



**Universidade do Minho**  
Licenciatura em Engenharia Informática

## **Unidade Curricular de Bases de Dados**

Ano Letivo de 2022/2023

### **Os Bichinhos**

**Diogo Paiva (100760), João Magalhães  
(A100740), Jorge Rodrigues (A101758),  
Rodrigo Gomes (100555)**

Junho, 2023

# BD

Data de Recepção	
Responsável	
Avaliação	
Observações	

## Os Bichinhos

**Diogo Paiva (100760), João Magalhães  
(A100740), Jorge Rodrigues (A101758),  
Rodrigo Gomes (100555)**

<<Junho,2023>>

# Resumo

No âmbito da unidade curricular Bases de Dados, foi desenvolvido um projeto que objetivou a consolidação das aprendizagens da unidade curricular, bem como colocar os alunos na posição de “Arquiteto de Bases de Dados”, testando assim as suas capacidades.

O programa desenvolvido visa auxiliar o veterinário Rodrigo e o enfermeiro João a gerir a associação “Os Bichinhos”, uma associação que ambiciona cuidar de animais abandonados ou vítimas de maus tratos. O projeto foi desenvolvido em conformidade com a ordem dos conceitos estudados nas aulas, sendo então estruturado à volta do Ciclo de Vida do desenvolvimento de uma Base de Dados.

Posto isto, concebemos todas as partes importantes para o sistema, incluindo um pequeno contexto e fundamentação do problema. Além disso, consultando os dois voluntários, fizemos uma lista dos requisitos, agrupando os por categoria, que serviram de base para o modelo conceptual do projeto. De seguida, desenvolvemos o modelo lógico e consideramos as interrogações dos futuros utilizadores de forma a garantir que satisfaz as necessidades destes. A quinta etapa é a implementação física, para isso utilizamos o “MySQL”, pois é a ferramenta utilizada na unidade curricular. Todas as fases anteriores foram devidamente validadas, podendo recorrer à opinião dos utilizadores. As etapas seguintes consistem na utilização de um programa desenvolvido por nós para importar dados para a Base de Dados, e posteriormente visualiza-los e realizar estatísticas com o “Power Bi”.

Assim, com todas as fases do desenvolvimento concluídas, podemos dar como terminado o projeto de Bases de Dados.

**Área de Aplicação:** Arquitetura e Desenvolvimento de Sistemas de Bases de Dados.

**Palavras-Chave:** Bases de Dados, Requisitos, Modelo Conceptual, Modelo Lógico, Normalização de dados, Validação do projeto, Interrogações de Utilizadores, MySQL, Power Bi.

# Índice

Resumo	iv
Índice v	
Índice de Figuras	vii
Índice de Tabelas	ix
1. Definição do Sistema	1
1.1. Contexto de aplicação	1
1.2. Fundamentação do sistema	1
1.3. Motivação e Objetivos	2
1.4. Análise da viabilidade do processo	3
1.5 Recursos e Equipa de Trabalho	4
1.6. Plano de Execução do Projeto	5
2. Levantamento e Análise de Requisitos	6
2.1. Método de levantamento e de análise de requisitos adotado	6
2.2. Organização dos requisitos levantados	7
2.2.1 Requisitos de descrição	7
2.2.2 Requisitos de exploração	9
2.2.3 Requisitos de controlo	10
2.3. Análise e validação geral dos requisitos	10
3. Modelação Conceptual	11
4. Modelação lógica	22
4.1 Construção e validação do modelo de dados lógico	22
4.2 Normalização de Dados	24
4.3 Apresentação e explicação do modelo lógico produzido	26
4.4 Validação do modelo com interrogações do utilizador	31
5. Implementação Física	34
5.1 Tradução do esquema lógico para o sistema de gestão de bases de dados escolhido em SQL	34
5.2 Tradução das interrogações do utilizador para SQL (alguns exemplos)	35
5.3 Definição e caracterização das vistas de utilização em SQL (alguns exemplos)	36
5.4 Cálculo do espaço da Bases de Dados (inicial e taxa de crescimento anual)	37
5.5 Indexação do Sistema de Dados	39
5.7 Plano de segurança e recuperação de dados	40
6. Implementação do Sistema de Recolha de Dados	43
6.1 Apresentação e modelo do sistema	43

6.2 Implementação do sistema de recolha	44
6.3 Funcionamento do sistema	46
7. Implementação do Sistema de Painéis de Análise	47
7.1 Definição e caracterização da vista de dados para análise	47
7.2 Povoamento das estruturas de dados para análise	47
7.3 Apresentação e caracterização dos dashboards implementados	48
8. Conclusão	49
9. Referências Bibliográficas	50

# Índice de Figuras

Figura 1- Diagrama de Gantt	5
Figura 2- Relacionamento Doador-Donativo	12
Figura 3- Relacionamento Donativo-Comprovativo	13
Figura 4- Relacionamento Funcionário-Donativo	13
Figura 5- Relacionamento Animal–Funcionário	14
Figura 6- Relacionamento Animal–Adotante	14
Figura 7- Modelo Conceptual	18
Figura 8- Entidade Animal	18
Figura 9- Entidade Donativo	19
Figura 10- Entidade Doador	19
Figura 11- Entidade Funcionário	19
Figura 12- Entidade Adotante	19
Figura 13- Entidade Comprovativo	20
Figura 14- Relacionamento Animal-Adotante	20
Figura 15- Relacionamento Animal-Funcionário	20
Figura 16- Relacionamento Funcionário-Donativo	21
Figura 17- Modelo Lógico	26
Figura 18- Relacionamento entre as tabelas Animal, TB_AnimalFuncionário e Funcionário	27
Figura 19- Relacionamento entre as tabelas Funcionário, TB_FuncionárioDonativo e Donativo	27
Figura 20- Relacionamento entre as tabelas Funcionário e Disponibilidade	28
Figura 21- Relacionamento entre as tabelas Funcionário e Competências	28
Figura 22- Relacionamento entre as tabelas Adotante e Contacto	29
Figura 23- Relacionamento entre as tabelas Donativo e Doador	29
Figura 24- Relacionamento entre as tabelas Donativo e Comprovativo	30
Figura 25- Relacionamento entre as tabelas Animal e Adotante	30
Figura 26- Código “MySql” da primeira interrogação	35
Figura 27- Código “MySql” da segunda interrogação	35
Figura 28- Código “MySql” da terceira interrogação	36
Figura 29- Código “MySql” da quarta interrogação	36
Figura 30- Código “MySql” da primeira vista	36
Figura 31- Código “MySQL” da segunda vista	37
Figura 32- Código “MySQL” da terceira vista	37
Figura 33- Dados referentes aos animais recolhidos no período entre o ano de 2018 e 2021, a nível nacional.	39
Figura 34- Código “MySQL” dos índices	39

Figura 35- Contas dos Administradores	40
Figura 36- Conta do voluntário "Duarte" e conta genérica	41
Figura 37- Código que realiza Backup	41
Figura 38- Programa que carrega um Backup	42
Figura 39- Inserção dos Animais em "csv"	44
Figura 40- Inserção dos Adotantes em "csv"	45
Figura 41- Execução do Código	45
Figura 42-Dashboard	48



# Índice de Tabelas

Tabela 1- Caracterização dos atributos dos funcionários	15
Tabela 2- Caracterização dos atributos dos donativos	15
Tabela 3- Caracterização dos atributos dos doadores	16
Tabela 4- Caracterização dos atributos dos adotantes	16
Tabela 5- Caracterização dos atributos dos animais	16
Tabela 6- Caracterização dos atributos dos comprovativos	17
Tabela 7- Tabela do tamanho utilizado pelas entidades na associação “Os bichinhos”	38

# **1. Definição do Sistema**

## **1.1. Contexto de aplicação**

Abandonar ou maltratar um animal de estimação pode ser considerado cruel e desumano. Embora a negligência face aos cuidados dos animais apresenta sinais de diminuição em Portugal, foi estimado que em 2020 tenham sido recolhidos mais de 30000 animais errantes, abandonados ou vítimas de maus tratos. Os animais que chegam às instituições provêm de ambientes diferentes, sendo que alguns resultam de maus tratos dos donos, outros de abandono e ainda há casos de animais que nunca tiveram uma família em primeiro lugar.

Para combater os maus tratos aos animais, a associação “Os Bichinhos” foi fundada no ano de 2009, e está localizada em Braga na Rua dos Capelistas desde o ano de 2010. Criada com o objetivo de acolher animais abandonados na cidade e arredores, a associação “Os Bichinhos” trabalha incansavelmente para fornecer um lar temporário seguro e amoroso para os animais que se encontram em situação de risco.

Os animais acolhidos pela associação “Os Bichinhos” são maioritariamente animais da cidade de Braga, isto porque a sede encontra-se numa das ruas principais da cidade, mas também não recusa ajudar animais provenientes de zonas mais rurais. Desta forma, é comum animais sem dono prévio necessitem de apoio ao qual a fundação age. Para além disso, encontramos também animais mais fora do comum animal doméstico, como por exemplo, répteis ou animais de quinta.

## **1.2. Fundamentação do sistema**

A associação depende do apoio financeiro e voluntário da comunidade para continuar a oferecer cuidados de qualidade para os animais. Os gastos com alimentação, medicamentos e outras necessidades dos animais são significativos, e a ajuda da comunidade é fundamental para garantir que os animais recebam a atenção necessária enquanto esperam por uma família adotiva.

Além dos voluntários temporários, a associação conta com uma equipa de dois membros fixos, o Rodrigo e o João, que exercem a profissão de veterinário e de enfermeiro veterinário/secretário respetivamente. Tanto o Rodrigo e o João têm um emprego fora a

associação dando um total de oito a dez horas semanais cada um. A gestão do centro é feita pelo João que utiliza o programa “Excel” da coleção de ferramentas da Microsoft para guardar dados à cerca dos animais que se encontram no centro, como também dos interessados em adotar um animal de companhia.

Dois colegas de casa, Jorge e Diogo, ambos estudantes de Engenharia Informática, decidiram adotar um gato da associação local de proteção animal. Eles já moravam juntos há alguns meses e estavam à procura de um animal de estimação para trazer alegria e vida para a casa. Quando chegaram ao centro, ficaram surpreendidos com a desorganização das fichas dos animais e dos respetivos novos donos. Movidos pela falta de apoio aos profissionais que fazem parte de uma causa tão nobre, os estudantes decidiram iniciar o desenvolvimento de um software especializado na área de gestão de centros de adoção com o objetivo de facilitar o trabalho dos voluntários. Para concretizar o propósito de simplificar as tarefas dos trabalhadores, os estudantes criarão um programa com as seguintes capacidades de gestão:

- Chegadas e saídas de animais.
- Gestão de doações.
- Despesas ligadas às necessidades dos animais.
- Informação dos novos donos.
- Gestão de voluntários.

### **1.3. Motivação e Objetivos**

Ao longo de mais de uma década no ativo, a associação “Os Bichinhos” têm enfrentado vários desafios incluindo a gestão adequada dos animais, programas de adoção e a coordenação de voluntários. Para ajudar a enfrentar esses desafios, a criação de uma base de dados pode ser uma solução eficaz. Os membros líderes da associação definiram os seguintes motivos para a troca do sistema atual para uma base de dados:

- Gestão de animais: Uma base de dados pode ajudar a gerir os animais, registando informações importantes sobre cada animal, como nome, raça, idade, histórico médico e outros cuidados especiais.
- Registo de voluntários: é importante manter um registo atualizado dos voluntários da associação, incluindo informações como o nome, contactos, habilidades, horas de disponibilidade e tarefas que desempenham.

- Saídas e entradas de animais: A base de dados pode ajudar a acompanhar o processo de adoção dos animais, registando informações sobre os novos donos, como o nome, morada, e-mail e outras informações relevantes.
- Recolha de donativos: A base de dados pode ser usada para gerir as doações feitas à equipa, registando informações sobre as doações, como o nome do doador, a data e o valor da doação.
- Monitorização de recursos: A base de dados pode ajudar a monitorizar as despesas da associação, registando informações sobre os produtos, como o tipo de produto, a data, o valor, o fornecedor e a quantidade armazenada.

Rodrigo e João, já familiarizados com todo este ramo de acolhimento animal, definiram alguns objetivos que querem ver cumpridos, com o auxílio do novo sistema de base de dados que será futuramente implementado:

- Diminuir os gastos com recursos essenciais para os animais,
- Definir horários para os voluntários,
- Facilitar o processo de adoção,
- Tornar o serviço mais especializado para cada animal,
- Tornar o acesso às informações dos animais mais rápido e eficaz

## 1.4. Análise da viabilidade do processo

O Rodrigo e o João acreditam vivamente de que uma atualização para o digital é essencial para o desempenho da organização. Uma base de dados proporcionaria, de forma eficiente e segura, uma nova forma de guardar dados importantes com um acesso mais fácil comparativamente ao modelo de armazenamento atual. Os dois colegas calculam que, o novo modelo, promoverá:

- Melhor gestão de recursos, recuperando assim num intervalo de 3 meses um valor próximo de 20% das despesas do último ano, o que por si só, é suficiente para pagar o investimento feito pela associação.
- Saber de forma instantânea informação à cerca dos animais, o que é vitalício para as diferentes funções que asseguram o bem-estar dos mesmos.
- Facilitar o processo de adoção para os futuros donos dos animais de companhia, porque o procedimento seria mais rápido e conveniente para os utilizadores.
- Determinar padrões nas necessidades dos animais, desta forma é possível realizar estimativas mais realistas do que estes realmente precisam os.
- Favorecer o trabalho voluntário através de um sistema de atribuição de turnos cómodos tendo em conta o tempo disponibilizado por cada voluntário.

