

# 1 Construção da Base de Dados

O processo de desenvolvimento de uma base de dados deve seguir uma abordagem estruturada e coerente, de forma a garantir integridade, eficiência e alinhamento com os objetivos do sistema. De acordo com Thomas Connolly e Carolyn Begg, no livro *Database Systems: A Practical Approach to Design, Implementation, and Management*, o desenvolvimento de uma base de dados compreende diversas fases, desde a recolha de requisitos até à modelação conceptual, lógica e física.

No âmbito do projeto SeePaw, o desenvolvimento da base de dados seguiu o Modelo de Desenvolvimento em Dois Níveis, composto pelas seguintes fases principais:

- Modelo Conceptual de Dados (MCD): representação abstrata das entidades do domínio, dos seus atributos e relacionamentos, sem dependência de um sistema de gestão de bases de dados específico;
- Modelo Lógico de Dados (MLD): tradução do modelo conceptual para um modelo relacional, compatível com um Sistema de Gestão de Bases de Dados Relacional (SGBDR).

Neste projeto, não foram aplicadas todas as etapas formais descritas na metodologia de Connolly & Begg, mas apenas as necessárias para garantir a correta estruturação da base de dados e a sua implementação prática. Assim, no modelo conceptual, foi realizada a verificação das entidades e ligações entre elas, assegurando que os relacionamentos respeitassem as regras de negócio e as cardinalidades definidas. Em alguns casos, procedeu-se à divisão de entidades com base nas cardinalidades e na necessidade de representar interações específicas, aplicando as regras de conversão mais adequadas.

Com base neste modelo conceptual simplificado, foi derivado o modelo lógico, onde cada entidade e relacionamento relevante foi traduzido para relações (tabelas), com a definição de chaves primárias, estrangeiras e restrições de integridade.

O objetivo principal deste trabalho foi conceber e implementar uma base de dados relacional funcional que suporte as principais operações da plataforma SeePaw, dedicada à gestão de animais para adoção e apadrinhamento.

A base de dados desenvolvida permite armazenar e relacionar informações sobre animais, utilizadores e centros de acolhimento animal (CAA), possibilitando operações como o registo de dados, gestão de perfis, apadrinhamento e atribuição de ownership, em conformidade com as regras de negócio definidas para o sistema.

## 1.1 Modelo Conceptual de Dados

### 1.1.1 Identificar Entidades

A identificação das entidades constitui a primeira etapa do processo de conceção do modelo conceptual da base de dados da plataforma SeePaw. Este passo tem como objetivo determinar e definir os principais objetos de interesse que o sistema deverá representar, servindo de base para a estruturação da informação e para a definição das relações entre os diferentes elementos do domínio.

No contexto do SeePaw, a análise dos requisitos funcionais e não funcionais permitiu extrair os substantivos e conceitos centrais mencionados nas especificações dos utilizadores e das equipas de desenvolvimento. Assim, foram identificadas entidades que representam objetos com existência própria e relevância no sistema, tais como Animal, Utilizador e CAA.

Entidade	Descrição	Ocorrência
Animal	Termo geral que representa qualquer animal que possa ser apadrinhado ou adotado especialmente. Pode-se referir a qualquer tipo de animal.	Um animal fica disponibilizado na plataforma quando o administrador de um CAA o adiciona.
Utilizador	Pessoa que se regista na plataforma e visualizar os animais e apadrinhá-los ou tornar-se Owner.	Um utilizador precisa se registar na plataforma para fazer qualquer funcionalidade.
CAA	Centro de Acolhimento Animal que deve ser registado por um utilizador com role que o permita.	Ao registar o CAA na plataforma é possível adicionar animais à plataforma que estejam no CAA e gerir pedidos de Ownership.

### 1.1.2 Identificar o tipo de relacionamentos entre entidades

Após a identificação das entidades, o passo seguinte na conceção do modelo conceptual da base de dados do SeePaw consiste em determinar os relacionamentos existentes entre essas entidades. Este processo tem como objetivo representar de forma clara as ligações e interações que ocorrem no domínio da aplicação, permitindo compreender como os diferentes elementos do sistema se associam entre si.

