Além disso, um sistema bem construído contribui para a transparência nas operações.

Ao aplicar tecnologia na gestão de ONGs, promove-se não apenas a modernização dessas entidades, mas também a ampliação do impacto social que elas são capazes de gerar. Tais itens serão levados em consideração para a realização deste trabalho.

### 1.1 Estudo de Caso

O **Projeto Patas Urbanas** é uma iniciativa de uma ONG brasileira voltada para o resgate, cuidado e controle de animais de rua nas áreas urbanas. Essa ONG atua principalmente no acolhimento de cães e gatos abandonados, oferecendo alimentação, tratamento médico, castração, e, promovendo adoções responsáveis. Diante da crescente demanda e da complexidade do trabalho realizado, tornou-se necessária a criação de um sistema de banco de dados para organizar, registrar e gerir todas as informações envolvidas nas operações do projeto.

O sistema proposto visa atender de forma eficiente às necessidades de registro e controle de animais resgatados, voluntários, funcionários, clínicas parceiras e pessoas interessadas em adotar. Para isso, são coletadas informações básicas de cada animal, como nome, espécie, raça, porte, sexo, estado de saúde, se é castrado e a data de entrada na ONG.

Os animais são atendidos e tratados por clínicas veterinárias parceiras, que realizam os procedimentos médicos conforme a necessidade de cada caso. Sobre

as clínicas, são armazenadas informações como CNPJ, nome, endereço, contato e especialidades. Cada clínica possui um ou mais veterinários, que armazenam, além das informações básicas, o CFMV e a especialidade. Cada animal pode ser atendido por mais de uma clínica ao longo do seu processo de reabilitação. Durante a consulta, um veterinário e um voluntário ficam responsáveis em acompanhar o animal e é armazenado a data, descrição e prescrição.

Além disso, a ONG tem o controle de todas as vacinas aplicadas em cada animal, essas vacinas são aplicadas em clínicas veterinárias parceiras. É armazenado de vacina, o nome, lote, via de aplicação, data de validade e o código.

A ONG conta com o apoio de voluntários, responsáveis por tarefas como resgate, transporte e cuidado diário. Para cada voluntário são registradas as informações pessoais (nome, CPF, telefone, e-mail, data de nascimento), além da área de atuação e data de ingresso na ONG. Um voluntário pode monitorar outros voluntários.

Além dos voluntários, a ONG conta com funcionários contratados que atuam no gerenciamento dos recursos da ONG. Para cada funcionário contratado, o sistema armazena informações pessoais, sua remuneração, horas de trabalho, cargo, data de contratação e o tipo de contratos. Diferentemente dos voluntários, somente os contratados têm permissão para controlar os recursos da ONG, garantindo um gerenciamento mais seguro e eficiente.

As pessoas interessadas em adotar têm seus dados registrados no sistema, incluindo nome completo, CPF, telefone e endereço. Para as adoções, são armazenadas informações como data e alguma observação.

Os recursos possuem identificação única, descrição, origem do recurso, quantidade, e são classificados como dinheiro, alimento e objeto. Essa classificação permite um controle mais preciso sobre a aplicação de cada recurso recebido e garante que os recursos sejam destinados corretamente conforme as necessidades da ONG.

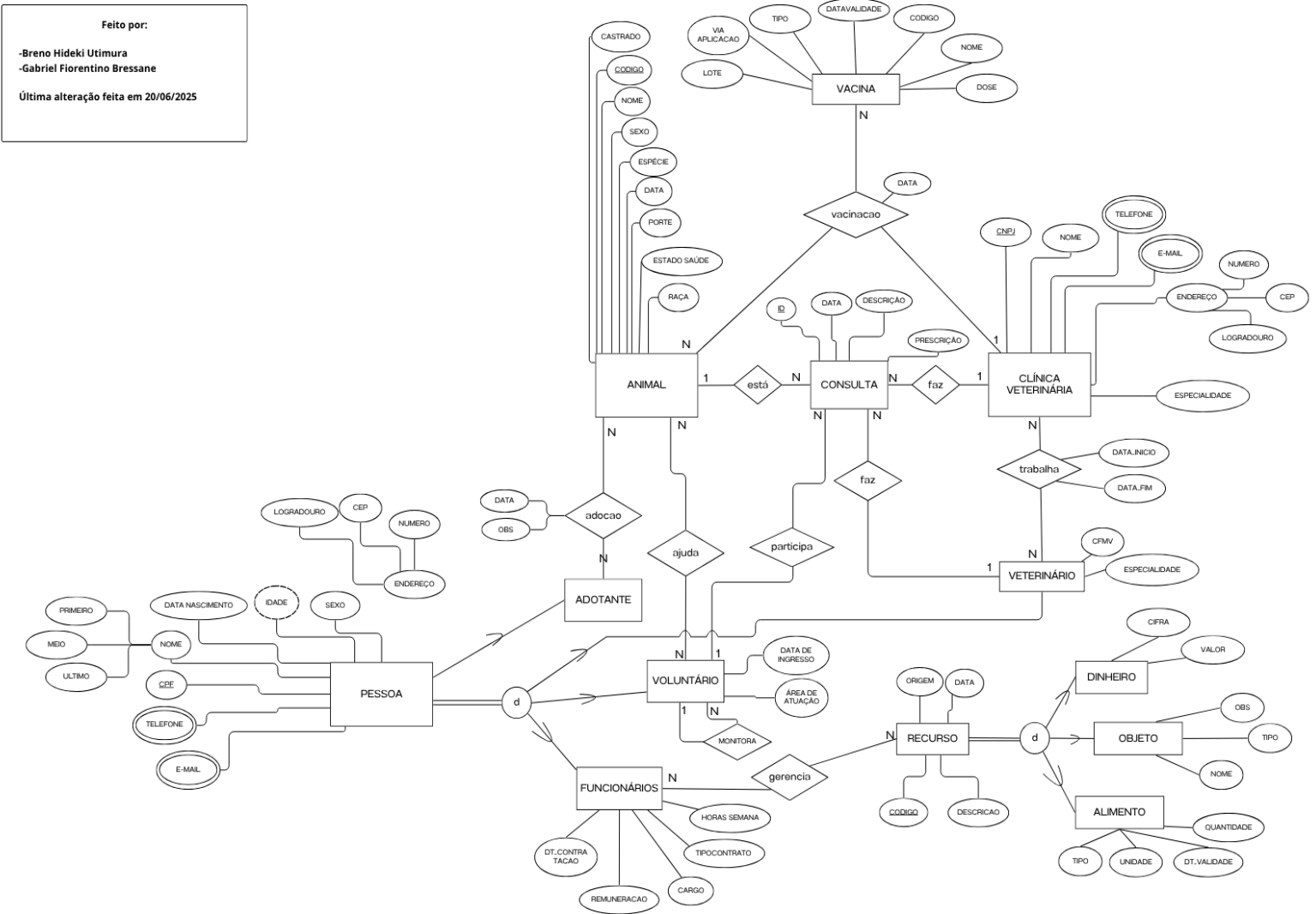
Com esse sistema, o Projeto Patas Urbanas pode manter um controle eficiente de suas atividades, garantindo o controle de cada ação, a organização de