## 1.5 Recursos e Equipa de Trabalho

Os recursos para o desenvolvimento do projeto são divididos em dois grupos:

- Humanos
  - Membros da associação, voluntários e os estudantes convidados.
- Materiais
  - Hardware (1 servidor, 2 computadores dos funcionários).
  - Software (MySQL, brModelo, Power BI).

A equipa de trabalho será constituída pelos seguintes elementos:

- Pessoal interno
  - O veterinário Rodrigo, o enfermeiro veterinário João e os voluntários
    - Funcionamento da associação.
    - Tratamento dos animais.
    - Gestão de recursos.
    - Transferir os animais para os donos.
- Pessoal externo
  - Os estudantes de engenharia informática Jorge e Diogo
    - Levantamento dos requisitos.
    - Modelação do sistema.
    - Implementação do sistema físico.
    - Implementação do sistema de recolha de dados.
    - Implementação do sistema de painéis de análise.
- Outros
  - Novos donos de animais posteriormente ao cuidado da instituição e doadores
    - Inquérito de opiniões e validação do serviço.

## 1.6. Plano de Execução do Projeto

Um plano de execução é fulcral para manter a organização do projeto. Por isso, os estudantes e trataram de criar um plano sob a forma de diagrama de Gantt. Para esse efeito, foram realizadas reuniões com o João e o Rodrigo, de forma a conceber um plano que satisfazia todos os intervenientes.

Apesar de o plano ter sido realizado com cuidado, os alunos, com alguma ingenuidade, os alunos não consideraram aspetos importantes como os feriados, as épocas de frequências e também outros trabalhos para outras unidades curriculares. Por esse motivo, a partir da modelação conceptual, existiram alguns atrasos que atrasaram o desenvolvimento do projeto. Ainda assim, conseguimos terminar todas as etapas antes da data de entrega prevista.

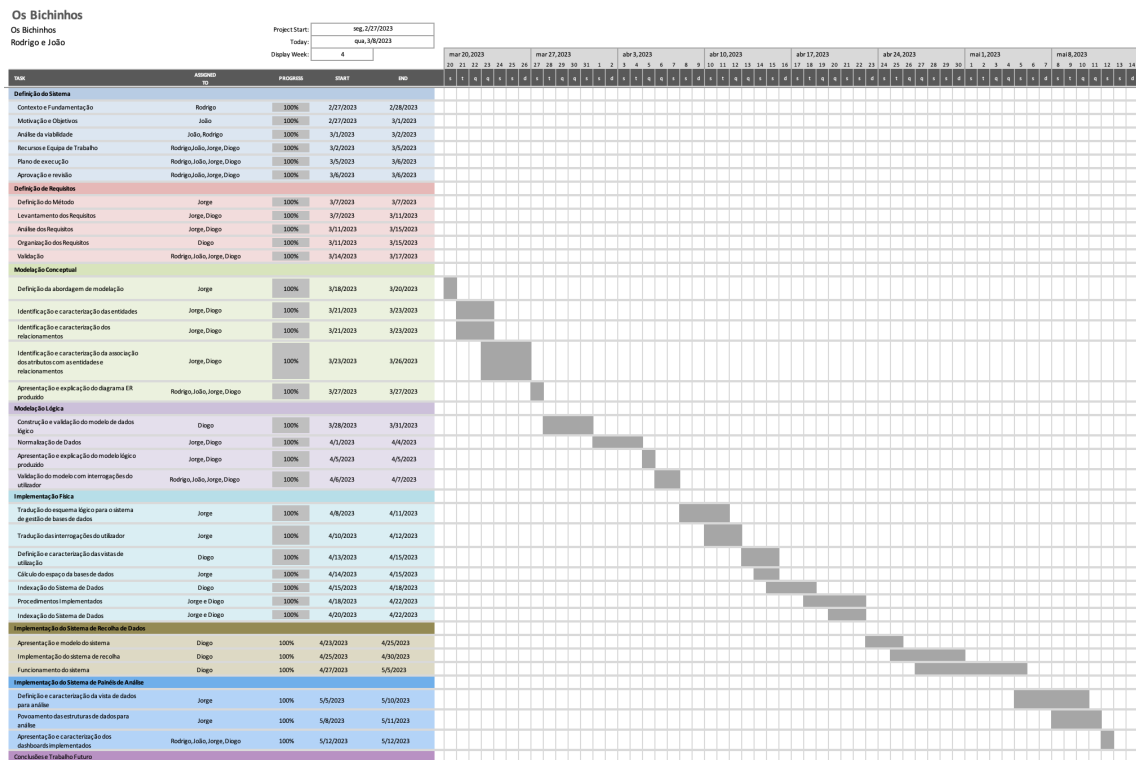


Figura 1- Diagrama de Gantt

## **2. Levantamento e Análise de Requisitos**

### **2.1. Método de levantamento e de análise de requisitos adotado**

A fim de conceber uma Base de Dados robusta, que assegure as funcionalidades que permitem a atividade da associação, é necessário recorrer a diversos métodos de obtenção de informação.

O levantamento da informação permite aos dois estudantes juntamente com os membros efetivos da associação estabelecer um número finito de requisitos que sejam desejados pelos futuros usuários da base de dados. A lista de requisitos é a base para os modelos que vão ser arquitetados, e também serve como pilar das capacidades da Base de Dados, podendo com a revisão prévias dos membros, ser modificada.

Os métodos de recolha utilizados neste projeto foram os seguintes:

- **Reuniões:** No decorrer do levantamento dos requisitos foram realizadas reuniões com os dois membros efetivos da associação. Desta forma, conseguimos entender as necessidades que um funcionário precisa para desempenhar as suas funções, e também que permissões cada utilizador terá no acesso à Base de Dados.
- **Observação:** Em simultâneo com as reuniões, os estudantes tiveram permissão de assistir ao funcionamento da associação. Deste modo, seria possível comparar algumas das descrições fornecidas nas reuniões com a perspetiva de um arquiteto de Bases de Dados. Assim, é quase garantido que todos os dados relevantes à atividade da associação são mantidos seguros na Base de Dados.
- **Questionários:** A comunicação com membros que se encontram fora da associação também é relevante para quem planeia a Base de Dados. Em virtude de melhorar as taxas de adoção e os donativos, a satisfação dos adotantes e doadores é sempre um aspeto considerável.

## 2.2. Organização dos requisitos levantados

Com a aplicação dos diferentes métodos de levantamento, por fim, os estudantes conceberam um documento de requisitos. Na versão final do documento, é esperado que todas necessidades dos funcionários sejam cumpridas, e também que guie os estudantes até à próxima fase do projeto.

O documento de recolha dos estudantes está dividido em três vertentes de requisitos: Descrição, Manipulação e Controlo. Individualmente, cada requisito contém uma data e hora, descrição, a área onde se aplicam, a fonte para o requisito e por fim o analista que o levantou.

### 2.2.1 Requisitos de descrição

A primeira vertente é a de descrição. Esta agrupa os requisitos que referiam a criação de objetos na base de dados. Aqui conseguimos observar os primeiros passos da formação de uma entidade do futuro modelo conceptual.

Cada entrada segue a seguinte estrutura: N°; *Data e hora*; *Descrição*; *Área*; *Fonte*; *Analista*

1. 07/03/2023 17:00; Cada animal da associação deverá ser registado através de um "Id" único; Funcionários; Rodrigo; Diogo
2. 07/03/2023 19:00; É necessário armazenar dados do animal, nomeadamente, nome, idade, perfil do animal, data de nascimento, registo clínico, raça/espécie, sexo, peso, categoria do animal, data de adoção, cor, data de chegada e data de saída; Funcionários; Rodrigo; Diogo
3. 08/03/2023 12:00; Caso o animal seja resgatado da rua e não seja possível obter qualquer informação pessoal do mesmo, atributos como idade, nome, data de nascimento e registo clínico devem ser preenchidos a vazio; Funcionários; Rodrigo; Diogo
4. 08/03/2023 12:00; Aquando da adoção do animal, a data de adoção é registada obrigatoriamente; Funcionários; Rodrigo; Diogo
5. 10/03/2023 17:00; Os animais devem estar agrupados por categorias, facilitando a sua localização; Funcionários; João; Diogo
6. 07/03/2023 17:00; Cada donativo feito à associação deve ser registado com um "id" único; Funcionários; Rodrigo; Diogo
7. 10/03/2023 17:00; É necessário armazenar informações relativas aos donativos recebidos, nomeadamente a data de validade do donativo, se necessário, a quantidade do mesmo e a sua categoria; Funcionários; João; Jorge
8. 07/03/2023 17:00; Cada entidade doadora deve ser registada com um "id"; Funcionários; Rodrigo; Diogo
9. 10/03/2023 17:00; É necessário armazenar informações relativas a quem fez os donativos, nomeadamente o nome de quem fez a doação, o contacto, o endereço de correio eletrónico e a data; Funcionários; João; Jorge
10. 10/03/2023 17:00; Caso o donativo seja dinheiro, a data de validade e a quantidade deve ser preenchida. Por outro lado, se se tratar de um recurso para os animais, o comprovativo de transferência bancária não deve existir. Se houver doação de bens sem



data de expiração, atributos como data de validade não devem ser preenchidos; Funcionários; João; Jorge

11. 11/03/2023 22:00; Para os funcionários da associação, em termos de estatuto, apenas se pretende considerar profissionais e voluntários, onde os profissionais terão funções e permissões administrativas, sendo identificados também como administradores; Funcionários; João; Jorge
12. 09/03/2023 14:00; Funcionários podem ter mais que uma competência dentro da associação; Funcionários; João; Jorge
13. 07/03/2023 17:00; Cada funcionário da associação deverá ser registado por um "id"; Funcionários; Rodrigo; Diogo
14. 11/03/2023 22:00; É necessário guardar dados dos voluntários, tais como o seu nome, o contacto, o endereço de correio eletrónico, dias disponíveis, competências, estatuto e morada; Funcionários; João; Jorge
15. 07/03/2023 17:00; Cada adotante deve ser registado com um "id" único; Funcionários; Rodrigo; Diogo
16. 08/03/2023 11:00; Cada adotante deverá ficar registado no sistema com os seguintes dados: data de nascimento, sexo, morada, os contactos e a identificação do(s) animal/animais adotado(s); Funcionários; Rodrigo; Diogo
17. 08/03/2023 11:00; Cada pessoa que adota um animal, apesar de poder fazê-lo mais vezes, só fica registada uma vez no sistema. Novas adoções adicionam a identificação do animal adotado ao registo do adotante; Funcionários; João; Diogo
18. 07/03/2023 17:00; Os contactos correspondem a um ou mais números de telefone no caso dos adotantes. Nos restantes apenas é guardado um contacto; Funcionários; Rodrigo; Jorge
19. 07/03/2023 17:00; As moradas são compostas por Rua, Código-postal e número da porta; Funcionários; Rodrigo; Diogo
20. 10/03/2023 15:00; Um animal só pode ser adotado por um adotante; Funcionários; João; Jorge
21. 10/03/2023 15:00; Vários animais podem tirar partido do mesmo donativo feito à associação; Funcionários; João; Jorge
22. 10/03/2023 15:00; Os funcionários cuidam de um ou mais animais, assim como um animal pode receber tratamento de um ou mais funcionários, ficando registada a data do acesso ao animal; Funcionários; João; Jorge
23. 11/03/2023 18:00; Os funcionários devem conseguir aceder a um ou mais donativos, sendo registada a data de acesso; Funcionários; Rodrigo; Diogo
24. 11/03/2023 18:00; Os comprovativos devem conter informação relativa ao IBAN, o tipo de operação, o montante, o nome do banco e a data do movimento; Rodrigo; Diogo
25. 11/03/2023 18:00; Os Funcionários com o estatuto de Profissionais devem estar disponíveis todos os dias da semana; Rodrigo; Diogo

## 2.2.2 Requisitos de exploração

A segunda vertente é a vertente de exploração ou de manipulação. Os requisitos que aqui se encontram cobrem as diferentes funcionalidades que os funcionários precisam para a associação. Estas incluem exploração de dados, funções que manipulam dados entre outros.

Cada entrada segue a seguinte estrutura: N°; *Data e hora*; *Descrição*; *Área*; *Fonte*; *Analista*

1. 08/03/2023 12:00; Os funcionários devem ser capazes de, a qualquer momento, aceder à ficha dos animais; Funcionários; Rodrigo; Diogo
2. 08/03/2023 12:00; Os funcionários devem conseguir obter uma lista com todos os animais sem dono ordenados por data de chegada; Funcionários; Rodrigo; Diogo
3. 10/03/2023 17:00; Os administradores e os funcionários deverão poder listar os donativos por intervalos de tempo de validade; Funcionários; João; Jorge
4. 09/03/2023 14:00; A quantidade dos produtos deverá poder ser atualizada pelos funcionários; Funcionários; João; Jorge
5. 08/03/2023 11:00; Os funcionários devem conseguir aceder às informações dos adotantes através de um contacto do adotante; Funcionários; João; Jorge
6. 08/03/2023 11:00; A qualquer momento, os funcionários devem conseguir obter o capital da associação; Funcionários; João; Jorge
7. 10/03/2023 15:00; Os administradores devem conseguir saber se um adotante é também doador; Funcionários; João; Jorge
8. 11/03/2023 22:00; Os administradores devem conseguir visualizar o montante doado por cada doador caso a doação foi feita em dinheiro; Funcionários; João; Jorge
9. 11/03/2023 22:00; Animais cuja saída foi à mais de 60 dias devem ser removidos da Base de Dados; Funcionários; João; Jorge
10. 11/03/2023 22:30; Os doadores que mais donativos fizeram à associação; Funcionários; João; Jorge
11. 11/03/2023 22:30; Deve ser possível ver as fichas de todos os funcionários destacando a sua disponibilidade e competências; João; Jorge
12. 11/03/2023 22:30; Deve ser possível ver as fichas de todos os adotantes com todos os contactos do mesmo; João; Jorge
13. 11/03/2023 22:30; Deve ser possível ver os doadores que fizeram doação em dinheiro com os respetivos comprovativos; João; Jorge
14. 11/03/2023 22:30; Deve ser possível um funcionário conseguir ver que animais um doador apoiou com as suas doações; João; Jorge

### 2.2.3 Requisitos de controlo

Por fim, a última vertente é a de controlo. Esta visa restringir o acesso à base de dados por parte dos utilizadores, destacando que utilizadores podem ou não podem executar funções ou consultar informações das tabelas.