A análise dos requisitos funcionais e narrativas dos utilizadores possibilita identificar essas ligações através de verbos ou expressões verbais, que indicam ações ou dependências entre entidades. No caso do SeePaw, são as seguintes:

Entidade 1	Multiplicidade	Relacionamento	Multiplicidade	Entidade 2
Utilizador	0..*	apadrinha	0..*	Animal
Utilizador	0..*	realiza atividades com	0..*	Animal
Utilizador	0..*	adiciona aos favoritos	0..*	Animal
Utilizador	0..1	é owner	0..*	Animal
CAA	1..1	abriga	0..*	Animal
CAA	1..1	é administrado por	1..*	Utilizador

Entidade	Atributo	Descrição	Tipo de dados & Tamanho	Nulos	Multivalor	Derivado	Valor por defeito
Animal	AnimalId	Identificador único do animal	10 caracteres variáveis	Não	Não	Não	-
	Nome	Nome do animal	255 caracteres variáveis	Não	Não	Não	-
	Estado	Estado do animal(Disponível, Apadrinhado.	EstadoAnimal Type	Não	Não	Não	-

		Totalmente Apadrinhado, Owner, Inativo)					
	Descricao	Descrição do animal	255 caracteres variáveis	Sim	Não	Não	-
	Tipo	Espécie de animal	EspecieType	Não	Não	Não	-
	Estatura	Porte do animal	PorteType	Não	Não	Não	-
	Sexo	Sexo do animal (fêmea ou macho)	SexoType	Não	Não	Não	-
	Cor	Cor do animal	255 caracteres variáveis	Não	Não	Não	-
	DataNascimento	Data Nascimento do animal	Data	Não	Não	Não	-
	Esterilizado	Animal com esterilização	Valor lógico	Não	Não	Não	-
	Raca	Raça do animal	RacasType	Não	Não	Não	-
	Custo	Custo para manter o animal	Número Inteiro	Não	Não	Não	-
	Caracteristicas	Traços do animal	255 caracteres variáveis	Não	Não	Não	-
	DataCriacao	Data da criação do animal	Data e hora	Não	Não	Não	-
	DataAtualizacao	Data de Atualização do animal	Data e hora	Não	Não	Não	-
	ImagemPrincipalUrl	Identificador único da imagem	10 caracteres variáveis	Não	Não	Não	-
	ImagemExtraUrl	Identificador único da imagem	10 caracteres variáveis	Não	Sim	Não	-

Entidade	Atributo	Descrição	Tipo de dados & Tamanho	Nulls	Multi-valor	Derivado	Valor por defeito
Utilizador	UtilizadorId	Identificador único do utilizador	10 Caracteres variáveis	Não	Não	Não	-

	Nome	Nome do utilizador	255 Caracteres variáveis	Não	Não	Não	-
	DataNascimento	Data de Nascimento do utilizador	Data	Não	Não	Não	-
	Rua	Rua do utilizador	255 caracteres variáveis	Não	Não	Não	-
	Cidade	Cidade do utilizador	255 caracteres variáveis	Não	Não	Não	-
	CodigoPostal	Código Postal do utilizador	255 caracteres variáveis	Não	Não	Não	-
	Telefone	Número de telemóvel do utilizador	Número Inteiro com 9 dígitos	Não	Não	Não	-
	Email	Email do utilizador	255 caracteres variáveis	Não	Não	Não	-
	Password	Password encriptada	255 caracteres variáveis	Não	Não	Não	-
	DataCriacao	Data em que utilizador se registou	Data e Hora	Não	Não	Não	-
	DataAtualizacao	Data em que utilizador de atualização do utilizador	Data e Hora	Não	Não	Não	-

Entidade	Atributo	Descrição	Tipo de dados & Tamanho	Nulls	Multi-valor	Derivado	Valor por defeito
----------	----------	-----------	-------------------------	-------	-------------	----------	-------------------

CAA	CAAIId	Identificador único do CAA	10 caracteres variáveis	Não	Não	Não	-
	Nome	Nome do CAA	255 caracteres variáveis	Não	Não	Não	-
	Rua	Rua do CAA	255 caracteres variáveis	Não	Não	Não	-
	Cidade	Cidade do CAA	255 caracteres variáveis	Não	Não	Não	-
	CodigoPostal	Código Postal do CAA	255 caracteres variáveis	Não	Não	Não	-
	Telefone	Número de telemóvel do CAA	Número Inteiro com 9 dígitos	Não	Não	Não	-
	NIF	Número de Identificação Fiscal do CAA	Número Inteiro com 9 dígitos	Não	Não	Não	-
	HoraAbertura	Horas a que abre o CAA	Hora	Não	Não	Não	-
	HoraFecho	Horas a que fecha o CAA	Hora	Não	Não	Não	-
	Imagem PrincipalUrl	Identificador único da imagem	10 caracteres variáveis	Não	Não	Não	-
	DataCriacao	Data da criação do animal	Data e hora	Não	Não	Não	-
	DataAtualizacao	Data de Atualização do animal	Data e hora	Não	Não	Não	-