dados importantes e a otimização de recursos. Dessa forma, a ONG fortalece sua atuação e amplia o impacto positivo nas comunidades urbanas e promovendo bem-estar animal.

## 2. Esquema Conceitual

### 2.1 Diagrama Entidade-Relacionamento

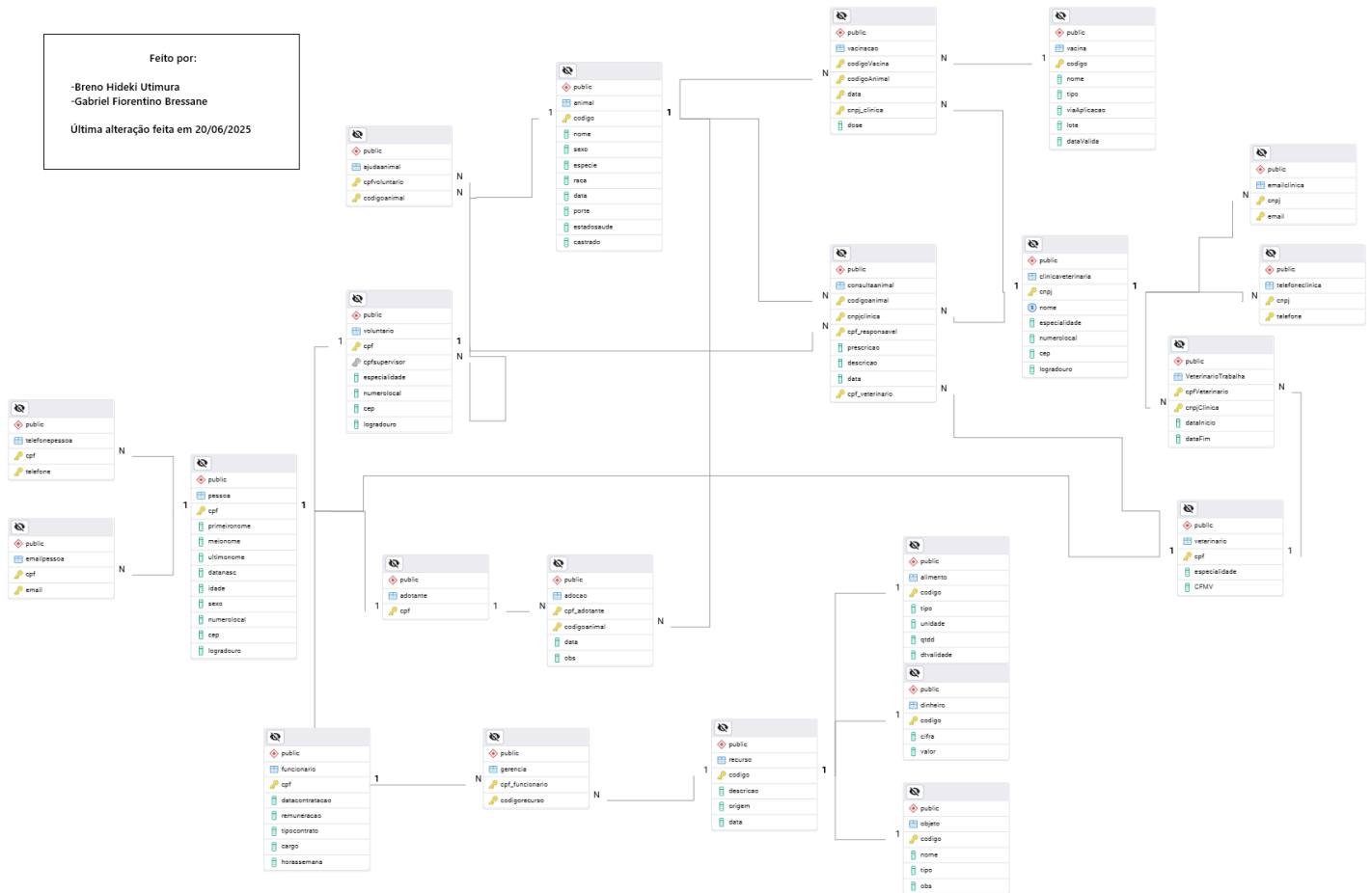
Figura 1 - DER Patas Urbanas



Fonte: Autoria própria

## 2.1 Diagrama Relacional

Figura 2 - Diagrama Relacional



Fonte: Autoria própria

### 3. Esquema Lógico

### 3.1 Mapeamento da ONG Patas Urbanas

```
Animal (codigo, nome, sexo, especie, data, porte, estadoSaude, raca,
castrado);
```