Cada entrada segue a seguinte estrutura: *Nº; Data e hora; Descrição; Área; Fonte; Analista*

1. 09/03/2023 14:00; Um animal só pode ser registado por um administrador; Funcionários; Rodrigo; Diogo
2. 09/03/2023 14:00; Apenas os administradores podem adicionar donativos à base de dados; Funcionários; Rodrigo; Jorge
3. 09/03/2023 14:00; Apenas os administradores podem consultar informações relativas a um donativo feito em dinheiro; Funcionários; Rodrigo; Jorge
4. 09/03/2023 14:00; Apenas os administradores terão a possibilidade de adicionar novos funcionários; Funcionários; Rodrigo; Jorge
5. 11/03/2023 22:00; Apenas os administradores terão a possibilidade de alterar os dados dos funcionários, à exceção do campo "dias disponíveis", o qual poderá ser acedido e alterado tanto por administradores como por funcionários; Funcionários; João; Jorge
6. 09/03/2023 14:00; Apenas os administradores podem adicionar ou remover adotantes.; Funcionários; Rodrigo; Jorge
7. 09/03/2023 14:00; Qualquer funcionário, independentemente do seu estatuto, pode alterar um registo já feito de um adotante; Funcionários; Rodrigo; Jorge
8. 09/03/2023 14:00; Qualquer funcionário, independentemente do seu estatuto, pode alterar um registo já feito de um donativo; Funcionários; Rodrigo; Jorge

## 2.3. Análise e validação geral dos requisitos

Em seguida ao levantamento dos requisitos, os membros intervenientes no projeto realizaram a validação dos mesmos. Os estudantes juntamente com os voluntários encontraram-se para debater a praticabilidade do trabalho elaborado até ao momento.

De forma geral, o processo ocorreu de forma plana e sem muitas queixas por parte dos funcionários da associação. Na maior parte dos casos, os funcionários não acharam a presença dos estudantes invasiva como também acharam que as perguntas e as reuniões foram bastante pertinentes.

Contudo, existiu momentos de alguma reclamação por parte dos voluntários e a existência de situações impraticáveis.

Inicialmente, os estudantes definiram algumas condições como a existência de apenas um contacto para os adotantes. Posteriormente, os funcionários destacaram a importância de um adotante poder ter mais do que um contacto, visto que há casos onde a ficha do adotante é

de um menor, sendo assim prudente guardar o contacto da entidade paternal, e também em casos de um animal pertencer a uma família, onde é conveniente ter o contacto de ambos os adultos.

Anteriormente, os funcionários também incidiram na possibilidade de manter uma ligação a cada animal que utiliza um determinado donativo. Os estudantes, já pensando na implementação dos requisitos num modelo, sugeriram que seria inconveniente para o funcionamento da Base de Dados manter essa relação, procurando uma solução melhor como guardar o acesso dos funcionários aos donativos em sua vez.

### 3. Modelação Conceptual

#### 3.1. Apresentação da abordagem de modelação realizada

Depois de obtermos todas as informações necessárias e definirmos os requisitos, é hora de começarmos a planejar a estrutura e o design da base de dados que queremos criar. Para facilitar esse processo, a utilização de um Diagrama ER (Entidade Relacionamento) é aconselhável, pois trata-se de um fluxograma que ilustra as relações entre diferentes entidades, tais como pessoas, objetos, conceitos, entre outros. A construção desse diagrama é trivial e envolve a identificação das entidades que abrangem o problema, os seus relacionamentos e os atributos.

Essa modelagem conceitual deve ser representada de acordo com as necessidades de gestão, e, dessa forma, descrever um modelo comportamental. No nosso caso, realizamos essa modelagem com o objetivo de gerir efetivamente um sistema de informação relativo a um centro de adoções de animais, levando em conta as peculiaridades e particularidades de um sistema real.

#### 3.2. Identificação e caracterização das entidades

Durante a avaliação do funcionamento de uma associação de adoção de animais, identificamos as entidades essenciais para o seu gerenciamento eficiente. Essas entidades incluem funcionários, doações, doadores, adotantes e animais.

Para atender aos requisitos específicos dessa organização, cada entidade foi atribuída com um conjunto de dados elementares que ajudam a caracterizá-las de forma precisa. Os atributos atribuídos são os seguintes:

- **Funcionário:** é a entidade que representa quem trabalha para a associação. É identificado através de um id e é caracterizado pelo seu nome, estatuto, email, morada

(constituída por rua, código postal e nº da porta, sendo por isso um atributo composto), contacto, competências (constituída por uma ou mais sendo assim um atributo multivalorado) e a disponibilidade.

- **Donativo:** representa os donativos feitos à associação e é representado por um id, a categoria onde se enquadra, data de validade e a sua quantidade.
- **Comprovativo:** compõe as informações relativas ao comprovativo do donativo feito à associação e é representado por um id, um valor, IBAN e a data de quando foi feito a doação
- **Doador:** é a entidade que caracteriza as pessoas que contribuem com os donativos e são identificados por um id, nome completo, contacto, o comprovativo do donativo (podendo haver mais que um e daí ser um multivalorado), email e a data em que o donativo foi feito.
- **Adotante:** é as pessoas que adotam os animais presentes na associação e é caracterizado por um id, nome, data de nascimento, sexo, contacto (podendo este ser um ou mais números por isso, é um atributo multivalorado), e-mail e a morada (constituída por rua, código postal e nº da porta, sendo por isso um atributo composto).
- **Animal:** representa os animais que compõem a associação e são identificados por um id, nome, idade, perfil do animal, data de nascimento, registo clínico, categoria, cor, data da sua adoção, data da saída e de chegada à associação, sexo, raça e o seu peso.

Todos os “id’s” que são usados para identificar cada entidade são únicos e por isso, são chaves primárias.

Ao recolher e manter esses dados de forma sistemática, a associação pode gerenciar com eficiência todo o processo de adoção de animais e manter um registo completo de todas as informações relevantes relacionadas aos funcionários, animais, doadores, adotantes. Além disso, esses dados ajudam a gerenciar as doações que são recebidas.

### 3.3. Identificação e caracterização dos relacionamentos

Nesta modelação englobamos vários tipos de relacionamentos entre as entidades. Assim, apresentamos uma análise e explicação individual feita a cada um desses relacionamentos:

#### • Relacionamento Doador - Donativo

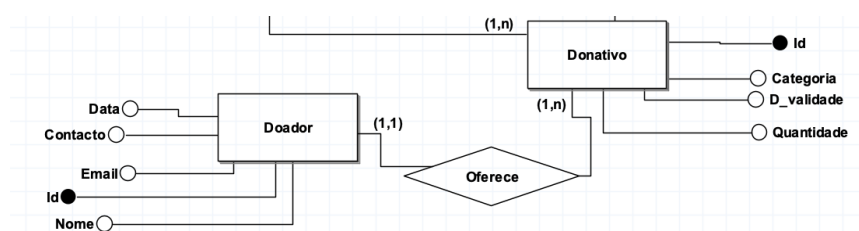


Figura 2- Relacionamento Doador-Donativo

**Relacionamento:** Doador oferece Donativo

**Descrição:** Com o intuito de se poder oferecer donativos provenientes de um doador, é importante armazenar a informação desta forma para, aquando de uma nova contribuição à empresa, ao receber novos donativos, se poder aceder diretamente ao doador registado no sistema e não ter de rever os donativos existentes para analisar os doadores já existentes.

**Multiplicidade:** Doador (1,1) - Donativo (1,n)

Um doador pode oferecer um ou mais donativos e um donativo tem obrigatoriamente de ter um Doador.

### • Relacionamento Donativo – Comprovativo

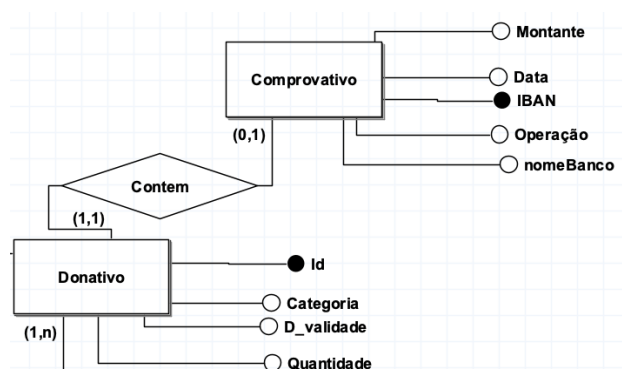


Figura 3- Relacionamento Donativo-Comprovativo

**Relacionamento:** Donativo contem Comprovativo

**Descrição:** Com o intuito de se poder guardar informação adicional dos donativos provenientes de um doador, é importante armazenar a informação do comprovativo dos mesmos aquando de uma nova contribuição à empresa monetariamente para que seja possível através de um comprovativo se poder aceder diretamente ao/s donativo/s monetário/s registado/s no sistema e não ter de procurar os mesmos um a um.

**Multiplicidade:** Donativo (1,1) - Comprovativo (0,1)

Um donativo pode de ter ou não um comprovativo e um comprovativo tem que corresponder a um só donativo

### • Relacionamento Funcionário - Donativo

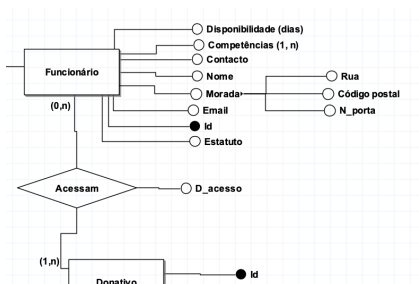


Figura 4- Relacionamento Funcionário-Donativo

**Relacionamento:** Funcionário acede a Donativo

**Descrição:** Com o intuito dos funcionários poderem ter acesso aos donativos feitos à associação, é importante que a informação seja guardada desta forma, para que, quando chegarem novos funcionários à associação, estes poderem aceder aos donativos na sua totalidade registados no sistema.

**Multiplicidade:** Funcionário (0,n) - Donativo (1,n)

Um Funcionário pode ou não aceder a um ou mais Donativos feitos à associação.

**Atributos:** No relacionamento “Acedem” entre Funcionário e Donativo existe o atributo “D\_acesso”, pois o funcionário acede a uma certa hora aos donativos, sendo assim possível registar essa informação.

### • Relacionamento Animal - Funcionário

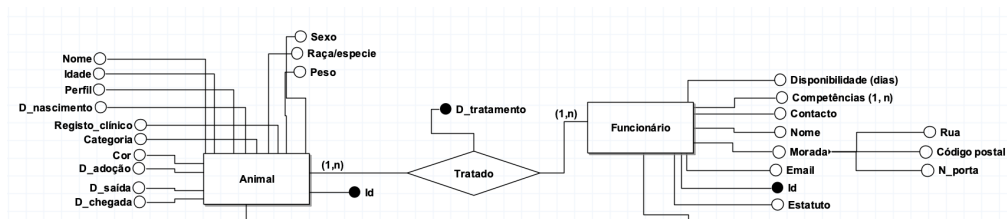


Figura 5- Relacionamento Animal–Funcionário

**Relacionamento:** Animal tratado Funcionário

**Descrição:** Com o intuito dos animais poderem ser tratados pelos funcionários, é importante que a informação seja guardada desta forma, para que, à medida que os animais são tratados pelos mesmos, os funcionários terem acesso à informação do animal.

**Multiplicidade:** Animal (1,n) - Funcionário (1,n)

Um ou mais animais são tratado por um ou mais Funcionários.

**Atributos:** No relacionamento “Tratado” entre Animal e Funcionário existe o atributo “D\_tratamento”, pois o animal é tratado numa certa data, sendo assim possível registar essa informação.

### • Relacionamento Animal – Adotante

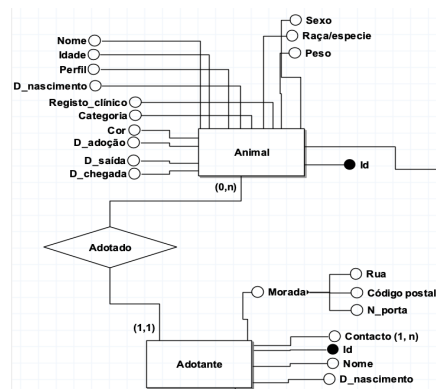


Figura 6- Relacionamento Animal–Adotante

**Relacionamento:** Animal adotado Adotante

**Descrição:** Com o intuito dos animais poderem ser adotados por adotantes, é importante que a informação seja guardada desta forma, para que, à medida que os animais são adotados, os donos terem acesso à informação do animal.

