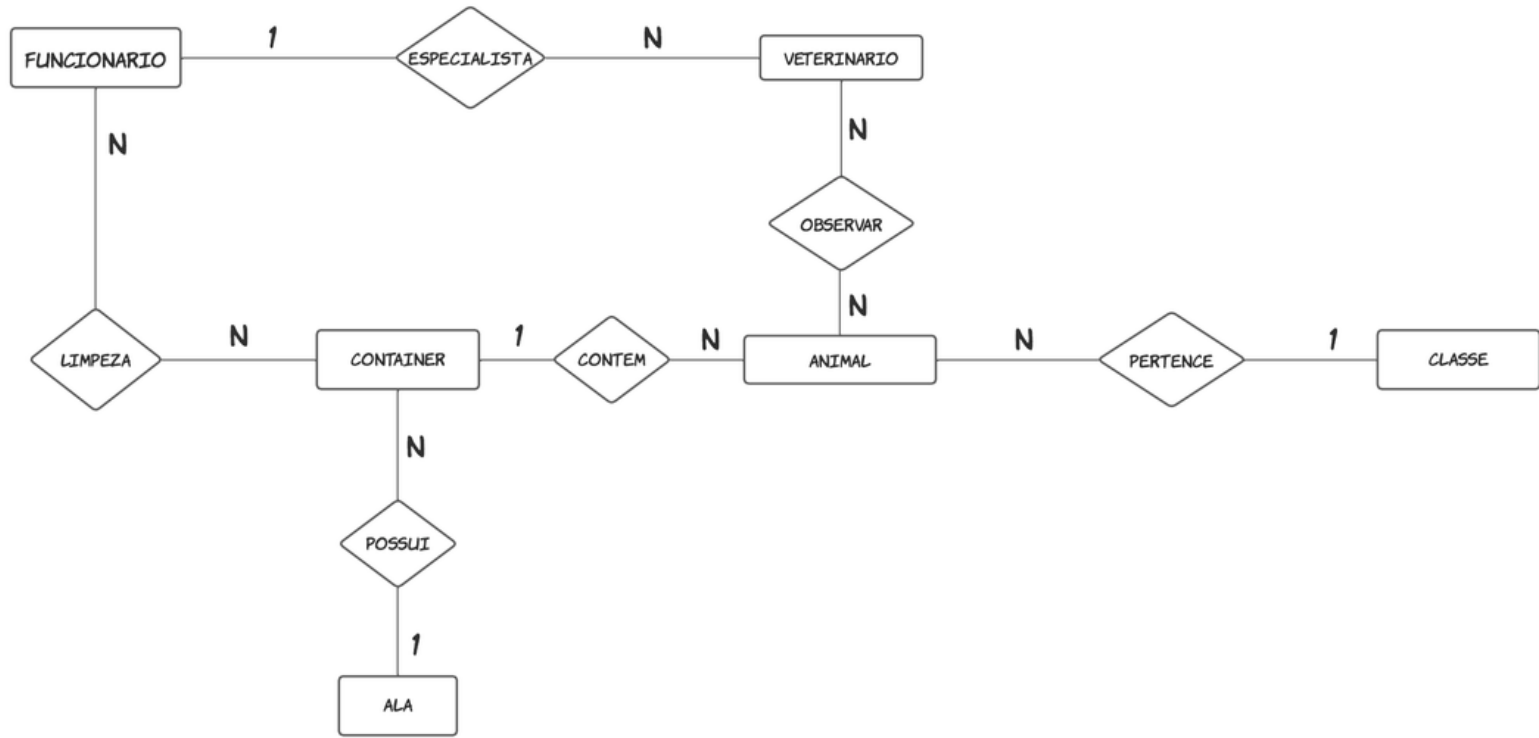


# DER



# MER

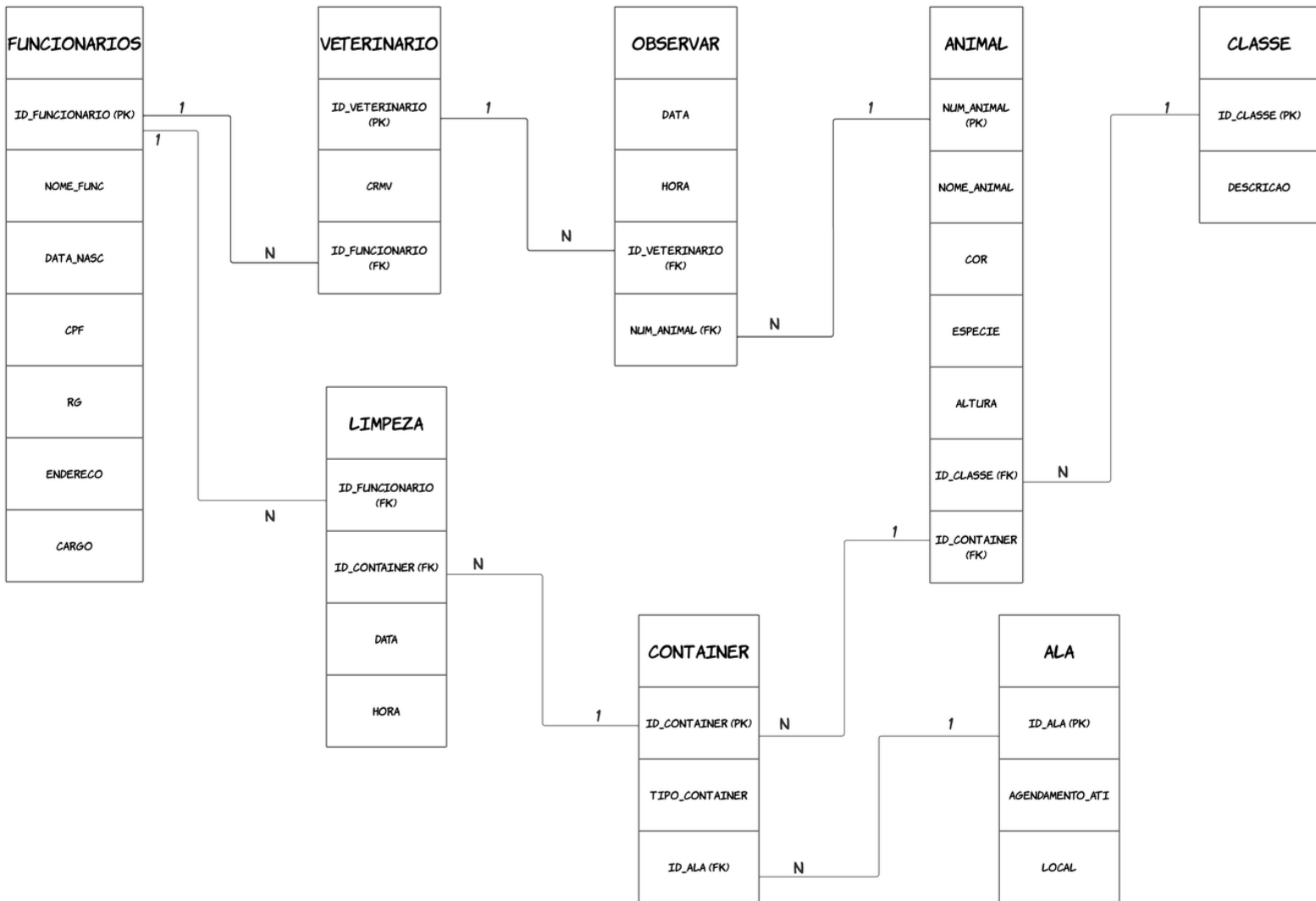


Tabela	Funcionarios			
Descrição	armazenar as informações dos funcionarios			
Observações	Está Tabela Tem Associação Com As tabelas veterinario e limpeza			
Campos				
Nome	Descrição	Tipo De Dado	Tamanho	Reações De Domínio
id_funcionario	Identificador único para cada funcionário.	INT		Pk/Identity
nome_func	Nome completo do funcionário.	VARCHAR	60	NOT NULL
data_nasc	Data de nascimento do funcionário.	DATE		NOT NULL
CPF	Número do Cadastro de Pessoa Física do funcionário.	VARCHAR	15	NOT NULL
rg	Número do Registro Geral do funcionário.	VARCHAR	20	NOT NULL
endereco	Endereço completo do funcionário.	VARCHAR	60	NOT NULL
cargo	Cargo ou função desempenhada pelo funcionário.	VARCHAR	35	NOT NULL

Tabela	VETERINARIO			
Descrição	armazenar as informações dos veterinario			
Observações	Está Tabela Tem Associação Com As tabelas funcionario e observar			
Campos				
Nome	Descrição	Tipo De Dado	Tamanho	Reações De Domínio
id_veterinario	Identificador único para cada veterinário.	INT		Pk/Identity
CRMV	Número do Conselho Regional de Medicina Veterinária atribuído ao veterinário.	VARCHAR	60	NOT NULL
ID_funcionario	Chave estrangeira que referencia o ID_funcionario da tabela Funcionarios.	DATE		FK/NOT NULL