Vacina (codigo, nome, dataValidade, tipo, viaAplicacao, lote)

Clínica Veterinária (CNPJ, nome, especialidade, numeroLocal, cep, logradouro);

Veterinário(#CPF, especialidade, CFMV);

Pessoa (CPF, primeiroNome, meioNome, ultimoNome ,dataNasc, idade, sexo, numeroLocal, cep, logradouro);

Recurso (codigo, descricao, origem, data);

Dinheiro (#codigo, cifra, origem, valor);

Alimento (#codigo, tipo, unidade, dtValidade, quantidade);

Objeto (#codigo, nome, obs, tipo);

Voluntário (dataIngresso, areaAtuacao, #CPF, #CPFsupervisor);

Adotante (#CPF);

Funcionário (#CPF, dataContratação, remuneração, tipoContrato, cargo, horasSemana);

Consulta Animal (codigo, data, descricao, prescricao, #codigoAnimal, #CNPJclinica, #CPF\_responsavel, #CPF\_veterinario);

Vacinacao(#codigoVacina, #codigoAnimal, #codigoClinica, data, dose)

AjudaAnimal (#CPFvoluntario, #codigoaAnimal);

Adocao (#CPF\_adotante, #codigoAnimal, data, obs);

TelefonePessoa (#CPF, email);

EmailPessoa (#CPF, email);

EmailClinica (#CNPJ, email);

Gerencia (#CPF\_funcionario, #codigoRecurso);

VeterinarioTrabalha(#CPF\_veterinario, #CPNJ\_clinica, dataInicio, dataFim).

### 3.2 Domínio

Tabela Adoção

Atributo	Tipo de Dados	Restrição Domínio	Restrição Entidade	Restrição Referencial
CPF_adotante	VARCHAR(11)	Obrigatório e apenas dígitos numéricos	Chave Primária	Sim, ele vem da Tabela Adotante
codigoAnimal	INTEGER	Obrigatório e maior que zero	Chave Primária	Sim, ele vem da Tabela Animal
data	DATE	Obrigatório	-	-



obs	VARCHAR(60)	Não Obrigatório	-	-
-----	-------------	-----------------	---	---

Tabela Adotante

Atributo	Tipo de Dados	Restrição Domínio	Restrição Entidade	Restrição Referencial
CPF	VARCHAR(11)	Obrigatório e apenas dígitos numéricos	Chave Primária	Sim, da Tabela Pessoa

Tabela Ajuda animal

Atributo	Tipo de Dados	Restrição Domínio	Restrição Entidade	Restrição Referencial
CPFvoluntario	VARCHAR(11)	Obrigatório e apenas dígitos numéricos	Chave Primária	Sim, vem da Tabela Voluntário
codigoAnimal	INTEGER	Obrigatório e maior que zero	Chave Primária	Sim, virá da Tabela Animal

Tabela Alimento

Atributo	Tipo de Dados	Restrição Domínio	Restrição Entidade	Restrição Referencial
codigo	INTEGER	Obrigatório e maior que zero	Chave Primária	Sim, da tabela Recurso
tipo	VARCHAR(60)	Obrigatório	-	-
unidade	VARCHAR(60)	Obrigatório	-	-
qtdd	INTEGER	Obrigatório	-	-
dtValidade	DATE	Obrigatório	-	-

Tabela Animal

Atributo	Tipo de Dados	Restrição Domínio	Restrição	Restrição
----------	---------------	-------------------	-----------	-----------

			Entidade	Referencial
codigo	INTEGER	Obrigatório e maior que zero	Chave Primária	-
nome	VARCHAR(60)	Obrigatório	-	-
sexo	CHAR(1)	Obrigatório e tem que ser 'M' ou 'F'	-	-
especie	VARCHAR(60)	Obrigatório	-	-
raca	VARCHAR(60)	Obrigatório	-	-
data	DATE	Obrigatório	-	-
porte	VARCHAR(60)	Obrigatório	-	-
estadoSaude	VARCHAR(20)	Obrigatório e tem que ser ("Saudável" e "Em tratamento")	-	-
castrado	INTEGER	Obrigatório e tem que ser 1 ou 0	-	-

Tabela Clínica Veterinária

Atributo	Tipo de Dados	Restrição Domínio	Restrição Entidade	Restrição Referencial
CNPJ	VARCHAR(14)	Obrigatório	Chave Primária	-
nome	VARCHAR(60)	Obrigatório	Valor único	-
especialidade	VARCHAR(60)	Obrigatório	-	-
numeroLocal	INTEGER	Obrigatório	-	-
cep	CHAR(8)	Obrigatório	-	-
logradouro	CHAR(50)	Obrigatório	-	-

Tabela Consulta Animal

Atributo	Tipo de Dados	Restrição	Restrição	Restrição
----------	---------------	-----------	-----------	-----------