**Multiplicidade:** Animal (0,n) - Adotante (1,1)

Vários animais podem ser adotados por um adotante como também podem não ter nenhum dono.

### 3.4. Identificação e caracterização da associação dos atributos com as entidades e relacionamentos

ENTIDADE: Funcionário

Atributos	Tipo	Nulo	Composto	Multivalorado	Derivado	Candidato
Id	INT	Não	Não	Não	Não	Sim
Disponibilidade	VARCHAR (20)	Não	Não	Não	Não	Não
Competências	TEXT	Não	Não	Sim	Não	Não
Contacto	INT	Não	Não	Não	Não	Não
Nome	VARCHAR (40)	Não	Não	Não	Não	Não
Rua	VARCHAR (50)	Não	Não	Não	Não	Não
Código postal	INT	Não	Não	Não	Não	Não
Nº porta	INT	Não	Não	Não	Não	Não
Email	VARCHAR (40)	Não	Não	Não	Não	Não
Estatuto	VARCHAR (30)	Não	Não	Não	Não	Não

Tabela 1- Caracterização dos atributos dos funcionários

ENTIDADE: Donativo

Atributos	Tipo	Nulo	Composto	Multivalorado	Derivado	Candidato
Id	INT	Não	Não	Não	Não	Sim
D_validade	DATE	Sim	Não	Não	Não	Não
Quantidade	DOUBLE	Não	Não	Sim	Não	Não
Categoria	VARCHAR (20)	Não	Não	Não	Não	Não

Tabela 2- Caracterização dos atributos dos donativos

ENTIDADE: Doador

Atributos	Tipo	Nulo	Composto	Multivalorado	Derivado	Candidato
Id	INT	Não	Não	Não	Não	Sim
Nome	VARCHAR (40)	Não	Não	Não	Não	Não
Contacto	INT	Não	Não	Sim	Não	Não



Email	VARCHAR (40)	Não	Não	Não	Não	Não
Data	Date	Não	Não	Não	Não	Não
Comprovativo	VARCHAR (45)	Sim	Não	SIM	Não	Não

Tabela 3- Caracterização dos atributos dos doadores

ENTIDADE: Adotante

Atributos	Tipo	Nulo	Composto	Multivalorado	Derivado	Candidato
Id	INT	Não	Não	Não	Não	Sim
Contacto	INT	Não	Não	Não	Não	Não
Nome	VARCHAR (40)	Não	Não	Não	Não	Não
Rua	VARCHAR (50)	Não	Não	Não	Não	Não
Código postal	INT	Não	Não	Não	Não	Não
Nº porta	INT	Não	Não	Não	Não	Não
D_Nascimento	DATE	Não	Não	Não	Não	Não
Sexo	CHAR (1)	Não	Não	Não	Não	Não

Tabela 4- Caracterização dos atributos dos adotantes

ENTIDADE: Animal

Atributos	Tipo	Nulo	Composto	Multivalorado	Derivado	Candidato
Id	INT	Não	Não	Não	Não	Sim
Nome	VARCHAR (40)	Sim	Não	Não	Não	Não
idade	INT	Sim	Não	Não	Não	Não
Perfil	VARCHAR (45)	Sim	Não	Não	Não	Não
D_nascimento	DATE	Sim	Não	Não	Não	Não
Registo clínico	TEXT	Sim	Sim	Não	Não	Não
Categoria	VARCHAR (20)	Não	Não	Não	Não	Não
Cor	VARCHAR (20)	Não	Não	Não	Não	Não
D_adoção	DATE	Sim	Não	Não	Não	Não
D_saída	DATE	Sim	Não	Não	Não	Não
D_chegada	DATE	Não	Não	Não	Não	Não
Sexo	CHAR (1)	Não	Não	Não	Não	Não
Raça/Espécie	VARCHAR (20)	Não	Não	Não	Não	Não
Peso	INT	Não	Não	Não	Não	Não

Tabela 5- Caracterização dos atributos dos animais

ENTIDADE: Comprovativo

Atributos	Tipo	Nulo	Composto	Multivalorado	Derivado	Candidato
Id	INT	Não	Não	Não	Não	Sim
Data	DATE	Sim	Não	Não	Não	Não

IBAN	VARCHAR (40)	Sim	Não	Não	Não	Não
Montante	DOUBLE	Sim	Não	Não	Não	Não
NomeBanco	VARCHAR (45)	Sim	Não	Não	Não	Não
Tipo	VARCHAR (20)	Sim	Não	Não	Não	Não

Tabela 6- Caracterização dos atributos dos comprovativos

### 3.5. Detalhe ou generalização das entidades

Embora algumas das nossas entidades tenham atributos do mesmo tipo, não achamos que seja essencial criar uma superclasse para herdar esses atributos. Por exemplo, as entidades Funcionário, Doador e Adotante compartilham os atributos de morada e contato, no entanto, não consideramos crucial criar uma superclasse que inclua todos esses atributos, embora seja possível e compreensível fazê-lo.

## 3.6. Apresentação e explicação do diagrama ER

Depois de uma explicação minuciosa de cada entidade presente no sistema, bem como dos seus relacionamentos, incluindo os respectivos atributos, apresentamos o seguinte modelo conceitual de forma abrangente.

Com uma compreensão mais clara do modelo que desejamos implementar em nossa base de dados de uma associação de acolhimento animal, podemos agora prosseguir para a elaboração de uma nova modelagem e, depois de ter o modelo validade, avançar para a sua implementação física.

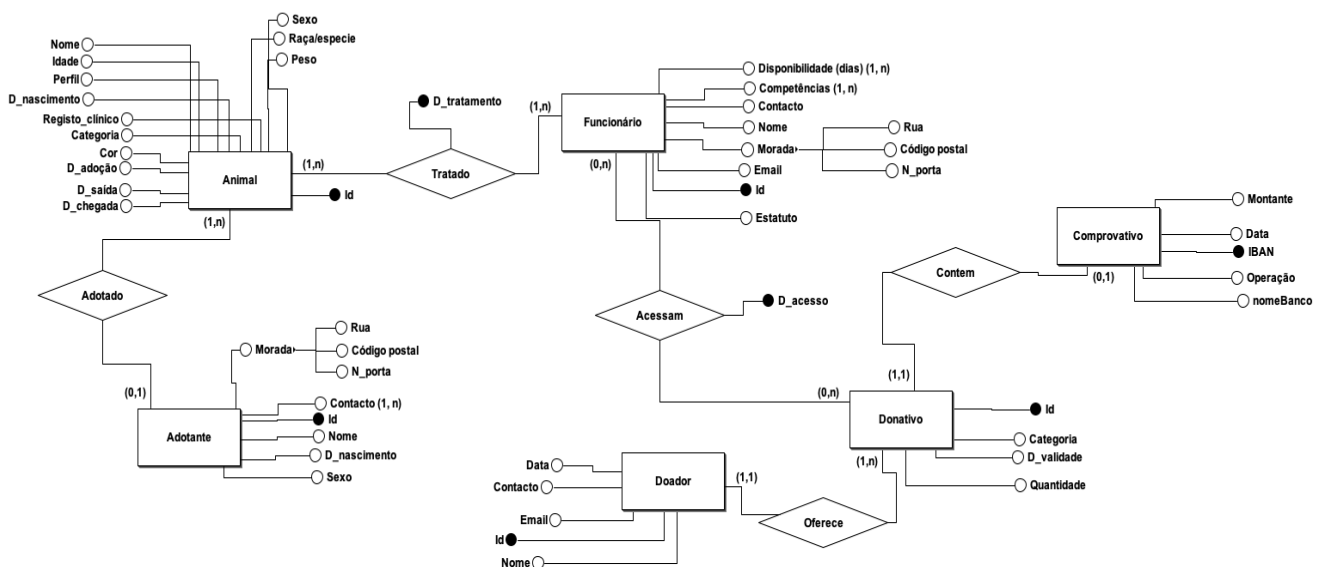


Figura 7- Modelo Conceptual

## 3.7. Validação do modelo de dados produzido

Como forma de avaliar o nosso modelo conceitual para o poder titular como modelo final, é necessário verificar se é possível responder aos requisitos previamente definidos.

1. Cada animal da associação deverá ser registrado através de um "Id" único.

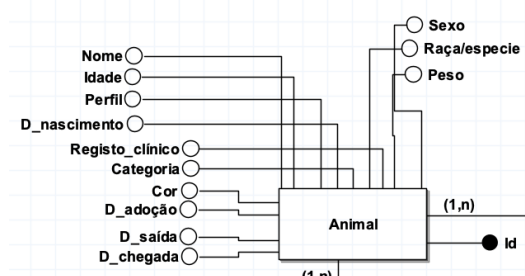


Figura 8- Entidade Animal

Todos os atributos mencionados nos requisitos estão associados à entidade.

2. Cada donativo feito à associação deve ser registado com um "id" único.

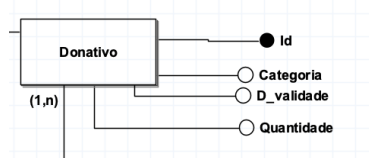


Figura 9- Entidade Donativo

Todos os atributos mencionados nos requisitos estão associados à entidade.

3. Cada entidade doadora deve ser registada com um "id".

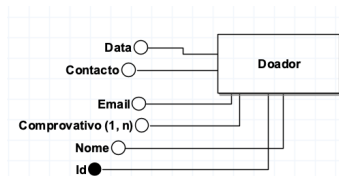


Figura 10- Entidade Doador

Todos os atributos mencionados nos requisitos estão associados à entidade.

4. Cada funcionário da associação deverá ser registado por um "id".

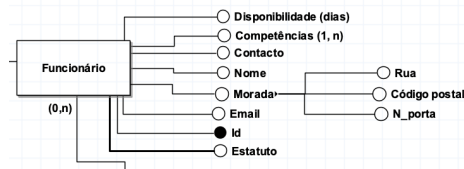


Figura 11- Entidade Funcionário

Todos os atributos mencionados nos requisitos estão associados à entidade.

5. Cada adotante deve ser registado com um "id" único.

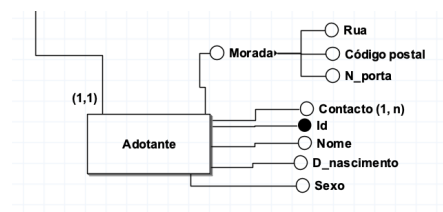


Figura 12- Entidade Adotante

Todos os atributos mencionados nos requisitos estão associados à entidade.

6. Os comprovativos devem conter informação relativa ao IBAN, o tipo de operação, o montante, o nome do banco e a data do movimento.

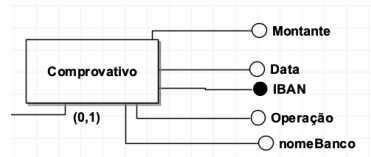


Figura 13- Entidade Comprovativo

7. Cada pessoa que adota um animal, apesar de poder fazê-lo mais vezes, só fica registada uma vez no sistema.
8. Um animal só pode ser adotado por um adotante.

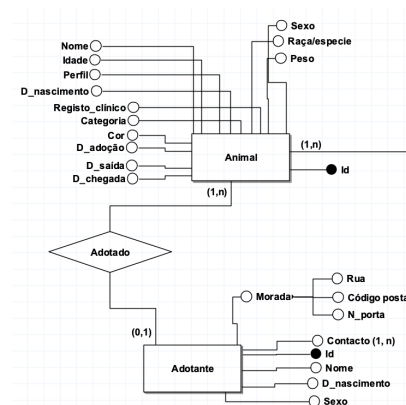


Figura 14- Relacionamento Animal-Adotante

Através da cardinalidade das relações entre as 2 entidades, cumpre-se os requisitos.

9. Os funcionários cuidam de um ou mais animais, assim como um animal pode receber tratamento de um ou mais funcionários, ficando registada a data do acesso ao animal.

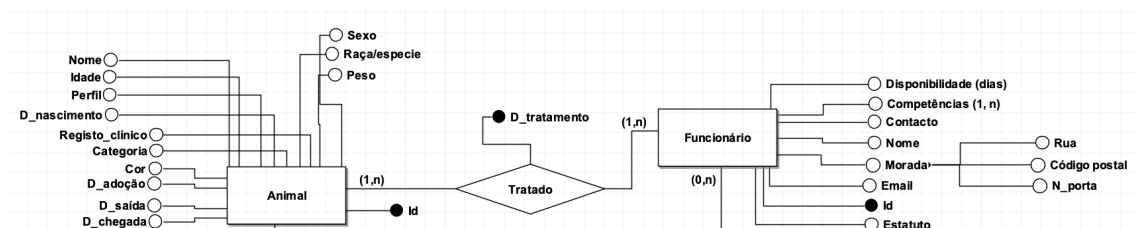


Figura 15- Relacionamento Animal-Funcionário

Através da cardinalidade das relações entre as 2 entidades, cumpre-se o requisito. A data que cada funcionário tratava cada animal é registada através de um atributo no relacionamento entre as duas entidades.

10. Os funcionários devem conseguir aceder a um ou mais donativos, sendo registada a data de acesso.

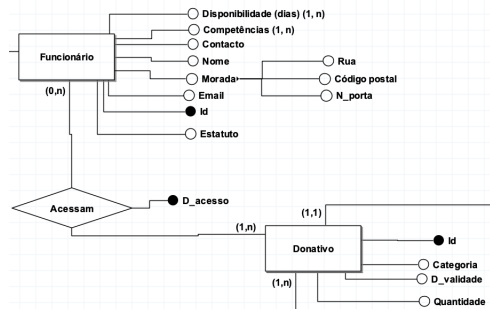


Figura 16- Relacionamento Funcionário-Donativo

Através da cardinalidade das relações entre as 2 entidades, cumpre-se os requisitos. O instante em que o funcionário acede aos donativos é registada através de um atributo no relacionamento entre as duas entidades. Os requisitos do modelo que mencionam como as entidades serão guardadas podem ser confirmadas através da figura respetiva.