Tabela	OBSERVAR			
Descrição	armazenar as informações das observações dos animais			
Observações	Está Tabela Tem Associação Com As tabelas veterinario e animal			
Campos				
Nome	Descrição	Tipo De Dado	Tamanho	Reações De Domínio
data	Data agendada para a consulta.	DATE		NOT NULL
hora	Horário agendado para a consulta.	VARCHAR	5	NOT NULL
id_veterinario	Chave estrangeira que referencia o ID_veterinario da tabela Veterinario.	INT		FK/NOT NULL
num_animal	Chave estrangeira que referencia o Num_animal da tabela Animal.	INT		FK/NOT NULL

Tabela	ANIMAL			
Descrição	armazenar as informações dos animais			
Observações	Está Tabela Tem Associação Com As tabelas container e classe			
Campos				
Nome	Descrição	Tipo De Dado	Tamanho	Reações De Domínio
Num_animal	Identificador único para cada animal.	INT		PK/NOT NULL
Nome_animal	Nome do animal.	VARCHAR	60	NOT NULL
Cor	Cor do animal.	VARCHAR	45	NOT NULL
Especie	Espécie do animal.	VARCHAR	45	NOT NULL
Altura	Altura do animal.	VARCHAR	12	NOT NULL
id_classe	Chave estrangeira que referencia o ID_classe da tabela Classe.	INT		FK/NOT NULL
id_container	Chave estrangeira que referencia o ID_container da tabela Container.	INT		FK/NOT NULL

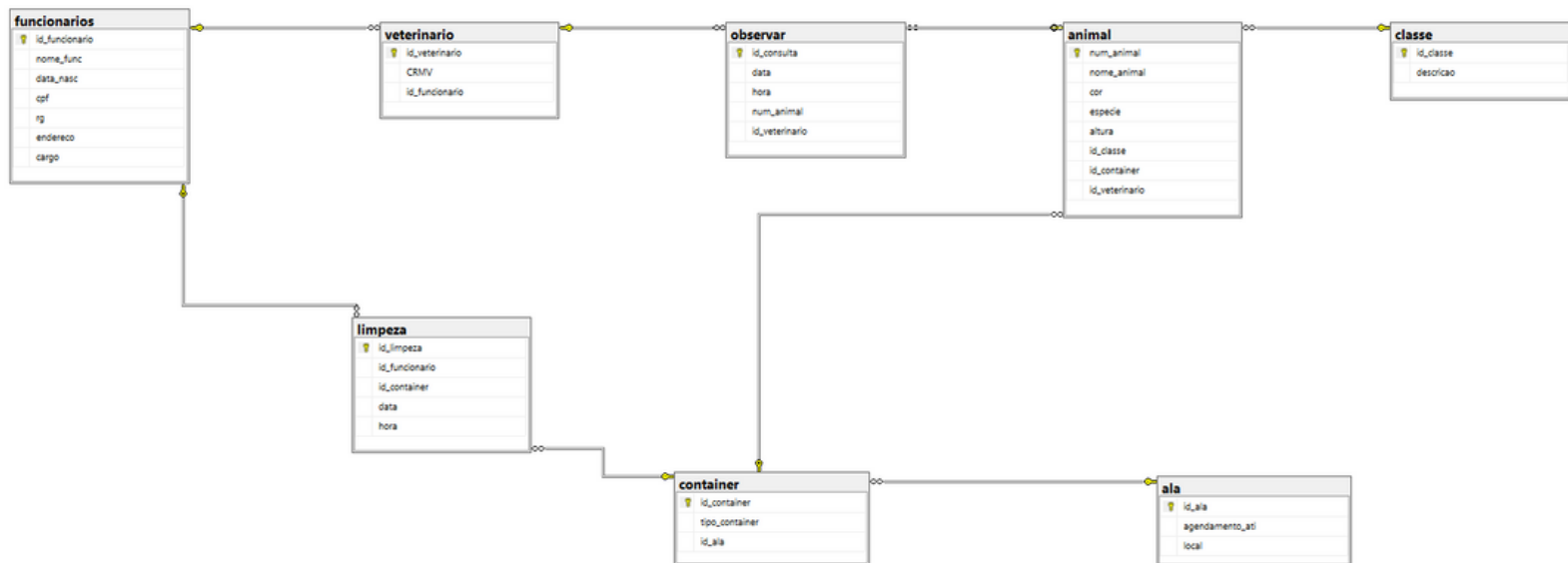
Tabela	CLASSE			
Descrição	armazenar as informações das classe dos animais			
Observações	Está Tabela Tem Associação Com A tabela animal			
Campos				
Nome	Descrição	Tipo De Dado	Tamanho	Reações De Domínio
id_classe	Identificador único para cada classe.	INT		PK/NOT NULL
Descricao	Descrição técnica das características específicas da classe.	VARCHAR	80	NOT NULL