#### 1.1.5 Determinar domínios dos atributos

Depois de identificar e associar os atributos às respetivas entidades e relacionamentos, o passo seguinte no desenvolvimento do modelo conceptual da base de dados do SeePaw consiste em definir os domínios de cada atributo. Um domínio corresponde

ao conjunto de valores válidos que um determinado atributo pode assumir, estabelecendo limites e regras que garantem a consistência e integridade dos dados armazenados.

A definição dos domínios permite especificar, para cada atributo, o tipo de dados, o tamanho máximo permitido e, quando aplicável, o conjunto restrito de valores possíveis. Por exemplo, o atributo sexo da entidade Animal pode ter como domínio apenas os valores “Macho” e “Fêmea”

Esta etapa é essencial para assegurar que os dados inseridos no sistema respeitam as regras de negócio e os requisitos funcionais definidos, evitando inconsistências e garantindo que a base de dados mantém a qualidade e fiabilidade esperadas.

Atributo	Tipos de dados & Tamanho	Restrições
EstadoAnimalType	15 caracteres variáveis	Deve assumir valores pré definidos: “Disponível”, “Apadrinhado”, “Totalmente Apadrinhado”, “Ownership”, “Inativo”
EspecieType	50 caracteres variáveis	Deve assumir valores fixos de espécies de animais.
PorteType	15 caracteres variáveis	Deve assumir valores pré definidos: “Pequeno”, “Medio”, “Grande”
SexoType	15 caracteres variáveis	Deve assumir valores pré definidos: “Macho”, “Fêmea”
RacasType	15 caracteres variáveis	Deve assumir valores fixos de raças de animais.

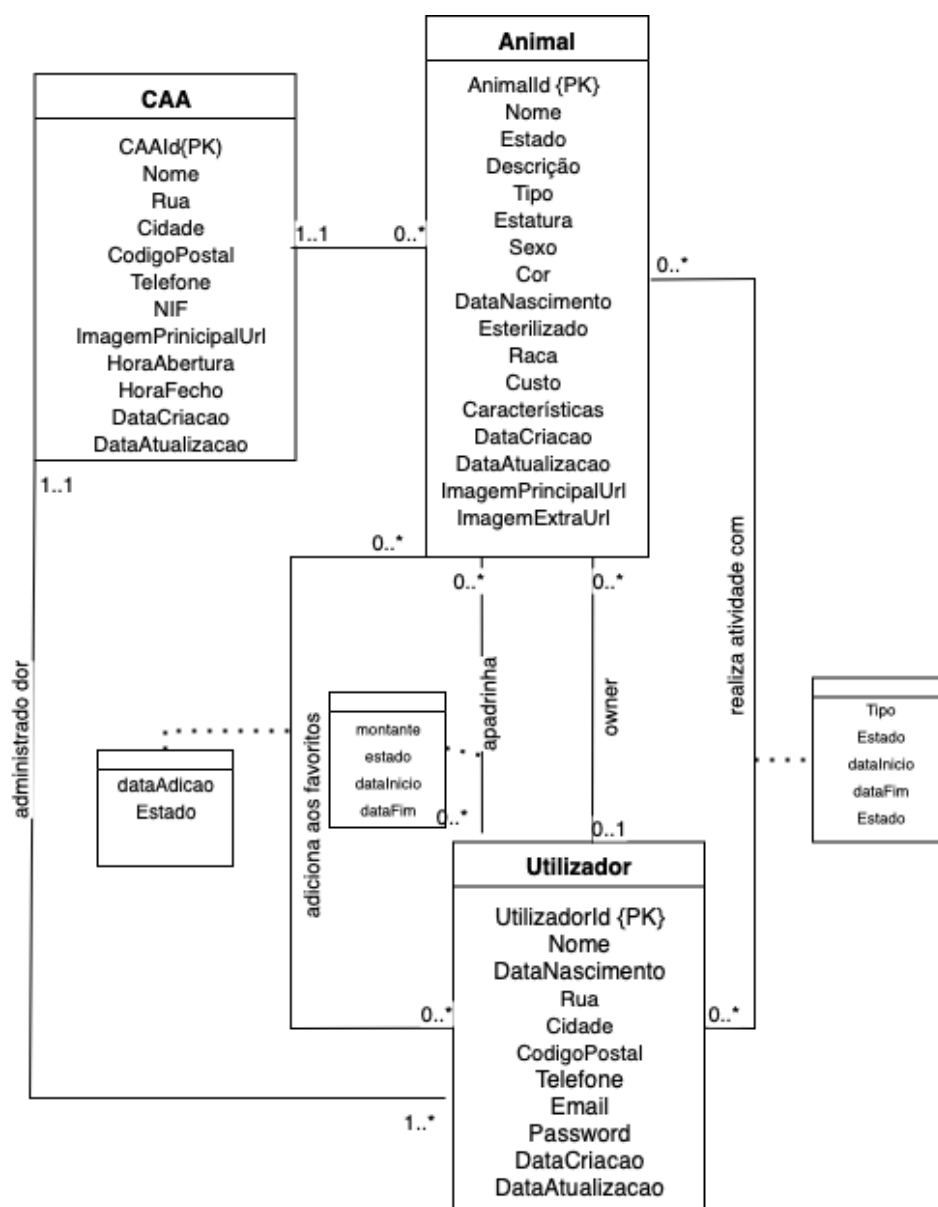
#### 1.1.6 Determinar chaves candidatas, primárias e alternativas