Assim, uma vez que todos os requisitos são abrangidos pela nossa proposta de modelo conceptual, consideramos o mesmo como sendo o nosso modelo final, e podemos por isso, avançar para a próxima fase.

## 4. Modelação lógica

### 4.1 Construção e validação do modelo de dados lógico

O passo seguinte ao desenvolvimento da Base de Dados é construir um modelo lógico baseado na sua contraparte conceptual. No Modelo Lógico apenas encontramos tabelas que correspondem às entidades do modelo anterior ou, em casos onde se justifique, também correspondem a certos relacionamentos. Nesta nova medida aparecem conceitos importantes na área de Bases de Dados, como por exemplo o conceito de “Chave-Primária” ou “Chave-Estrangeira”, dos quais demonstram ser especialmente relevantes para esta fase do projeto.

Uma vez feita a conversão, obtivemos as seguintes tabelas:

1. Adotante:
  - 1.1. Chave Primária: Id
  - 1.2. Atributos: Id: INT, Nome: VARCHAR(40), D\_nascimento (data de nascimento):DATE, Sexo: CHAR(1), N\_Porta (Número da Porta): INT, Rua: VARCHAR(50), Código Postal: INT
  - 1.3. Chave Estrangeira: Não existente
  - 1.4. Atributos Não Obrigatórios: Nenhum
2. Funcionário:
  - 2.1. Chave Primária: Id
  - 2.2. Atributos: Id: INT, Disponibilidade: VARCHAR(50), Nome: VARCHAR(45), Estatuto: VARCHAR(30), Email: CHAR(40), N\_Porta (Número da Porta): INT, Rua: VARCHAR(45), Código Postal: INT, Número: INT
  - 2.3. Chave Estrangeira: Não existente
  - 2.4. Atributos Não Obrigatórios: Nenhum
3. Doador:
  - 3.1. Chave Primária: Id
  - 3.2. Atributos: Id: INT, Email: VARCHAR(40), Número(Contacto): INT, Nome: VARCHAR(40).
  - 3.3. Chave Estrangeira: Não existente
  - 3.4. Atributos Não Obrigatórios: Nenhum
4. Donativo:
  - 4.1. Chave Primária: Id e Doador\_Id (Id do doador)
  - 4.2. Atributos: Id: INT, Categoria: VARCHAR(20),D\_validade: DATETIME, Quantidade: DOUBLE

4.3. Chave Estrangeira: Doador\_Id

4.4. Atributos Não Obrigatórios :D\_validade e Quantidade.

5. Comprovativo:

5.1. Chave Primária: IBAN e Donativo\_Id (Id do donativo)

5.2. Atributos: IBAN: VARCHAR(45), Operação: VARCHAR(45), Montante, Quantidade: DOUBLE, Nome Banco: VARCHAR(60), Data Movimento: DATETIME

5.3. Chave Estrangeira: Donativo\_Id

5.4. Atributos Não Obrigatórios: Nenhum

6. Animal:

6.1. Chave Primária: Id

6.2. Atributos: Nome: VARCHAR(40), Idade: INT , Perfil: VARCHAR(45), D\_nascimento: DATE, Registo\_clinico :TEXT, Categoria VARCHAR(20), Cor: VARCHAR(20), D\_adoção: DATETIME, D\_saida: DATETIME, D\_chegada: DATETIME, Sexo CHAR(1), Raça e espécie: VARCHAR(20), Peso: INT, Adotante\_Id : INT

6.3. Chave Estrangeira: Adotante\_Id (Id do adotante)

6.4. Atributos Não Obrigatórios: Nome, Idade, Perfil, D\_nascimento, Registo\_clínico, D\_adoção, D\_saída, Adotande\_Id

De forma a gerar um modelo lógico coerente com o modelo conceptual, é necessário prestar atenção a atributos multi-valorados, pois, por vezes, é benéfico criar uma tabela separada da tabela principal. Assim, obtemos as seguintes:

7. Contacto:

7.1. Chave Primária: Número, Adotante\_Id (Id do adotante)

7.2. Atributos: Número: INT, Adotante\_Id: INT

7.3. Chave Estrangeira: Adotante\_Id

7.4. Atributos Não Obrigatórios: Nenhum

8. Competência:

8.1. Chave Primária: Aptidão, Funcionário\_Id (Id do funcionário)

8.2. Atributos: Aptidão: VARCHAR(40), Funcionário\_Id: INT

8.3. Chave Estrangeira: Funcionário\_Id

8.4. Atributos Não Obrigatórios: Nenhum

9. Disponibilidade:

9.1. Chave Primária: DiaDaSemana, Funcionário\_Id (Id do funcionário)



- 9.2. Atributos: DiaDaSemana: ENUM('Segunda-feira', 'Terça-feira', 'Quarta-feira', 'Quinta-feira', 'Sexta-feira', 'Sábado', 'Domingo'), Funcionário\_Id: INT
- 9.3. Chave Estrangeira: Funcionário\_Id
- 9.4. Atributos Não Obrigatórios: Nenhum

Como referido anteriormente, em certos casos, relacionamentos devem ser convertidos na sua própria tabela. Visto que no nosso modelo conceptual existem dois relacionamentos em que a cardinalidade corresponde a “n:m”, estes terão uma nova forma adequada ao modelo lógico do projeto. Desta forma, obtivemos as seguintes tabelas:

10. TB\_AnimalFuncionário:

- 10.1. Chave Primária: D\_tratamento, Funcionário\_Id (Id do funcionário), Animal\_Id (Id do animal)
- 10.2. Atributos: D\_tratamento: DATETIME, Funcionário\_Id: INT, Animal\_Id: INT
- 10.3. Chave Estrangeira: Funcionário\_Id, Animal\_Id
- 10.4. Atributos Não Obrigatórios: Nenhum

11. TB\_FuncionárioDonativo:

- 11.1. Chave Primária: D\_acesso, Funcionário\_Id (Id do funcionário), Donativo\_Id (Id do donativo)
- 11.2. Atributos: D\_acesso: DATETIME, Funcionário\_Id: INT, Donativo\_Id: INT
- 11.3. Chave Estrangeira: Donativo\_Id, Animal\_Id
- 11.4. Atributos Não Obrigatórios: Nenhum

## 4.2 Normalização de Dados

Com a construção de o modelo lógico, caso não houvesse uma fundamentação apropriada do modelo conceptual, pode colmatar em diversos defeitos relacionados com a eficácia do projeto. De modo a confirmar a autenticidade e a precisão do modelo, iremos recorrer às Formas Normais, que consistem num grupo de normas que ambicionam reduzir a redundância dos dados, a utilização de valores nulos, perdas de informação e certificar a atomização dos atributos.

Para podermos afirmar que uma Base de Dados está normalizada temos que, obrigatoriamente, garantir que todas as tabelas estão normalizadas, o que, por sua vez, implica a não repetição de grupos de dados e a não existência de anomalias.

A **Primeira Fórmula Normal** exige os seguintes pontos:

1. Cada tabela têm uma chave primária
2. Os valores de todos os atributos são atômicos
3. Não podem existir grupos de dados repetidos

Analisando o modelo lógico descrito, podemos facilmente ver que existe chave primária em todas as tabelas inclusive as que provieram de relacionamentos. Relativamente ao segundo ponto, rapidamente podemos confirmar que grande parte dos requisitos é atômico, pois muitos são representados com valores numéricos (INT e DOUBLE), apenas um carácter (CHAR(1)) e também por datas (DATETIME), o que indicia a representação de um atributo com apenas um valor. Desta forma, restam apenas os valores representados por mais do que um carácter. No entanto, verificando individualmente os restantes dos atributos, averiguamos que todos estes representam apenas valores únicos e não valores multivalorados. Alguns exemplos serão atributos como o “Nome”, “Rua” e “Email”, que representam apenas um de cada. Por fim, observando as tabelas conseguimos também apurar que não há grupos de dados repetidos. É importante esclarecer que no caso das tabelas que representam relações de “N:M” pode aparecer repetições das chaves estrangeiras, o que não indicia o desrespeito da regra fundamental, mas sim demonstra que as fragmentações dos dados em tabelas diferentes estão corretas.

De seguida, iremos validar a **Segunda Fórmula Normal** que, sucintamente, espera que todos os atributos não pertencentes à chave primária sejam dependetes da mesma e que a Primeira Fórmula Normal esteja validada. Observando os diferentes atributos não primos nas diversas tabelas, fica claro que a Segunda Fórmula Normal está devidamente aplicada. Um exemplo de aplicação seria a relação que os atributos da tabela “Adotante” possuem com o “Id” da mesma. Tanto o “Nome” como a “Rua” demonstram dependência funcional do “Id”, pois existe apenas um “Nome” e uma “Rua” por cada “Id” inserido na Base de Dados.

Finalmente, no caso de atender às duas fórmulas anteriores, podemos validar a **Terceira Fórmula Normal**. A aplicação desta fórmula obriga a que todos os atributos não primos sejam mutuamente independentes, não havendo assim dependências funcionais transitivas. Se verificarmos nas tabelas do modelo, podemos afirmar que a Terceira Fórmula Normal está em prática. No nosso modelo, em cada tabela apenas há dependências funcionais relacionadas com a respetiva chave primária, o que indicia a aplicação das duas últimas Fórmulas Normais.

Assim, podemos afirmar que o modelo lógico desenvolvido está normalizado.

## 4.3 Apresentação e explicação do modelo lógico produzido

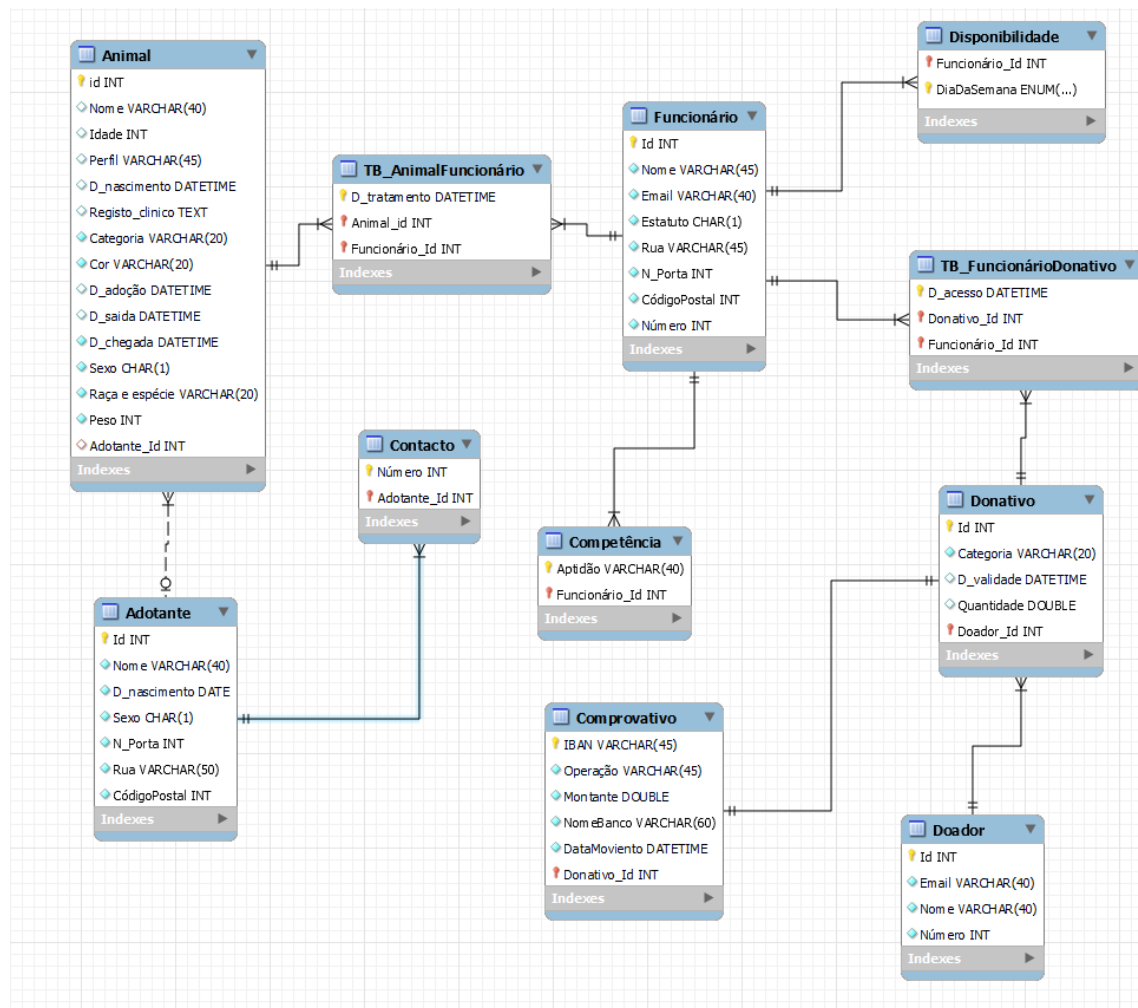


Figura 17- Modelo Lógico

Com base nos pontos 4.1 e 4.2, surge o modelo representado na figura 17 através da ferramenta “MySQL Workbench”. Para além das noções referidas anteriormente no ponto 4.1 que, essencialmente, descreve as entidades do modelo, é importante referir algumas componentes como: as relações entre as entidades, a escolha das chaves estrangeiras e algumas restrições dos atributos.