		Domínio	Entidade	Referencial
id	INTEGER	Obrigatório e maior que zero	Chave Primária	-
codigoAnimal	INTEGER	Obrigatório e maior que zero	Chave Primária	Sim, da tabela Animal
CNPJclinica	VARCHAR(14)	Obrigatório e apenas dígitos numéricos	Chave Primária	Sim, da tabela Clínica Veterinária
CPF_responsavel	VARCHAR(11)	Obrigatório e apenas dígitos numéricos	Chave Primária	Sim, vem da tabela Voluntário
cpf_veterinario	VARCHAR(11)	Obrigatório e apenas dígitos numéricos	Chave Primária	Sim, vem da tabela Veterinário
prescricao	VARCHAR(90)	Obrigatório	-	-
descricao	VARCHAR(90)	Obrigatório	-	-
data	DATE	Obrigatório	-	-

Tabela Dinheiro

Atributo	Tipo de Dados	Restrição Domínio	Restrição Entidade	Restrição Referencial
codigo	INTEGER	Obrigatório e maior que zero	Chave Primária	Sim, da tabela Recurso
cifra	VARCHAR(3)	Obrigatório	-	-
valor	DECIMAL(9,2)	Obrigatório e tem que ser maior que zero	-	-

Tabela EmailClinica

Atributo	Tipo de Dados	Restrição Domínio	Restrição Entidade	Restrição Referencial
CNPJ	VARCHAR(14)	Obrigatório e apenas dígitos numéricos	Chave Primária	Sim, da tabela Clínica Veterinária
email	VARCHAR(40)	Obrigatório	Chave Primária	-

Tabela EmailPessoa

Atributo	Tipo de Dados	Restrição Domínio	Restrição Entidade	Restrição Referencial
CPF	VARCHAR(11)	Obrigatório e apenas dígitos numéricos	Chave Primária	Sim, vem da tabela Pessoa
email	VARCHAR(40)	Obrigatório	Chave Primária	-

Tabela Funcionário

Atributo	Tipo de Dados	Restrição Domínio	Restrição Entidade	Restrição Referencial
CPF	VARCHAR(11)	Obrigatório e apenas dígitos numéricos	Chave Primária	Sim, da Tabela Pessoa
dataContratação	DATE	Obrigatório	-	-
remuneração	DECIMAL(9,2)	Obrigatório	-	-
tipoContrato	CHAR(30)	Obrigatório e tem que ser "CLT", "PJ",	-	-

		“estágio” ou “meio periodo”		
cargo	CHAR(60)	Obrigatório	-	-
horasSemana	INTEGER	Obrigatório e tem que ser maior que 0	-	-

Tabela Gerencia

Atributo	Tipo de Dados	Restrição Domínio	Restrição Entidade	Restrição Referencial
CPF_funcionario	VARCHAR(11)	Obrigatório e apenas dígitos numéricos	Chave Primária	Sim, vem da Tabela Funcionário
codigoRecurso	INTEGER	Obrigatório e maior que zero	Chave Primária	Sim, da tabela Recurso

Tabela Objeto

Atributo	Tipo de Dados	Restrição Domínio	Restrição Entidade	Restrição Referencial
codigo	INTEGER	Obrigatório e maior que zero	Chave Primária	Sim, da tabela Recurso
nome	VARCHAR(60)	Obrigatório	-	-
tipo	VARCHAR(60)	Obrigatório	-	-
obs	VARCHAR(90)	Não Obrigatório	-	-

Tabela Pessoa

Atributo	Tipo de Dados	Restrição Domínio	Restrição Entidade	Restrição Referencial
CPF	VARCHAR(11)	Obrigatório	Chave Primária	-
primeiroNome	VARCHAR(60)	Obrigatório	-	-

meioNome	VARCHAR(60)	Não Obrigatório	-	-
ultimoNome	VARCHAR(60)	Obrigatório	-	-
dataNasc	DATE	Obrigatório	-	-
idade	INTEGER	Obrigatório e entre 18 e 150	-	-
sexo	CHAR(1)	Obrigatório e tem que ser 'M' ou 'F'	-	-
numeroLocal	INTEGER	Obrigatório	-	-
cep	CHAR(8)	Obrigatório	-	-
logradouro	CHAR(50)	Obrigatório	-	-

Tabela Recurso

Atributo	Tipo de Dados	Restrição Domínio	Restrição Entidade	Restrição Referencial
codigo	INTEGER	Obrigatório e maior que zero	Chave Primária	-
descricao	VARCHAR(100)	Obrigatório	-	-
origem	VARCHAR(60)	Obrigatório e tem que ser ("Doação", "Recurso Próprio", "Eventos")	-	-
data	DATE	Obrigatório	-	-

Tabela TelefoneClinica

Atributo	Tipo de Dados	Restrição Domínio	Restrição Entidade	Restrição Referencial
CNPJ	VARCHAR(14)	Obrigatório e apenas dígitos numéricos	Chave Primária	Sim, da tabela Clínica Veterinária
telefone	VARCHAR(12)	Obrigatório	Chave Primária	-

Tabela TelefonePessoa

Atributo	Tipo de Dados	Restrição Domínio	Restrição Entidade	Restrição Referencial
CPF	VARCHAR(11)	Obrigatório e apenas dígitos numéricos	Chave Primária	Sim, vem da tabela Pessoa
telefone	VARCHAR(12)	Obrigatório	Chave Primária	-