Nesta etapa do processo de conceção do modelo conceptual da base de dados do SeePaw, procede-se à identificação das chaves candidatas de cada entidade, ou seja, dos conjuntos mínimos de atributos capazes de identificar univocamente cada ocorrência dessa entidade. A partir das chaves candidatas identificadas, é seleccionada uma chave primária, que servirá como identificador principal e exclusivo de cada registo, sendo as restantes consideradas chaves alternativas.

A escolha adequada da chave primária é essencial para garantir a integridade e unicidade dos dados, evitando duplicações e assegurando um acesso eficiente às informações. No caso da base de dados do SeePaw, por exemplo, as entidades Animal, Abrigo e Utilizador possuem respetivamente atributos como AnimalId, CAALId e UtilizadorId, definidos como chaves primárias. Estes identificadores únicos permitem distinguir inequivocamente cada registo, independentemente de existirem nomes ou outros atributos repetidos.

Entidade	Chave Primária	Chave Alternativa
Animal	AnimalId	-
CAA	CAAIId	-
Utilizador	UtilizadorId	-

### 1.1.7 Diagrama E/R com identificação das chaves primárias



## 2 Modelo Lógico de Dados

Após a construção do modelo conceptual da base de dados do SeePaw, o passo seguinte consiste em traduzir esse modelo conceptual para um modelo lógico de dados, representando de forma estruturada as entidades, relacionamentos e atributos previamente identificados. Esta fase tem como principal objetivo converter o diagrama



Entidade–Relacionamento (ER) num conjunto de tabelas (relações) que possam ser implementadas num sistema de gestão de bases de dados relacional.

Durante este processo, cada entidade forte identificada no modelo conceptual é mapeada diretamente para uma tabela, onde os seus atributos se tornam colunas, e a respetiva chave primária é mantida como identificador único. As entidades fracas e os relacionamentos são igualmente transformados em relações próprias ou em chaves estrangeiras (foreign keys), de acordo com a sua cardinalidade e dependência.

No contexto da plataforma SeePaw, este passo permite estruturar a informação relativa a Animais, Abrigos, Utilizadores, Apadrinhamentos, Adoções, entre outras, assegurando que o modelo lógico resultante reflete corretamente as regras de negócio e restrições de integridade definidas na fase conceptual.

## 2.1 Derivar relações para o modelo lógico de dados

### Entidades Fortes

As entidades CAA, Animal, Utilizador são entidades fortes, ou seja, possuem chaves primárias próprias que as identificam univocamente (CAAIId, AnimalId, UtilizadorId). No modelo lógico, cada uma destas entidades foi convertida diretamente numa tabela, incluindo todos os seus atributos simples.