### A. As relações entre as entidades e escolha das chaves estrangeiras

Geralmente, uma conversão de um modelo conceptual para um modelo lógico proporciona mudanças nos relacionamentos e, por vezes, gera novas entidades sob a forma de tabela. Com isto, o nosso caso, ficou com as seguintes relações:

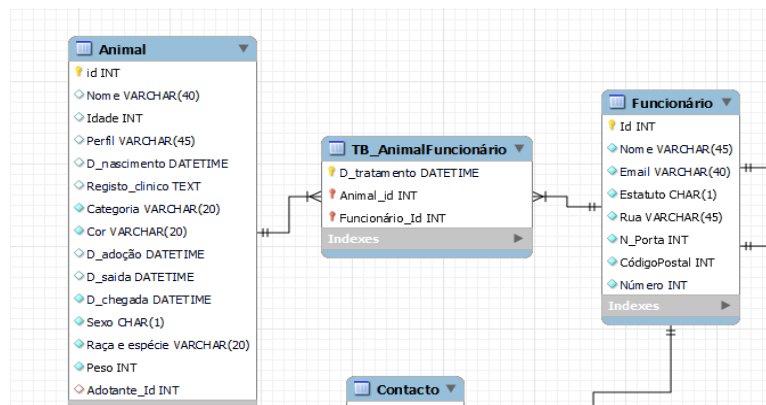


Figura 18- Relacionamento entre as tabelas Animal, TB\_AnimalFuncionário e Funcionário

A figura 18 representa o relacionamento “Animal Funcionário” do modelo conceptual que, como ilustra a figura, foi desdobrado em três tabelas. A relação entre as tabelas corresponde a “1:N” entre o Animal e a “TB\_AnimalFuncionário” e, de forma idêntica, surge um relacionamento “1:N” entre o Funcionário e a “TB\_AnimalFuncionário”. As chaves estrangeiras aparecem na tabela “TB\_AnimalFuncionário” e correspondem às chaves primárias das duas outras tabelas.

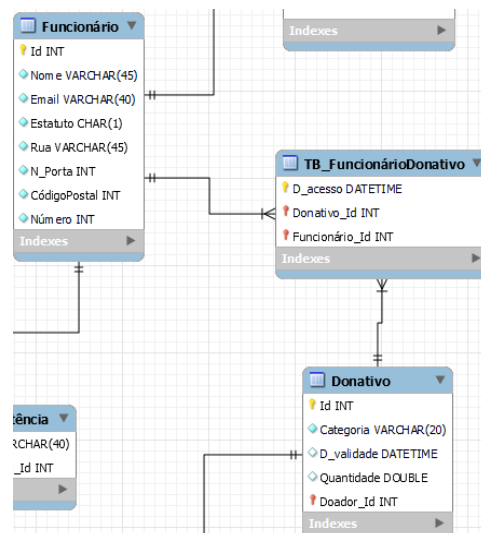


Figura 19- Relacionamento entre as tabelas Funcionário, TB\_FuncionárioDonativo e Donativo

A figura 19 representa o relacionamento “Funcionário Donativo” do modelo conceptual que possui um comportamento ao caso anterior. A relação entre as tabelas corresponde a “1:N” entre o Funcionário e a “TB\_FuncionárioDonativo” e, de forma idêntica, surge um relacionamento “1:N” entre o Donativo

e a “TB\_FuncionárioDonativo”. As chaves estrangeiras aparecem na tabela “TB\_FuncionárioDonativo” e correspondem às chaves primárias das duas outras tabelas.

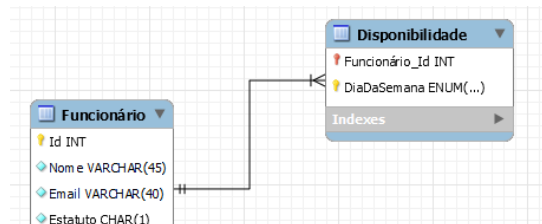


Figura 20- Relacionamento entre as tabelas  
Funcionário e Disponibilidade

A figura 20 representa o relacionamento “Funcionário Disponibilidade”. O seguinte caso, à semelhança de outros deste modelo, são gerados devido à existência de atributos multivalorados e há necessidade de gerar uma nova tabela por questões de normalização de dados. Deste modo, aparece o relacionamento “1:N” entre a tabela Funcionário e a tabela Disponibilidade. A chave estrangeira, neste relacionamento, aparece na tabela Disponibilidade e corresponde à chave primária da tabela Funcionário

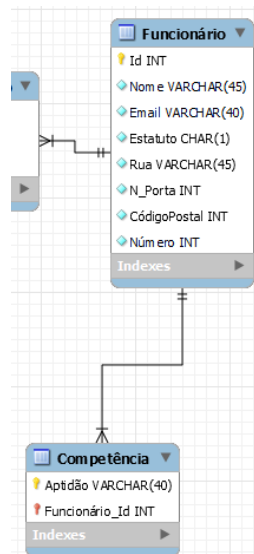


Figura 21- Relacionamento entre as tabelas  
Funcionário e Competências

A figura 21 representa o relacionamento “Funcionário Competência”. O seguinte caso, à semelhança do anterior, é novo no modelo lógico. O que está ilustrado, é o relacionamento “1:N” entre a tabela Funcionário e a tabela Competência. A chave estrangeira, neste relacionamento, aparece na tabela Competência e corresponde à chave primária da tabela Funcionário.

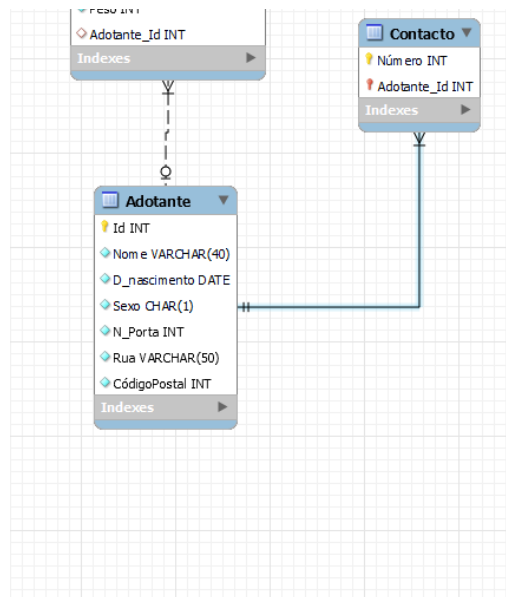


Figura 22- Relacionamento entre as tabelas Adotante e Contacto

A figura 22 representa o relacionamento “Adotante Contacto”. Este é último caso de tabela e relacionamento gerado por atributos multivalorados. O que está representado, é o relacionamento “1:N” entre a tabela Adotante e a tabela Contacto.

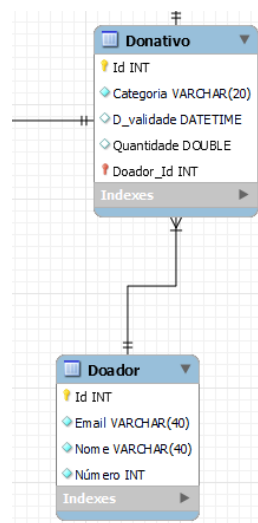


Figura 23- Relacionamento entre as tabelas Donativo e Doador

A figura 23 representa o relacionamento “Donativo Doador”. Em semelhança ao modelo conceptual, existe uma relação “1:N” entre o Doador e o Donativo. Relativamente às chaves estrangeiras, esta surge na tabela Donativo porque um Doador pode fazer mais do que um donativo.

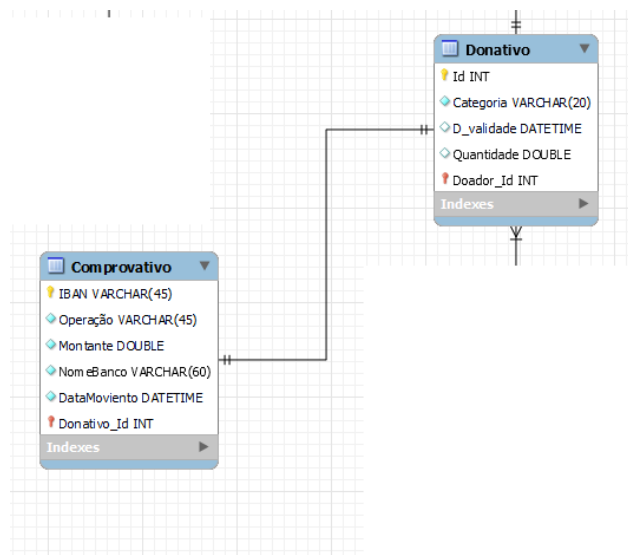


Figura 24- Relacionamento entre as tabelas  
Donativo e Comprovativo

A figura 24 representa o relacionamento “Donativo Comprovativo”. Em semelhança ao modelo conceptual, existe uma relação “1:1” entre o Doador e o Donativo. Relativamente às chaves estrangeiras, esta aparece na tabela Comprovativo, pois um Donativo pode não ter comprovativo associado.

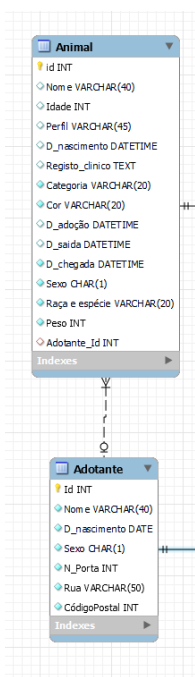


Figura 25- Relacionamento entre as tabelas  
Animal e Adotante

A figura 25 representa o relacionamento “Adotante Animal”. Este caso é único no modelo, pois, apesar de existir uma relação “1:N” entre o Adotante e o Animal, um Animal não têm que

estar obrigatoriamente associado a um Adotante. A nível lógico, isto faz com que a chave estrangeira possa ter o valor “null” ao contrário de todas as outras no modelo.

## B. As restrições dos Atributos

De modo a otimizar e a controlar as inserções nas tabelas é preciso criar algumas restrições no que pode ou não ser atribuído a um campo. Os atributos que possuem restrição no nosso modelo são os seguintes:

Os **números** de telefone presentes nas tabelas Doador, Funcionário e Contacto são guardados como inteiros. Contudo, como um número de telefone precisa de ter 9 dígitos, é colocada uma restrição que obriga o valor a ter 9 dígitos.

Os valores **Montante** e **Quantidade** presentes nas tabelas Comprovativo e Donativo são também restritos, visto que não há montantes nem quantidades negativas.

O **Estatuto** dos Funcionários é restrito a duas opções: “V” para voluntário ou “P” para profissional.

O **Sexo** dos Animais é restrito a duas opções: “F” para feminino ou “M” para masculino.

Os **Códigos Postais** que surgem nas tabelas Adotante e Funcionário são guardados como inteiros. Deste modo, é necessário restringir o código postal a sete dígitos.

## 4.4 Validação do modelo com interrogações do utilizador

De modo a validar o modelo lógico concebido, iremos agora seleccionar as interrogações que consideramos mais interessantes e vamos verificar se cumprimos os requisitos e se, consequentemente, o modelo consegue satisfazer os mesmos.

### 1. Listar os donativos por intervalos de tempo de validade ordenados por prazo mais perto de expirar

Para este efeito, podemos seleccionar da tabela todos os donativos cujas datas se encontrem entre os parâmetros no intervalo, podendo ainda seleccionar só aqueles que tenham uma data de validade. Por fim ordenamos os donativos por ordem ascendente (datas de validade mais curtas surgem primeiro).



## **2. Obter o capital da associação**

Neste segundo caso, é possível realizar uma operação como uma junção à esquerda das tabelas Donativo e Comprovativo de modo a perceber que donativos foram feitos em dinheiro. De seguida, de modo a eliminar o que não nos importa, vamos fazer uma seleção onde o houve correspondência, deixando apenas as doações com comprovativo. Por fim, obtemos o somatório da “Quantidade”.

## **3. Visualizar se um adotante é também doador**

O fator que une os adotantes com os doadores são os números telefónicos. Deste modo, é possível juntar a entidade Contacto com Adotante através do valor em comum. De seguida, em semelhança à interrogação anterior, utilizamos uma junção à esquerda para unir a tabela resultante da junção com a tabela dos doadores, seguindo o mesmo fator anterior. Por fim, selecionamos apenas os casos que são relevantes para a interrogação através de uma seleção onde o houve correspondência.

## **4. Ordenar os doadores por número de donativos**

Na quarta interrogação apenas aparecem duas entidades: Doador e Donativo. O primeiro passo é realizar uma junção de maneira a associar os donativos aos respetivos doadores. Neste ponto, temos várias entradas onde o Doador aparece repetido (Uma entrada por cada doação feita), então a etapa seguinte é agrupar os donativos por o “Id” do doador com a contagem de donativos que ele fez. Por fim, resta apresentar a listagem por ordem decrescente da contagem.