Tabela Vacina

Atributo	Tipo de Dados	Restrição Domínio	Restrição Entidade	Restrição Referencial
codigo	INTEGER	Obrigatório e maior que zero	Chave Primária	-
nome	VARCHAR(60)	Obrigatório	-	-
tipo	VARCHAR(60)	Obrigatório	-	-
viaAplicacao	VARCHAR(60)	Obrigatório e tem que ser ("Oral" e "Injetavel (Subctanea)", "Injetavel (Intramuscular)" e Intranasal)	-	-
lote	VARCHAR(60)	Obrigatório	-	-
dataValidade	DATE	Obrigatório	-	-

Tabela Vacinacao

Atributo	Tipo de Dados	Restrição Domínio	Restrição Entidade	Restrição Referencial
codigoVacina	INTEGER	Obrigatório e maior que zero	Chave Primária	Sim, ele vem da Tabela Vacina
codigoAnimal	INTEGER	Obrigatório e	Chave Primária	Sim, ele vem da

		maior que zero		Tabela Animal
CNPJ	VARCHAR(14)	Obrigatório e apenas dígitos numéricos	Chave Primária	Sim, da tabela Clínica Veterinária
data	DATE	Obrigatório	-	-
dose	INTEGER	Obrigatório	-	-

Tabela Veterinario

Atributo	Tipo de Dados	Restrição Domínio	Restrição Entidade	Restrição Referencial
CPF	VARCHAR(11)	Obrigatório e apenas dígitos numéricos	Chave Primária	Sim, vem da tabela Pessoa
Especialidade	VARCHAR(60)	Obrigatório	-	-
CFMV	VARCHAR(13)	Obrigatório	-	-

Tabela VeterinarioTrabalha

Atributo	Tipo de Dados	Restrição Domínio	Restrição Entidade	Restrição Referencial
CPF_veterinario	VARCHAR(11)	Obrigatório e apenas dígitos numéricos	Chave Primária	Sim, vem da Tabela Veterinario
CNPJ_clinica	VARCHAR(14)	Obrigatório e apenas dígitos numéricos	Chave Primária	Sim, vem da Tabela Clinica
dataInicio	DATE	Obrigatório	-	-



dataFim	DATE	-	-	-
---------	------	---	---	---

Tabela Voluntário

Atributo	Tipo de Dados	Restrição Domínio	Restrição Entidade	Restrição Referencial
CPF	VARCHAR(11)	Obrigatório e apenas dígitos numéricos	Chave Primária	Sim, da Tabela Pessoa
CPFsupervisor	VARCHAR(11)	Obrigatório e apenas dígitos numéricos	-	Sim, referenciando da mesma Tabela Voluntário
especialidade	VARCHAR(60)	Obrigatório	-	-
numeroLocal	INTEGER	Obrigatório	-	-
cep	CHAR(8)	Obrigatório	-	-
logradouro	CHAR(50)	Obrigatório	-	-

#### 4. Scripts de Criação do Banco de Dados e Tabelas

##### 4.1 Criando DataBase

SQL

```
CREATE DATABASE PatasUrbanas;
```

##### 4.2 Criando Schema

SQL

```
CREATE SCHEMA EsquemaPatasUrbanas;
```

##### 4.3 Script Criação Tabelas

## Tabela Adocao

SQL

```
CREATE TABLE Adocao (  
    CPF_adotante VARCHAR(11) NOT NULL REFERENCES Adotante(CPF),  
    codigoAnimal INTEGER NOT NULL REFERENCES Animal(codigo),  
    data DATE NOT NULL,  
    obs VARCHAR(60),  
    PRIMARY KEY (CPF_adotante, codigoAnimal)  
);
```

## Tabela Adotante

SQL

```
CREATE TABLE Adotante (  
    CPF VARCHAR(11) PRIMARY KEY REFERENCES Pessoa(CPF)  
);
```

## Tabela AjudaAnimal

SQL

```
CREATE TABLE AjudaAnimal (  
    CPFvoluntario VARCHAR(11) NOT NULL REFERENCES Voluntario(CPF),  
    codigoAnimal INTEGER NOT NULL REFERENCES Animal(codigo),  
    PRIMARY KEY (CPFvoluntario, codigoAnimal)  
);
```

## Tabela Alimento

SQL

```
CREATE TABLE Alimento (  
    codigo INTEGER PRIMARY KEY REFERENCES Recurso(codigo),  
    tipo VARCHAR(60) NOT NULL,  
    unidade VARCHAR(60) NOT NULL,  
    qtdd INTEGER NOT NULL DEFAULT 0 CHECK (qtdd >= 0),  
    dtValidade DATE NOT NULL  
);
```

## Tabela Animal

SQL

```
CREATE TABLE Animal (  
    codigo INTEGER PRIMARY KEY CHECK (codigo > 0),  
    nome VARCHAR(60) NOT NULL,  
    sexo CHAR(1) NOT NULL CHECK (sexo IN ('M', 'F')),  
    especie VARCHAR(60) NOT NULL,  
    raca VARCHAR(60) NOT NULL,  
    data DATE NOT NULL,  
    porte VARCHAR(60) NOT NULL,  
    estadoSaude VARCHAR(20) NOT NULL CHECK (estadoSaude IN ('Saudável', 'Em  
tratamento')),  
    castrado INTEGER NOT NULL DEFAULT 0 CHECK (castrado IN (0, 1))  
);
```

## Tabela ClinicaVeterinaria

SQL

```
CREATE TABLE ClinicaVeterinaria (  
    CNPJ VARCHAR(14) PRIMARY KEY,  
    nome VARCHAR(60) NOT NULL UNIQUE,  
    especialidade VARCHAR(60) NOT NULL,  
    numeroLocal INTEGER NOT NULL,  
    cep CHAR(8) NOT NULL,  
    logradouro CHAR(50) NOT NULL  
);
```