### Relacionamentos 1:N (um-para-muitos)

Este é o tipo de relação recorrente no modelo conceptual. Os relacionamentos:

- Um CAA gere vários Animais → (1:N)
- Um Utilizador pode ser owner de vários animais → (1:N)

De acordo com a metodologia de Connolly & Beg, nestas situações o atributo chave primária da entidade do lado "1" é copiado para a entidade do lado "N" como chave estrangeira (FK).

### Relacionamentos N:M (muitos-para-muitos)

O modelo conceptual inclui relações que envolvem múltiplas ocorrências de cada lado, como:

- Um Utilizador pode ter vários Animais Favoritos, e cada Animal pode ser favorito de vários utilizadores.
- Um Utilizador pode Apadrinhar vários Animais, e cada Animal pode ser apadrinhado por vários utilizadores.
- Um Utilizador pode ter atividades com vários Animais, e cada animal pode ter atividades (momentos distintos) com vários utilizadores.

Segundo a metodologia de Connolly & Beg, cada relação N:M é transformada numa nova tabela associativa, contendo as chaves primárias de ambas as entidades participantes (como foreign keys), podendo também incluir atributos próprios do relacionamento (como DataCriação, Montante, Estado, etc.).

### Atributos Multivalorados

No modelo conceptual do SeePaw, a entidade Animal possui o atributo ImagemExtraUrl, que representa as imagens adicionais associadas a cada Animal. Como um animal pode ter várias imagens, este é considerado um atributo multi-valorado.

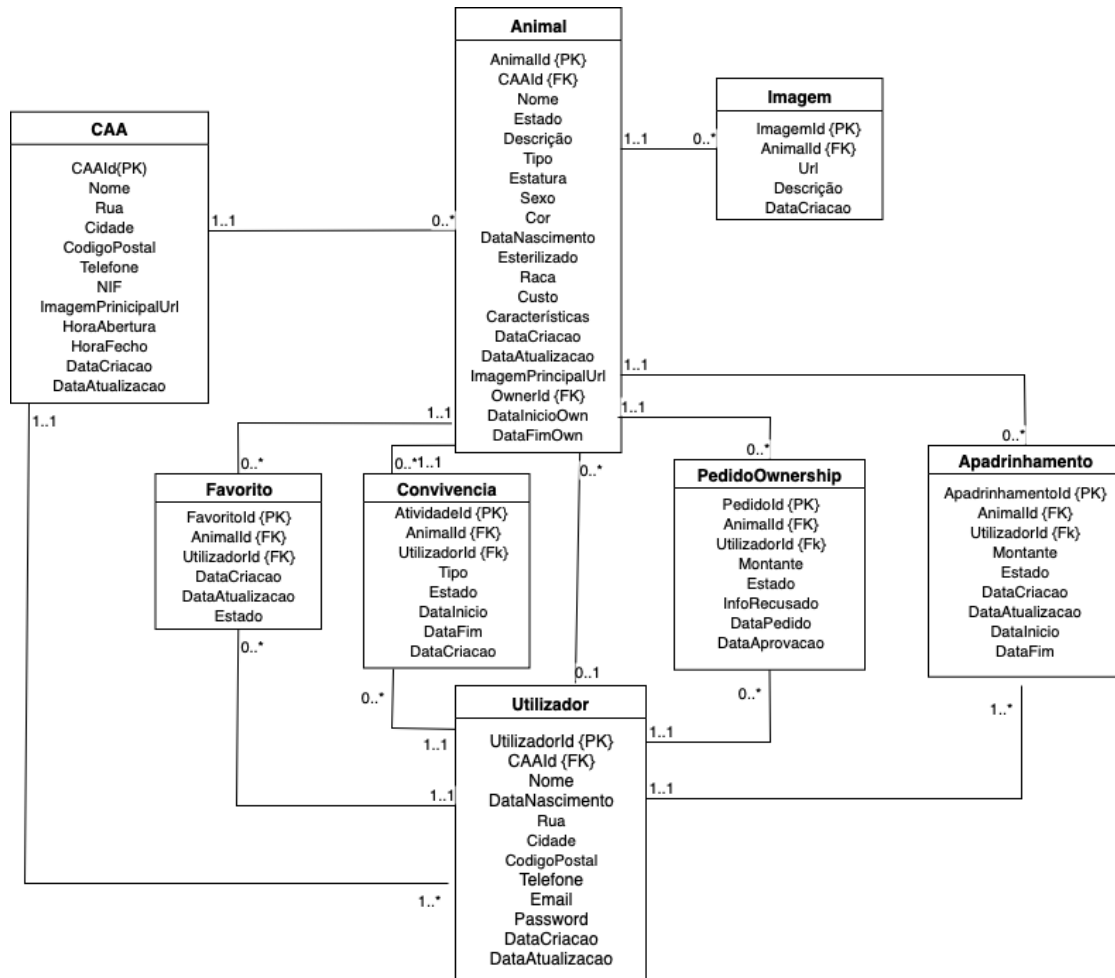
De acordo com a metodologia descrita por Connolly & Begg, sempre que um atributo é multivalorado, deve ser criada uma relação (tabela) para o representar. Esta nova tabela inclui o atributo multivalorado propriamente dito (Url), e a chave primária da entidade original (AnimalId) como chave estrangeira (FK).