Além de manipulação de dados, é importante também mencionar que, de modo a manter a consistência da Base de Dados, que seria importante desenvolver rotinas de inserção de dados que asseguram, de forma simples, que não haja inconsistências com os dados. Os seguintes pontos são exemplos de procedimentos que precisaram de uma rotina na sua implementação física:

### **1. Adoção de um Animal**

Para garantirmos que a entrada é feita corretamente, precisaremos de estar atentos a dois casos distintos: Caso seja a primeira adoção, ou caso já tenha feito adoções. Na primeira situação, será preciso criar uma ficha para o adotante, enquanto na segunda apenas será preciso corresponder o animal ao seu dono. Na prática, seria conveniente ter dois procedimentos diferentes, onde no primeiro criava-se a conta e fazia-se a correspondência entre o animal e o novo dono, e no segundo, apenas fazíamos a correspondência.

## **2. Aceder a um Donativo**

Controlar o acesso aos bens da associação é essencial para o funcionamento correto da mesma. Desta forma, em semelhança à adoção, é preciso criar uma rotina que facilite o processo de aceder a um donativo. Para isso, podemos utilizar uma rotina que faz a correspondência entre o donativo e o respetivo funcionário que o acedeu, utilizando os identificadores do funcionário e do donativo.

## **3. Funcionário trata/visita um animal**

Em semelhança ao segundo ponto, é importante conhecer quem tratou um determinado animal membro da associação. Para esse objetivo, é necessário conceber uma rotina que faça uma associação entre o animal tratado e o respetivo tratador. Assim, a associação fica um ambiente mais seguro para os animais residentes.

## 5. Implementação Física

### 5.1 Tradução do esquema lógico para o sistema de gestão de bases de dados escolhido em SQL

A etapa que sucede a modelação lógica é a implementação física em “MySQL”. Primeiramente, para agilizar o processo de conversão, utilizamos uma funcionalidade do “MySQL Workbench” que faz uma primeira conversão para um *script* em “MySql”. Esta abordagem gerou a maioria das estruturas, mas existem detalhes que precisaram de ser mudados de forma a manter a coerência com aquilo que têm vindo a ser feito. Os detalhes referidos correspondem a restrições que o “MySQL” não conseguia prever através do modelo lógico e também a valores que a conversão atribui automaticamente que não estão corretos. As alterações realizadas foram as seguintes:

As restrições são o que permite atributos como “Código Postal” e “Número” poder serem guardados sob a forma de números inteiros. A redução do que pode ou não aparecer numa determinada coluna é importante para evitar dados que não possam ser relacionados no seu devido contexto. Os casos onde houve necessidade de restringir a gama de valores a ser colocadas nas colunas são os seguintes:

- Atributos “Sexo” e “Estatuto”, pois utilizamos um caracter para guardar a informação pretendida. Desta forma, limitamos as opções visto que tanto para o “Sexo” como para o “Estatuto” apenas á duas opções, ou em contexto físico, apenas dois caracteres possíveis.
- O “Número” correspondente aos contactos de diversas entidades é limitado de forma a ter nove dígitos.
- O “Código Postal” que surge em duas tabelas distintas deve conter exatamente sete dígitos numéricos.

No caso das tabelas “Contacto”, “Disponibilidade” e “Competências”, umas que representam atributos multivalorados para as respetivas identidades, decidimos utilizar a opção “ON DELETE CASCADE”, uma *flag* que diz ao motor que, caso a entrada da tabela referenciada pela chave estrangeira seja apagada, então as respetivas entras de “Contacto”, no caso dos adotantes, de “Disponibilidade” e “Competências”, no caso dos funcionários, deveram também ser eliminadas.

## 5.2 Tradução das interrogações do utilizador para SQL (alguns exemplos)

De modo a cumprir o propósito da Base de Dados, é ainda relevante garantir que todas as interrogações dos utilizadores são traduzidas para operações de consulta sobre os Dados. Estas *queries* podem ficar guardadas na própria Base de Dados sob a forma de procedimentos que, por sua vez, permitirão uma forma fácil e versátil de executar o código, e ainda de ser utilizado por outros programas externos.

Tendo em conta as interrogações referidas no ponto 4.4, vamos expor a tradução para “MySQL” das interrogações anteriores.

1. **Listar os donativos por intervalos de tempo de validade ordenados por prazo mais perto de expirar**

```
DELIMITER $$
CREATE PROCEDURE listar_donativos(IN date1 DATE, IN date2 DATE)
BEGIN
  SELECT * FROM Donativo
  WHERE Donativo.D_validade BETWEEN date1 AND date2 or Donativo.D_validade AND Donativo.D_validade IS NOT NULL
  ORDER BY Donativo.D_validade ASC;
END $$
DELIMITER ;
```

Figura 26- Código “MySQL” da primeira interrogação

2. **Obter o capital da associação**

```
SELECT SUM(Donativo.Quantidade) AS Capital FROM Donativo
LEFT JOIN comprovativo ON comprovativo.Donativo_id = Donativo.id
WHERE comprovativo.Donativo_id IS NOT NULL;
```

Figura 27- Código “MySQL” da segunda interrogação

### 3. Visualizar se um adotante é também doador

```
SELECT DISTINCT Adotante.* from Contacto  
INNER JOIN Adotante ON Adotante.Id = Contacto.Adotante_Id  
LEFT JOIN doador ON Contacto.Número = doador.Número  
WHERE doador.Número IS NOT NULL;
```

Figura 28- Código “MySql” da terceira interrogação

### 4. Ordenar os doadores por número de donativos

```
SELECT Doador.*,count(Donativo.id) AS `Total Doador` FROM Donativo  
INNER JOIN Doador ON Donativo.Doador_Id = Doador.Id  
GROUP BY Doador.id  
ORDER BY `Total Doador` DESC;
```

Figura 29- Código “MySql” da quarta interrogação

## 5.3 Definição e caracterização das vistas de utilização em SQL (alguns exemplos)

Um ponto importante de uma Base de Dados é traduzir as necessidades dos utilizadores para código. Por vezes, é do interesse do utilizador poder visualizar certos dados que podem não estar na mesma tabela. Assim, surge a opção de criar vistas para satisfazer os utilizadores. Alguns exemplos de vistas da nossa Base de Dados são as seguintes:

### 1. Funcionários com as respetivas competências e disponibilidade

```
CREATE VIEW FuncionarioCompetenciasDisponibilidade AS  
SELECT funcionario.Nome, GROUP_CONCAT(DISTINCT Competencia.Aptidão) AS comp_combinadas, GROUP_CONCAT(DISTINCT disponibilidade.DiaDaSemana) AS Disp_combinadas  
FROM mydb.funcionario  
INNER JOIN mydb.Competencia ON Funcionario.Id = Competencia.Funcionario_Id  
INNER JOIN mydb.disponibilidade ON Funcionario.Id = disponibilidade.Funcionario_Id  
GROUP BY mydb.funcionario.Nome;
```

Figura 30- Código “MySql” da primeira vista

A seguinte vista permite a quem a invoca ver os funcionários com as suas competências e disponibilidade. É especialmente útil no momento de dividir tarefas pelos voluntários, garantindo que não falta nada aos animais

## 2. Funcionários com as respetivas competências e disponibilidade

```
CREATE VIEW AdotanteContactos AS
SELECT adotante.*,GROUP_CONCAT(DISTINCT contacto.Número) AS Contactos from adotante
INNER JOIN mydb.contacto ON adotante.Id = contacto.Adotante_Id
GROUP BY adotante.id;
```

Figura 31- Código “MySQL” da segunda vista

Esta representação permite ver os adotantes com os respetivos números de telefone. Pode ser útil caso um dos contactos não esteja operacional.

## 3. Associação do comprovativo ao doador

```
CREATE VIEW DoadorEmDinheiro AS
SELECT comprovativo.IBAN,comprovativo.Operação,comprovativo.Montante,comprovativo.NomeBanco,comprovativo.DataMovimento,Doador.Nome from Comprovativo
INNER JOIN mydb.donativo ON comprovativo.Donativo_Id = Donativo.Id
INNER JOIN mydb.doador ON Donativo.Doador_Id = doador.id;
```

Figura 32- Código “MySQL” da terceira vista

Por último, esta vista facilita o processo de verificar quem foi o doador que fez o donativo em dinheiro. Aqui aparecem informações relativas ao método de pagamento.

## 5.4 Cálculo do espaço da Bases de Dados (inicial e taxa de crescimento anual)

O cálculo do espaço ocupado pela Base de Dados é relevante para questões como o desempenho do sistema, planeamento de *backup* e para a monitorização do uso de armazenamento. Tendo em conta o nosso caso, é importante estar atento a uma particularidade presente na tabela “Animal” e na tabela “Disponibilidade”. O registo clínico, presente na tabela “Animal” é do tipo “TEXT”, o que pode variar conforme com as entradas da tabela. O mesmo acontece na tabela “Disponibilidade” onde o atributo “DiaDaSemana” é do tipo “ENUM”. Desta

forma, iremos considerar para situações semelhantes um uso de “80” bytes de memória em ambos os casos.

Com base no conhecimento prévio sobre o tamanho ocupado por diferentes tipos de dados, como INT, CHAR, VARCHAR, DOUBLE e outros, realizamos o cálculo do espaço ocupado em disco por cada atributo de cada tabela.

Tabela	Atributos	Tamanho (em bytes)
<b>Adotante</b>	Id (INT), Nome (VARCHAR(40)), D_nascimento (DATE), Sexo (CHAR(1)), N_Porta (INT), Rua (VARCHAR(50)), Código Postal (INT)	$4 + 40 + 8 + 1 + 4 + 50 + 4 = 111$
<b>Funcionário</b>	Id (INT), Disponibilidade (VARCHAR(50)), Nome (VARCHAR(45)), Estatuto (VARCHAR(30)), Email (CHAR(40)), N_Porta (INT), Rua (VARCHAR(45)), Código Postal (INT), Número (INT)	$4 + 50 + 45 + 30 + 40 + 4 + 45 + 4 + 4 = 226$
<b>Doador</b>	Id (INT), Email (VARCHAR(40)), Número (INT), Nome (VARCHAR(40))	$4 + 40 + 4 + 40 = 88$
<b>Donativo</b>	Id (INT), Categoria (VARCHAR(20)), D_validade (DATETIME), Quantidade (DOUBLE)	$4 + 20 + 8 + 8 = 40$
<b>Comprovativo</b>	IBAN (VARCHAR(45)), Operação (VARCHAR(45)), Montante (Quantidade: DOUBLE), Nome Banco (VARCHAR(60)), Data Movimento (DATETIME)	$45 + 45 + 8 + 60 + 8 = 166$
<b>Animal</b>	Id (INT), Nome (VARCHAR(40)), Idade (INT), Perfil (VARCHAR(45)), D_nascimento (DATE), Registo_clinico (TEXT), Categoria (VARCHAR(20)), Cor (VARCHAR(20)), D_adoção (DATETIME), D_saída (DATETIME), D_chegada (DATETIME), Sexo (CHAR(1)), Raça e espécie (VARCHAR(20)), Peso (INT), Adotante_Id (INT)	$4 + 40 + 4 + 45 + 8 + 80 + 20 + 20 + 8 + 8 + 8 + 1 + 20 + 4 + 4 = 290$
<b>Contacto</b>	Número (INT), Adotante_Id (INT)	$4 + 4 = 8$
<b>Competência</b>	Aptidão (VARCHAR(40)), Funcionário_Id (INT)	$40 + 4 = 44$
<b>Disponibilidade</b>	DiaDaSemana (ENUM('Segunda-feira', 'Terça-feira', 'Quarta-feira', 'Quinta-feira', 'Sexta-feira', 'Sábado', 'Domingo')), Funcionário_Id (INT)	$80 + 4 + 4 = 88$
<b>TB_AnimalFuncionário</b>	D_tratamento(8), Funcionário_Id(4), Animal_Id(4)	$8 + 4 + 4 = 16$
<b>TB_FuncionárioDonativo</b>	D_acesso (8), Funcionário_Id (4), Donativo_Id (4)	$8 + 4 + 4 = 16$

Tabela 7- Tabela do tamanho utilizado pelas entidades na associação “Os bichinhos”

Desta forma, o tamanho total da nossa base de dados seria, sem povoamento, 1099 bytes. No entanto, para uma estimativa real, tendo em conta o povoamento do nosso modelo faria um total de:

$$3 * 111 + 226 * 4 + 6 * 88 + 11 * 40 + 1 * 166 + 290 * 8 + 8 * 8 + 7 * 44 + 5 * 88 + ((730 + 2 + 600) * 8 + 3) * 16 + 16 * 200 = 1745452 \text{ bytes}$$

É de notar que os valores obtidos são uma mera estimativa. Seguindo este modelo de utilização, no final do primeiro ano teríamos 1.707 kb de espaço em disco na nossa base de dados.

Segundo um estudo do Instituto da Conservação da Natureza e das Florestas registou se um aumento de 39,12% quando comparamos o ano de 2020 com o de 2021. Tendo em conta este crescimento à uma necessidade de reforçar o número de funcionários da associação e um consequente investimento para toda a estrutura da mesma, de forma a conseguir acolher mais animais e poder alocar mais recursos para os mesmos.

Ano	Nº de animais recolhidos
2018	35.733
2019	31 966
2020	31 339
2021	43603

Figura 33- Dados referentes aos animais recolhidos no período entre o ano de 2018 e 2021, a nível nacional.

## 5.5 Indexação do Sistema de Dados

Um índice em numa Base de Dados escrita em “MySQL” é uma estrutura auxiliar que melhora a eficiência das operações de pesquisa sobre os dados. Este é automaticamente criado em colunas que sejam chave primária, sendo designados de índices implícitos.