## Tabela ConsultaAnimal

SQL

```
CREATE TABLE ConsultaAnimal (  
    id INTEGER PRIMARY KEY,  
    data DATE NOT NULL,  
    descricao VARCHAR(90) NOT NULL,
```

```

prescricao VARCHAR(90) NOT NULL,
codigoAnimal INTEGER NOT NULL REFERENCES Animal(codigo),
CNPJclinica VARCHAR(14) NOT NULL REFERENCES ClinicaVeterinaria(CNPJ),
CPF_responsavel VARCHAR(11) NOT NULL REFERENCES Voluntario(CPF),
CPF_veterinario VARCHAR(11) NOT NULL REFERENCES Veterinario(CPF)
);

```

### Tabela Dinheiro

SQL

```

CREATE TABLE Dinheiro (
    codigo INTEGER PRIMARY KEY REFERENCES Recurso(codigo),
    cifra VARCHAR(3) NOT NULL,
    valor DECIMAL(9,2) NOT NULL DEFAULT 0 CHECK (valor > 0)
);

```

### Tabela EmailClinica

SQL

```

CREATE TABLE EmailClinica (
    CNPJ VARCHAR(14) NOT NULL REFERENCES ClinicaVeterinaria(CNPJ),
    email VARCHAR(40) NOT NULL,
    PRIMARY KEY (CNPJ, email)
);

```

### Tabela EmailPessoa

SQL

```

CREATE TABLE EmailPessoa (
    CPF VARCHAR(11) NOT NULL REFERENCES Pessoa(CPF),
    email VARCHAR(40) NOT NULL,
    PRIMARY KEY (CPF, email)
);

```

## Tabela Funcionario

SQL

```
CREATE TABLE Funcionario (  
    CPF VARCHAR(11) PRIMARY KEY REFERENCES Pessoa(CPF),  
    dataContratacao DATE NOT NULL,  
    remuneracao DECIMAL(9,2) NOT NULL DEFAULT (1518.00),  
    tipoContrato CHAR(30) NOT NULL CHECK (tipoContrato IN ('CLT', 'PJ',  
'estagio', 'meio periodo')),  
    cargo CHAR(60) NOT NULL,  
    horasSemana INTEGER NOT NULL CHECK (horasSemana > 0)  
);
```

## Tabela Gerencia

SQL

```
CREATE TABLE Gerencia (  
    CPF_funcionario VARCHAR(11) NOT NULL REFERENCES Funcionario(CPF),  
    codigoRecurso INTEGER NOT NULL REFERENCES Recurso(codigo),  
    PRIMARY KEY (CPF_funcionario, codigoRecurso)  
);
```

## Tabela Objeto

SQL

```
CREATE TABLE Objeto (  
    codigo INTEGER PRIMARY KEY REFERENCES Recurso(codigo),  
    nome VARCHAR(60) NOT NULL,  
    tipo VARCHAR(60) NOT NULL,  
    obs VARCHAR(90)  
);
```

## Tabela Pessoa

SQL

```
CREATE TABLE Pessoa (  
    CPF VARCHAR(11) PRIMARY KEY,  
    primeiroNome VARCHAR(60) NOT NULL,  
    meioNome VARCHAR(60) DEFAULT '',  
    ultimoNome VARCHAR(60) NOT NULL,  
    dataNasc DATE NOT NULL,  
    idade INTEGER NOT NULL CHECK (idade >= 18 AND idade <= 150),  
    sexo CHAR(1) NOT NULL CHECK (sexo IN ('M', 'F')),  
    numeroLocal INTEGER NOT NULL,  
    cep CHAR(8) NOT NULL,  
    logradouro CHAR(50) NOT NULL  
);
```

## Tabela Recurso

SQL

```
CREATE TABLE Recurso (  
    codigo INTEGER PRIMARY KEY CHECK (codigo > 0),  
    descricao VARCHAR(100) NOT NULL,  
    origem VARCHAR(60) NOT NULL CHECK (origem IN ('Doação', 'Recurso  
Próprio', 'Eventos')),  
    data DATE NOT NULL  
);
```

## Tabela TelefonePessoa

SQL

```
CREATE TABLE TelefonePessoa (  
    CPF VARCHAR(11) NOT NULL REFERENCES Pessoa(CPF),  
    telefone VARCHAR(12) NOT NULL,  
    PRIMARY KEY (CPF, telefone)  
);
```

## Tabela TelefoneClinica

SQL

```
CREATE TABLE TelefoneClinica (  
    CNPJ VARCHAR(14) NOT NULL REFERENCES ClinicaVeterinaria(CNPJ),  
    telefone VARCHAR(12) NOT NULL,  
    PRIMARY KEY (CNPJ, telefone)  
);
```

## Tabela Vacina

SQL

```
CREATE TABLE Vacina (  
    codigo INTEGER PRIMARY KEY CHECK (codigo > 0),  
    nome VARCHAR(60) UNIQUE NOT NULL,  
    tipo VARCHAR(60) NOT NULL,  
    viaAplicacao VARCHAR(60) NOT NULL CHECK (viaAplicacao IN  
('Oral', 'Injetavel (Subcutanea)', 'Injetavel (Intramuscular)', 'Intranasal')),  
    lote VARCHAR(60) NOT NULL,  
    dataValidade DATE NOT NULL  
);
```

## Tabela Voluntario

SQL

```
CREATE TABLE Voluntario (  
    CPF VARCHAR(11) PRIMARY KEY REFERENCES Pessoa(CPF),  
    CPFsupervisor VARCHAR(11) REFERENCES Voluntario(CPF),  
    especialidade VARCHAR(60) NOT NULL,  
    numeroLocal INTEGER NOT NULL,  
    cep CHAR(8) NOT NULL,  
    logradouro CHAR(50) NOT NULL  
);
```