O par formado pela chave primária da entidade original e o valor do atributo multivalorado compõe a chave primária composta da nova relação, garantindo que cada combinação Animal + Imagem é única.

A tabela PedidoOwnership constitui um caso particular. Segundo as regras de modelação, o relacionamento entre Utilizador (que submete o pedido) e Animal (objeto do pedido) seria de 1 para N, o que, em teoria, poderia ser representado apenas com uma *foreign key* no lado “muitos”. No entanto, por razões funcionais e de negócio, foi necessário criar uma tabela específica para registar os pedidos de ownership, de modo a armazenar informação temporal e de estado (como *DataInício*, *Montante*, *Termos* e *Estado do Pedido*).

Assim, a criação desta tabela vai além da simples tradução estrutural do modelo conceptual é uma decisão de modelação orientada pelos requisitos funcionais da plataforma SeePaw, permitindo acompanhar o histórico e o estado de cada pedido, algo que uma simples relação 1:N não seria capaz de representar adequadamente.

## 2.2 Diagrama E/R com identificação das chaves primárias



## 3 Dicionário de Dados

Os Tipos de Dados estão de acordo com o software de Base de Dados que é o SQLite.

### 3.1 Tabela: CAA

Atributo	Descrição	Tipo de Dados (SQLite)	Nulls	Multi-valor	Derivado	Valor por Defeito	Chave / Restrição
CAAId	Identificador único do CAA	TEXT(10)	Não	Não	Não	-	PK
Nome	Nome do CAA	TEXT(255)	Não	Não	Não	-	
Rua	Rua do CAA	TEXT(255)	Não	Não	Não	-	
Cidade	Cidade do CAA	TEXT(255)	Não	Não	Não	-	

Codigo Postal	Código Postal do CAA	TEXT(255)	Não	Não	Não	-	
Telefone	Número de telemóvel	INTEGER	Não	Não	Não	-	CHECK(LENGTH(Telefone)=9)
NIF	NIF do CAA	INTEGER	Não	Não	Não	-	UNIQUE & CHECK(LENGTH(NIF)=9)
Imagem PrincipalUrl	Url da imagem do CAA	TEXT(255)	Não	Não	Não	-	
HoraAbertura	Hora de abertura do CAA	TEXT(20)	Não	Não	Não	-	CHECK(horaAbertura GLOB '[0-2][0-9]:[0-5][0-9]')
HoraFecho	Hora de fecho do CAA	TEXT(20)	Não	Não	Não	-	TEXT CHECK(horaFecho GLOB '[0-2][0-9]:[0-5][0-9]')
DataCriacao	Data de criação do registo	DATETIME	Não	Não	Derivado	CURRENT_TIMESTAMP	
DataAtualizacao	Data de atualização do registo	DATETIME	Sim	Não	Não	-	

### 3.2 Tabela: Animal

Atributo	Descrição	Tipo de Dados (SQLite)	Nulos	Multivalor	Derivado	Valor por Defeito	Chave / Restrição
AnimalId	Identificador único do animal	TEXT(10)	Não	Não	Não	-	PK
CAAIId	CAA responsável	TEXT(10)	Não	Não	Não	-	FK → CAA(CAAId)
Nome	Nome do animal	TEXT(255)	Não	Não	Não	-	
Estado	Estado do animal	TEXT(15)	Não	Não	Não	-	CHECK("Disponível", "Apadrinhado",