Tabela	LIMPEZA			
Descrição	armazenar as informações das limpezas dos containers			
Observações	Está Tabela Tem Associação Com As tabelas funcionarios e container			
Campos				
Nome	Descrição	Tipo De Dado	Tamanho	Reações De Domínio
id_funcionario	Chave estrangeira que referencia o ID_funcionario da tabela Funcionario.	INT		FK/NOT NULL
id_container	Chave estrangeira que referencia o ID_container da tabela Container.	INT	80	FK/NOT NULL
DATA	Data agendada para a limpeza do container.	DATA		NOT NULL
HORA	Horário agendado para a limpeza do container.	VARCHAR	5	NOT NULL

Tabela	CONTAINER			
Descrição	armazenar as informações dos containers			
Observações	Está Tabela Tem Associação Com As tabelas limpeza,ala e animal			
Campos				
Nome	Descrição	Tipo De Dado	Tamanho	Reações De Domínio
id_container	Identificador único para cada container.	INT		PK/NOT NULL
tipo_container	Tipo de container utilizado para abrigar animais.	VARCHAR	80	NOT NULL
id_ala	Chave estrangeira que referencia o ID_ALA da tabela Ala.	INT		FK/NOT NULL

Tabela	ALA			
Descrição	armazenar as informações das alas			
Observações	Está Tabela Tem Associação Com As tabelas ala e container			
Campos				
Nome	Descrição	Tipo De Dado	Tamanho	Reações De Domínio
id_ala	Identificador único para cada ala.	INT		PK/NOT NULL
agendamento_ati	Tipo de atividade agendada para a ala (pode representar diferentes eventos ou finalidades da ala).	VARCHAR	80	NOT NULL
local	Local específico onde a ala está situada.	VARCHAR	80	NOT NULL

# DER - SMSS



# CÓDIGO - SMSS

```
CREATE DATABASE zoologico;  
USE zoologico;
```

```
CREATE TABLE funcionarios (  
  id_funcionario INT PRIMARY KEY,  
  nome_func VARCHAR(60),  
  data_nasc DATE,  
  cpf VARCHAR(15),  
  rg VARCHAR(20),  
  endereco VARCHAR(60),  
  cargo VARCHAR(35)  
);
```

```
CREATE TABLE veterinario (  
  id_veterinario INT PRIMARY KEY,  
  CRMV VARCHAR(20),  
  id_funcionario INT,  
  FOREIGN KEY (id_funcionario) REFERENCES funcionarios(id_funcionario)  
);
```

```
CREATE TABLE observar (  
  id_consulta INT PRIMARY KEY,  
  data DATE,  
  hora TIME,  
  num_animal INT,  
  id_veterinario INT,  
  FOREIGN KEY (num_animal) REFERENCES animal(num_animal),  
  FOREIGN KEY (id_veterinario) REFERENCES veterinario(id_veterinario)  
);
```

```
CREATE TABLE animal (  
  num_animal INT PRIMARY KEY,  
  nome_animal VARCHAR(60),  
  cor VARCHAR(45),  
  especie VARCHAR(45),  
  altura VARCHAR(12),  
  id_classe INT,  
  id_container INT,  
  id_veterinario INT,  
  FOREIGN KEY (id_classe) REFERENCES classe(id_classe),  
  FOREIGN KEY (id_container) REFERENCES container(id_container),  
  FOREIGN KEY (id_veterinario) REFERENCES veterinario(id_veterinario)  
);
```

```
CREATE TABLE classe (  
  id_classe INT PRIMARY KEY,  
  descricao VARCHAR(80)  
);
```

```
CREATE TABLE limpeza (  
  id_limpeza INT PRIMARY KEY,  
  id_funcionario INT,  
  id_container INT,  
  data DATE,  
  hora TIME,  
  FOREIGN KEY (id_funcionario) REFERENCES funcionarios(id_funcionario),  
  FOREIGN KEY (id_container) REFERENCES container(id_container)  
);
```

```
CREATE TABLE container (  
  id_container INT PRIMARY KEY,  
  tipo_container VARCHAR(80),  
  id_ala INT,  
  FOREIGN KEY (id_ala) REFERENCES ala(id_ala)  
);
```

```
CREATE TABLE ala (  
  id_ala INT PRIMARY KEY,  
  agendamento_ati VARCHAR(80),  
  local VARCHAR(80)  
);
```