## Tabela Vacinacao

SQL

```
CREATE TABLE Vacinacao (  
    codigoVacina INTEGER NOT NULL REFERENCES Vacina(codigo),  
    codigoAnimal INTEGER NOT NULL REFERENCES Animal(codigo),  
    CNPJ_clinica VARCHAR(14) NOT NULL REFERENCES ClinicaVeterinaria(CNPJ),  
    data DATE NOT NULL,  
    dose INTEGER NOT NULL,  
    PRIMARY KEY (codigoVacina, codigoAnimal, data)  
);
```

## Tabela Veterinario

SQL

```
CREATE TABLE Veterinario (  
    CPF VARCHAR(11) PRIMARY KEY REFERENCES Pessoa(CPF),  
    especialidade VARCHAR(60) NOT NULL,  
    CRMV VARCHAR(20) NOT NULL  
);
```

## Tabela VeterinarioTrabalha

SQL

```
CREATE TABLE VeterinarioTrabalha (  
    CPF_veterinario VARCHAR(11) NOT NULL REFERENCES Veterinario(CPF),  
    CNPJ_clinica VARCHAR(14) NOT NULL REFERENCES ClinicaVeterinaria(CNPJ),  
    dataInicio DATE NOT NULL,  
    dataFim DATE,  
    PRIMARY KEY (CPF_veterinario, CNPJ_clinica)  
);
```



## 5. Scripts de Inserção, Atualização e Remoção do Banco de Dados

### 5.1 Inserção

#### Inserir um novo animal

SQL

```
INSERT INTO Animal (codigo, nome, sexo, especie, raca, data, porte,
estadoSaude, castrado)
VALUES (11, 'Pipoca', 'F', 'Cão', 'Lhasa Apso', '2021-12-05', 'Pequeno',
'Saudável', 1);
```

#### Inserir um novo voluntário que cuidará do animal 10

SQL

```
INSERT INTO Pessoa (CPF, primeiroNome, meioNome, ultimoNome, dataNasc, idade,
sexo, numeroLocal, cep, logradouro)
VALUES('12131314151', 'Marina', 'Oliveira', 'Costa', '2000-05-12', 25, 'F',
11, '13579246', 'Rua do Bem');

INSERT INTO Voluntario(dataIngresso, areaAtuacao, CPF, CPFsupervisor)
VALUES ('2025-01-01', 'Cuidadora', '12131314151', '11111111111');

INSERT INTO AjudaAnimal (CPFvoluntario, codigoAnimal)
VALUES ('12131314151', 1);
```

#### Inserir uma nova consulta

SQL

```
INSERT INTO consultaanimal (id, data, descricao, prescricao, codigoAnimal,
CNPJclinica, CPF_responsavel, CPF_veterinario)
VALUES(10, '2025-06-01', 'Infecção de ouvido', 'Antibiótico por 5 dias', 1,
'12345678000101', '11111111111', '22222222222');
```

### 5.2 Atualização

### Atualizar o nome de um animal

```
SQL
UPDATE Animal
SET nome = 'Maxwell'
WHERE codigo = 10;
```

### Atualizar o estado de saúde de um animal

```
SQL
UPDATE Animal
SET estadoSaude = 'Saudável'
WHERE codigo = 2;
```

### Atualizar a área de atuação de um voluntário

```
SQL
UPDATE Voluntario
SET areaatuacao = 'Veterinário(a) Sênior'
WHERE CPF = '2222222222';
```

### Atualizar o endereço completo de uma clínica veterinária específica

```
SQL
UPDATE ClinicaVeterinaria
SET logradouro = 'Rua Nova Esperança', numeroLocal = 321, cep = '55556666'
WHERE CNPJ = '23456789000102';
```

## 5.3 Remoção

### Remover os telefones do Gabriel Bressane

```
SQL
DELETE FROM TelefonePessoa
WHERE CPF IN (
    SELECT CPF
```

```

FROM Pessoa
WHERE primeiroNome = 'Gabriel' AND ultimoNome = 'Bressane'
)

```

## 6. Consultas

### 6.1 Exibir os nomes e CPFs dos voluntários:

Algebra relacional:

Unset

```

 $\pi$  primeironome, ultimnome, cpf (pessoa  $\bowtie$  pessoa.cpf = voluntario.cpf
(voluntario))

```

### Consulta SQL

SQL

```

SELECT
    P.primeironome,
    P.ultimonome,
    P.cpf
FROM pessoa AS P
    JOIN voluntario AS V ON P.cpf = V.cpf;

```

### 6.2 Listar todos os animais adotados, com nome do animal, nome do adotante e data de adoção:

Algebra relacional:

Unset

```

 $\pi$  primeironome, ultimnome, AN.nome, A.data (pessoa  $\bowtie$  P.cpf = AD.cpf
(adotante  $\bowtie$  AD.cpf = A.cpf_adotante (adocao  $\bowtie$  A.codigoanimal = AN.codigo
(animal))))

```

### Consulta SQL

SQL

```
SELECT
    P.primeironome,
    P.ultimonome,
    AN.nome AS nomeAnimal,
    A.data
FROM adocao AS A
    JOIN adotante AS AD ON AD.cpf = A.cpf_adotante
    JOIN pessoa AS P ON P.cpf = AD.cpf
    JOIN animal AS AN ON AN.codigo = A.codigoanimal;
```