							“Apadrinhado Totalmente”, “Ownership”, “Inativo”)
Descrição	Descrição do animal	TEXT(250)	Não	Não	Não	-	
Tipo	Espécie do animal	TEXT(50)	Não	Não	Não	-	CHECK(Tipo IN ('Cão','Gato','Outro'))
Estatura	Porte do animal	TEXT(15)	Não	Não	Não	-	CHECK(Estatura IN ('Pequeno','Médio','Grande'))
Sexo	Sexo do animal	TEXT(15)	Não	Não	Não	-	CHECK(Sexo IN ('Macho','Fêmea'))
Cor	Cor do animal	TEXT(15)	Não	Não	Não	-	
Data Nascimento	Data de Nascimento do animal	DATETIME	Não	Não	Não	-	
Esterilizado	Indica se o animal é esterilizado	BOOLEAN	Não	Não	Não	-	
Raca	Raça do animal	TEXT(15)	Não	Não	Não	-	
Custo	Custo mensal de manutenção	REAL	Não	Não	Não	0,0	CHECK(Custo > 0)
Características	Traços e temperamento	TEXT(255)	Não	Não	Não	-	
Data Criação	Data de criação do registro	DATETIME	Não	Não	Derivado	CURRENT_TIMESTAMP	
Data Atualização	Data de atualização do registro	DATETIME	Sim	Não	Não	-	
Imagem PrincipalUrl	URL da imagem principal	TEXT(255)	Não	Não	Não	-	

OwnerId	Caso tenha Owner, o identificador único dele	TEXT(10)	Sim	Não	Não	-	FK → Utilizador(UtilizadorId)
DataInicioOwn	Data de Inicio do Ownership	DATETIME	Sim	Não	Não	-	
DataFimOwn	Data de fim do Ownership	DATETIME	Sim	Não	Não	-	

### 3.3 Tabela: Imagem

Atributo	Descrição	Tipo de Dados (SQLite)	Nulos	Multivalor	Derivado	Valor por Defeito	Chave / Restrição
ImageId	Identificador único da imagem	TEXT(10)	Não	Não	Não	-	PK
Url	URL da imagem	TEXT(255)	Não	Não	Não	-	
AnimalId	Animal associado	TEXT(10)	Não	Não	Não	-	FK → Animal(AnimalId)
Descrição	Descrição da imagem	TEXT(255)	Não	Não	Não	-	
DataCriação	Data de criação do registo	DATETIME	Não	Não	Derivado	CURRENT_TIMESTAMP	

### 3.4 Tabela: Utilizador

Atributo	Descrição	Tipo de Dados (SQLite)	Nulos	Multivalor	Derivado	Valor por Defeito	Chave / Restrição
----------	-----------	------------------------	-------	------------	----------	-------------------	-------------------

UtilizadorId	Identificador único do utilizador	TEXT(10)	Não	Não	Não	-	PK
CAAIId	Utilizador administrativo CAA	TEXT(10)	Sim	Não	Não	-	FK → CAA(CAAId)
Nome	Nome do utilizador	TEXT(255)	Não	Não	Não	-	
Email	Email do utilizador	TEXT(255)	Não	Não	Não	-	UNIQUE
Rua	Rua do utilizador	TEXT(255)	Não	Não	Não	-	
Cidade	Cidade do utilizador	TEXT(255)	Não	Não	Não	-	
CodigoPostal	Código Postal do utilizador	TEXT(255)	Não	Não	Não	-	
Telefone	Número de telemóvel	INTEGER	Não	Não	Não	-	CHECK(LENGTH(Telefone)=9)
DataNascimento	Data de nascimento	DATE	Não	Não	Não	-	
Password	Palavra-passe encriptada	TEXT(255)	Não	Não	Não	-	
DataCriação	Data e hora do registo	DATE TIME	Não	Não	Derivado	CURRENT_TIMESTAMP	
DataAtualização	Data de atualização do registo	DATE TIME	Sim	Não	Não	-	

### 3.5 Tabela: Favorito

Atributo	Descrição	Tipo de Dados (SQLite)	Nulos	Multivalor	Derivado	Valor por Defeito	Chave / Restrição
Favoritoid	Identificador do favorito	TEXT(10)	Não	Não	Não	-	PK
UtilizadorId	Utilizador associado	TEXT(10)	Não	Não	Não	-	FK → Utilizador(UtilizadorId) & UNIQUE
AnimalId	Animal favorito	TEXT(10)	Não	Não	Não	-	FK → Animal(AnimalId) & UNIQUE
Estado	Animal ainda é um favorito	TEXT(10)	Não	Não	Não	-	CHECK(IEstado IN ('Ativo','Inativo'))
DataCriação	Data em que foi marcado como favorito	DATETIME	Não	Não	Derivado	CURRENT_TIMESTAMP	
DataAtualização	Data de atualização do registo	DATETIME	Sim	Não	Não	-	

### 3.6 Tabela: Apadrinhamento

Atributo	Descrição	Tipo de Dados (SQLite)	Nulos	Multivalor	Derivado	Valor por Defeito	Chave / Restrição
Apadrinhamentoid	Identificador do apadrinhamento	TEXT(10)	Não	Não	Não	-	PK
UtilizadorId	Utilizador que apadrinha	TEXT(10)	Não	Não	Não	-	FK → Utilizador(UtilizadorId) & UNIQUE



AnimalId	Animal apadrinhado	TEXT(10)	Não	Não	Não	-	FK → Animal(AnimalId) & UNIQUE
Montante	Valor mensal da contribuição	REAL	Não	Não	Não	0,0	CHECK(Montante > 0)
Estado	Estado do apadrinhamento	TEXT(15)	Não	Não	Não	-	CHECK(Estado IN ('Ativo','Cancelado','Concluido'))
DataInicio	Data de início do apadrinhamento	DATE	Não	Não	Não	-	
DataFim	Data de término (se aplicável)	DATE	Sim	Não	Não	-	
DataCriacao	Data e hora do registo	DATE TIME	Não	Não	Derivado	CURRENT_TIMESTAMP	
DataAtualizacao	Data de atualização do registo	DATE TIME	Sim	Não	Não	-	

### 3.7 Tabela: PedidoOwnership

Atributo	Descrição	Tipo de Dados (SQLite)	Nulos	Multivalor	Derivado	Valor por Defeito	Chave / Restrição
PedidoOwnershipId	Identificador do pedido	TEXT(10)	Não	Não	Não	-	PK
UtilizadorId	Utilizador que efetuou o pedido	TEXT(10)	Não	Não	Não	-	FK → Utilizador(UtilizadorId) & UNIQUE
AnimalId	Animal solicitado	TEXT(10)	Não	Não	Não	-	FK → Animal(AnimalId) & UNIQUE
Estado	Estado atual do pedido	TEXT(15)	Não	Não	Não	Pendente	CHECK(Estado IN ('Pendente',

							'Em análise', 'Aprovado', 'Rejeitado'))
InfoRecusado	Razão de ter sido recusado	TEXT(255)	Sim	Não	Não	-	
Montante	Montante a pagar pelo animal	REAL	Não	Não	Não	0,0	
DataPedido	Data e hora do pedido	DATETIME	Não	Não	Derivado	CURRENT_TIMESTAMP	
DataAprovação	Data e hora do pedido	DATETIME	Sim	Não	Não	-	

### 3.8 Tabela: Convivência

Atributo	Descrição	Tipo de Dados (SQLite)	Nulos	Multivalor	Derivado	Valor por Defeito	Chave / Restrição
AtividadeId	Identificador da convivência	TEXT(10)	Não	Não	Não	-	PK
AnimalId	Animal associado	TEXT(10)	Não	Não	Não	-	FK → Animal(AnimalId)
UtilizadorId	Utilizador associado	TEXT(10)	Não	Não	Não	-	FK → Utilizador(UtilizadorId)
Tipo	Tipo de relação entre o animal e o utilizador	TEXT(15)	Não	Não	Não	-	CHECK(Tipo IN ('Apadrinhamento', 'Owner'))
Estado	Estado da convivência	TEXT(15)	Não	Não	Não	-	CHECK(Estado IN ('Ativo', 'Cancelado', 'Efetuado'))

DataInicio	Data e hora do início da convivência	DATETIME	Não	Não	Não	-	
DataFim	Data e hora do fim da convivência	DATETIME	Não	Não	Não	-	
DataCriacao	Data e hora do registro	DATETIME	Não	Não	Derivado	CURRENT_TIMESTAMP	