6.3 Mostrar o nome dos funcionários e o total de recursos que cada um gerencia:

Algebra relacional:

Unset

```
 $\pi$  primeironome, ultimonome, R.codigo (pessoa \bowtie P.cpf = F.cpf (funcionario \bowtie F.cpf = G.cpf_funcionario (gerencia \bowtie G.codigorecurso = R.codigo (recurso))))
```

Consulta SQL

SQL

```
SELECT
    P.primeironome,
    P.ultimonome,
    COUNT(*) AS TotalRec
FROM funcionario AS F
    JOIN pessoa AS P ON P.cpf = F.cpf
    JOIN gerencia AS G ON F.cpf = G.cpf_funcionario
    JOIN recurso AS R ON G.codigorecurso = R.codigo
GROUP BY P.primeironome, P.ultimonome;
```

6.4 Listar todos os animais que já receberam vacinação, indicando a vacina e a data:

Algebra relacional:

Unset

```
 $\Pi$  A.nome, vacina.nome, V.data (animal  $\bowtie$  A.codigo = V.codigoanimal  
(vacinacao  $\bowtie$  V.codigovacina = vacina.codigo (vacina)))
```

## Consulta SQL

SQL

```
SELECT  
    A.nome AS nomeAnimal,  
    vacina.nome AS nomeVacina,  
    V.data  
FROM vacinacao AS V  
    JOIN animal AS A ON V.codigoanimal = A.codigo  
    JOIN vacina ON V.codigovacina = vacina.codigo;
```

## 6.5 Quantidade de animais consultados por cada veterinário

Algebra relacional:

Unset

```
 $\Pi$  primeironome, ultimonomo, C.cpf_veterinario (pessoa  $\bowtie$  P.cpf = V.cpf  
(veterinario  $\bowtie$  V.cpf = C.cpf_veterinario (consultaanimal)))
```

## Consulta SQL

SQL

```
SELECT  
    P.primeironome,  
    P.ultimonome,  
    COUNT(C.cpf_veterinario) AS qtdAnimaisConsultados  
FROM consultaanimal AS C  
    JOIN veterinario AS V ON V.cpf = C.cpf_veterinario  
    JOIN pessoa AS P ON V.cpf = P.cpf  
GROUP BY P.primeironome, P.ultimonome  
ORDER BY COUNT(C.cpf_veterinario) DESC;
```

## 6.6 Total de dinheiro arrecadado no ano de 2025:

Algebra relacional:

Unset

```
σ R.data >= '2025-01-01' ∧ R.data <= '2025-12-31' (dinheiro ⋈ D.codigo = R.codigo (recurso))
```

### Consulta SQL

SQL

```
SELECT
    SUM(D.valor)
FROM dinheiro AS D JOIN recurso AS R ON D.codigo = R.codigo
WHERE R.data >= '2025-01-01' AND R.data <= '2025-12-31';
```

6.7 Mostrar o nome da clínica e quantidade distinta de animais já atendidos nela (consultas realizadas):

Algebra relacional:

Unset

```
π CL.nome, C0.codigoanimal (clinicaveterinaria ⋈ CL.cnpj = C0.cnpjclinica (consultaanimal))
```

### Consulta SQL

SQL

```
SELECT
    CL.nome,
    COUNT(DISTINCT C0.codigoanimal)
FROM consultaanimal AS C0
    JOIN clinicaveterinaria AS CL ON C0.cnpjclinica = CL.cnpj
GROUP BY CL.nome;
```

6.8 Animais que nunca foram adotados:

Algebra relacional:

Unset

```
 $\pi_{\text{codigo}(\text{animal})} - \pi_{\text{AD.codigoanimal}} (\text{animal} \bowtie \text{A.codigo} = \text{AD.codigoanimal}(\text{adocao}))$ 
```

### Consulta SQL

SQL

```
SELECT codigo
FROM animal
EXCEPT
SELECT AD.codigoanimal
FROM ANIMAL AS A
      JOIN ADOCAO AS AD ON A.codigo = AD.codigoanimal;
```

## 6.9 Quantidade de supervisionados por supervisor

Algebra relacional:

Unset

```
 $\pi_{P.primeiroNome, P.ultimoNome, V.cpf} (\text{Pessoa} \bowtie P.cpf = V.cpfSupervisor(\text{Voluntario}))$ 
```

### Consulta SQL

SQL

```
SELECT
    P.primeiroNome,
    P.ultimoNome,
    COUNT(V.cpf) AS qtd_supervisionados
FROM Voluntario AS V
      JOIN Pessoa AS P ON P.cpf = V.cpfSupervisor
GROUP BY V.cpfSupervisor, P.primeiroNome, P.ultimoNome;
```

6.10 Total de funcionários distintos, quantos estão trabalhando e quantos estão parados:

Algebra relacional:

Unset

--Todos que estão trabalhando

$\Pi$  F.cpf (Funcionario  $\bowtie$  F.cpf = G.cpf\_funcionario (Gerencia))

-- Funcionários que não estão trabalhando

$\Pi$ (F.cpf) (Funcionario) -  $\Pi$ (G.cpf\_funcionario) (Gerencia)

## Consulta SQL

SQL

SELECT

COUNT(DISTINCT F.cpf) AS total\_funcionarios,  
COUNT(DISTINCT G.cpf\_funcionario) AS estao\_trabalhando,  
COUNT(DISTINCT F.cpf) - COUNT(DISTINCT G.cpf\_funcionario) AS

estao\_parados

FROM Funcionario AS F

LEFT JOIN Gerencia AS G ON F.cpf = G.cpf\_funcionario;