A função principal de um índice é acelerar a velocidade das operações de seleção (*SELECT*) e em comandos que exigem consulta (*WHERE*, *GROUP BY*), mas, em contrapartida, tornam processos que envolvam modificação dos dados, como por exemplo, as operações de inserção e de atualização de dados. Isto posto, a criação de um índice é uma decisão importante que pode tanto beneficiar o projeto, mas também pode prejudicar a performance da Base de Dados.

Uma vez que já possuímos as *queries* transcritas em “MySQL”, podemos averiguar em que casos é justificável uso de índice ou não. Nas *queries* definidas, conseguimos perceber que a maioria das operações de seleção e de consulta utilizam chaves primárias que já possuem por defeito índices. Contudo, encontramos três casos que, baseados nas *queries* justificam o uso de indices. Os casos são os seguintes:

```
CREATE INDEX dataChegada ON Animal (D_chegada);
CREATE INDEX fk ON Animal (Adotante_Id);
CREATE INDEX dataValidade ON Donativo (D_validade);
```

Figura 34- Código “MySQL” dos índices

O primeiro índice incide na coluna “D\_chegada” da tabela “Animal”, pois há um procedimento que consulta a coluna.

O segundo índice incide na coluna “Adotante\_Id”, chave estrangeira da tabela “Animal”, porque é consultada numa *query* que consulta animais sem dono.



O terceiro e último índice, está presente na coluna “D\_validade” da tabela Donativo, como nos outros casos, surge em duas *queries* como a primeira apresentada no ponto 5.2.

É também importante referir que nenhuma destas colunas representa uma chave primária, visto que seria redundante a criação de um índice, e também que todas estas tabelas são esperadas, relativamente ao projeto, ter um número grande de entradas.

## 5.7 Plano de segurança e recuperação de dados

Um plano de segurança de dados é uma estratégia essencial ao funcionamento de uma Base de Dados. Para além de prevenir acessos indevidos a conteúdos sensíveis, pode também facilitar o desempenho das funções de cada utilizador, na medida em que utilizaram uma interface mais indicada para as suas funções. Deste modo, foram criados perfis de utilizadores que poderão utilizar a Base de Dados.

Primeiramente, foram criadas contas sem nenhum tipo de restrições. Estas contas, serão utilizadas pelos os administradores, mais concretamente, pelo João e o Rodrigo.

Estas contas não têm nenhuma restrição sobre o uso da Base de Dados, podendo invocar qualquer funcionalidade da mesma, ainda que fuja um pouco das suas responsabilidades dentro da associação.

```
1 CREATE USER 'rodrigo'@'localhost' IDENTIFIED BY 'rod123';
2 GRANT ALL PRIVILEGES ON *.* TO 'rodrigo'@'localhost';
3
4 CREATE USER 'joao'@'localhost' IDENTIFIED BY 'joao123';
5 GRANT ALL PRIVILEGES ON *.* TO 'joao'@'localhost' ;
```

Figura 35- Contas dos Administradores

As restantes contas são iguais no que toca a responsabilidades na Base de Dados. Estas estão destinadas a ser usadas pelos voluntários da associação. Desta forma, os usuários terão um acesso limitado à Base de Dados, podendo apenas usar procedimentos dentro do seu estatuto da associação.

Na seguinte figura podemos observar duas contas, onde a primeira é destinada a ser utilizada pelo “Duarte Montes”, um voluntário atual, e a segunda é um formato genérico para futuros voluntários da associação.

```
CREATE USER 'duarte1'@'localhost' IDENTIFIED BY 'du1';
GRANT EXECUTE ON PROCEDURE consumo_donativo TO 'duarte1'@'localhost';
GRANT EXECUTE ON PROCEDURE adicionar_contacto TO 'duarte1'@'localhost';
GRANT EXECUTE ON PROCEDURE FuncionárioAcedeDonativo TO 'duarte1'@'localhost';
GRANT EXECUTE ON PROCEDURE FuncionárioTrataAnimal TO 'duarte1'@'localhost';
GRANT EXECUTE ON PROCEDURE FuncionárioUsaDonativoEmAnimal TO 'duarte1'@'localhost';

CREATE USER 'voluntario2'@'localhost' IDENTIFIED BY 'vol2';
GRANT EXECUTE ON PROCEDURE consumo_donativo TO 'voluntario2'@'localhost';
GRANT EXECUTE ON PROCEDURE adicionar_contacto TO 'voluntario2'@'localhost';
GRANT EXECUTE ON PROCEDURE FuncionárioAcedeDonativo TO 'voluntario2'@'localhost';
GRANT EXECUTE ON PROCEDURE FuncionárioTrataAnimal TO 'voluntario2'@'localhost';
GRANT EXECUTE ON PROCEDURE FuncionárioUsaDonativoEmAnimal TO 'voluntario2'@'localhost';
```

Figura 36- Conta do voluntário "Duarte" e conta genérica

Um plano de recuperação de dados é também uma estratégia essencial ao funcionamento de uma Base de Dados. Este procedimento possibilita que o sistema faça uma cópia segurança do estado atual da Base de Dados que, caso seja necessário, possa ser recuperado. Para este efeito, desenvolvemos um programa simples em “python” que se conecta à Base de Dados e realiza uma cópia de segurança. Posteriormente, já no “Hardware” próprio, pode ser programado utilizando, por exemplo, usando o “Task scheduler” da Microsoft, para que o programa corra num intervalo de tempo pretendido

```
import subprocess
import datetime

host = 'localhost'
port = 3306
username = 'root'
password = 'nunonuno153'
database = 'mydb'

current_datetime = datetime.now().strftime('%Y%m%d_%H%M%S')
backup_file = f'backup_{current_datetime}.sql'

subprocess.run(['mysqldump', '-h', host, '-P', str(port), '-u', username,
               '-p' + password, database, '--result-file=' + backup_file])
```

Figura 37- Código que realiza Backup

Na figura supramencionada, encontramos um pequeno programa que cria um “backup” que posteriormente pode ser usado para carregar na Base de Dados.

Para facilitar o processo de carregamento dos “backups”, foi desenvolvido um programa em “python” para esse efeito.

```
import subprocess

username = 'root'
password = 'nunonuno153'
database = 'mydb'
backup_file = 'backup.sql' # Nome do ficheiro

command = f"mysql -u {username} -p{password} {database} < {backup_file}"
try:
    subprocess.run(command, shell=True, check=True)
    print("Backup file loaded successfully.")
except subprocess.CalledProcessError as e:
    print(f"Failed to load backup file: {e}")
```

Figura 38- Programa que carrega um Backup

Tanto o programa que salva o estado, como o que realiza o carregamento do estado foram concebidos em “python” por serem fáceis de adaptar caso seja possível, e também porque podem ser automatizados com outros programas. Os dois programas poderiam ser executados através de um terminal conectado à Base de Dados, mas poderia gerar confusões aos utilizadores que não estão acostumados a esse tipo de ambiente. Assim, foi fácil de tomar a decisão de usar um “script”, pois apenas é preciso alterar um pequeno campo em vez de escrever um comando completo e, também, porque é mais cómodo para os utilizadores.

## **6. Implementação do Sistema de Recolha de Dados**

### **6.1 Apresentação e modelo do sistema**

Uma Base de Dados é apenas útil se conter dados que possa guardar e manipular. Por esse motivo, é necessário encontrar um sistema que seja capaz de importar dados diretamente para a Base de Dados. Apesar de muitos dos dados serem inseridos no quotidiano da associação, é importante lembrar que já existem muitos dados atuais que precisam de ser inseridos, e também que é sempre vantajoso poder inserir dados em massa. Tendo em conta a forma antiga de guardar dados utilizada pela associação, o “Excel”, sabemos que grande parte da informação estará guardada em ficheiros desse tipo que, por sua vez, podem ser convertidos facilmente para “.csv”. Deste modo, desenvolvemos um programa que é capaz de alimentar a Base de Dados através de diversos ficheiros “.csv”. Para esse efeito, para ajudar na autenticidade da simulação, utilizamos também alguns dos procedimentos criados para o uso diário dos funcionários.

O programa foi desenvolvido na linguagem “python”, pois diversos interpretadores suportam o pacote “mysql” do qual podemos utilizar a funcionalidade “connector” para comunicar com a Base de Dados. Além disso, existe também o pacote “csv” que é bastante conveniente no que toca a percorrer os ficheiros em formato “.csv”.

Assim, pedimos ao João os ficheiros “Excel” que ele utilizava e, de forma a facilitar o processo de importação de dados, separamos em vários ficheiros “.csv”. Desta forma, caso queiram utilizar “Excel” para armazenar de forma temporária, podem seguir o padrão e agilizar o processo de alimentar a Base de Dados.

## 6.2 Implementação do sistema de recolha

Com os objetivos definidos, a etapa seguinte consiste em codificar o programa desejado. Deste modo, com diversas pesquisas averiguamos que a forma mais fácil seria, como dito anteriormente, utilizar os pacotes de “mysql” da linguagem python. Os interpretadores usados podem variar, mas um bom exemplo seria o “Anaconda”, que inclui todos os pacotes necessários para o programa.

Posto isto, realizamos um programa que é capaz de alimentar todas as tabelas, exceto as tabelas que visam guardar informações das interações entre funcionários e donativos e entre animais e funcionários. Esta decisão foi tomada, pois este tipo de interação deve ser registada na hora pelos funcionários, e também porque era algo complexo para o João gerir e não possuía dados anteriores.

Visto que existem várias tabelas e, como o código é relativamente repetitivo, iremos apenas expor o código que alimenta a tabela “Animal” e a tabela “Adotante”. A escolha de exemplos baseou-se na forma de como a inserção é feita. No caso dos animais, é utilizado o comando “INSERT” do “MySQL”, enquanto para o caso dos Adotantes usamos um procedimento. Para o programa ser executado, é exigido que as tabelas e os procedimentos já estejam criados.

```
def importDataAnimal(cursor):
    # Read data from CSV and insert into the database
    with open(csv_file_pathAnimal, 'r') as file:
        reader = csv.reader(file)

        for row in reader:
            # Check for empty values and replace them with NULL
            row = [value if value else None for value in row]

            # Adjust the column indexes as per your CSV file structure
            id = row[0]
            Nome = row[1]
            Idade = row[2]
            Perfil = row[3]
            D_nascimento = row[4]
            Registo_clinico = row[5]
            Categoria = row[6]
            Cor = row[7]
            D_adoção = row[8]
            D_saida = row[9]
            D_chegada = row[10]
            Sexo = row[11]
            Raça_espécie = row[12]
            Peso = row[13]
            Adotante_Id = row[14]

            # Insert data into the database
            insert_query = """
                INSERT INTO Animal (
                    id, Nome, Idade, Perfil, D_nascimento, Registo_clinico,
                    Categoria, Cor, D_adoção, D_saida, D_chegada, Sexo,
                    `Raça e espécie`, Peso, Adotante_Id
                )
                VALUES (%s, %s, %s, %s, %s, %s, %s, %s, %s, %s, %s, %s, %s, %s, %s)
            """
            values = (
                id, Nome, Idade, Perfil, D_nascimento, Registo_clinico,
                Categoria, Cor, D_adoção, D_saida, D_chegada, Sexo,
                Raça_espécie, Peso, Adotante_Id
            )
            cursor.execute(insert_query, values)
```

Figura 39- Inserção dos Animais em “csv”

De forma sucinta, o código presente na figura 39 começa por ler o abrir e iterar sobre o ficheiro “.csv”. Caso um valor esteja vazio numa determinada linha, o programa considera esse valor como nulo. De seguida, separa os dados pelas respetivas colunas e insere na Base de Dados.

```
def importDataAdocao(cursor):
    # Read data from CSV and determine the number of rows
    with open(csv_file_pathAdocao, 'r') as file:
        reader = csv.reader(file)

        for row in reader:
            row_count = len(row)
            if row_count == 2:
                # Invoke Procedure 1
                procedure_name = 'MaisQueUmaAdoção'
                # Unpack the rows as arguments for Procedure 1
                args = (row[0], row[1])

                # Call Procedure 1
                cursor.callproc(procedure_name, args)

            elif row_count == 9:
                # Invoke Procedure 2
                procedure_name = 'PrimeiraAdoção'
                args = (row[0], row[1], row[2], row[3], row[4],
                      row[5], row[6], row[7], row[8]) # Unpack the rows as arguments for Procedure 2

                # Call Procedure 2
                cursor.callproc(procedure_name, args)
```

Figura 40- Inserção dos Adotantes em “csv”

No que toca ao procedimento de leitura, os dois bocados de código são idênticos. Contudo, a inserção não é feita usando o “INSERT” do “MySQL”, mas sim um dos procedimentos criados. O programa analisa o número de campos que têm de ler e, desta forma, usa o procedimento mais adequado.

Antes de executar estes segmentos de código, é ainda feita a conexão à Base de Dados. Para isso, usamos o seguinte código:

```
try:
    # Connect to the database
    connection = mysql.connector.connect(**config)
    cursor = connection.cursor()

    # Import data from CSV
    importDataAnimal(cursor)
    importDataDoador(cursor)
    importDataDonativo(cursor)
    importDataComprovativo(cursor)
    importDataFuncionarios(cursor)
    importDataDisponibilidade(cursor)
    importDataCompetencias(cursor)
    importDataAdocao(cursor)
    # Commit the changes and close the connection
    connection.commit()
    cursor.close()
    connection.close()
    print('Data import completed.')

except mysql.connector.Error as error:
    print(f'Error: {error}')
```

Figura 41- Execução do Código

## 6.3 Funcionamento do sistema

O programa foi desenvolvido, pois seria mais rápido converter os registos anteriores da associação para a nova Base de Dados. Por isso, foi sempre tido em conta que a importação seria de ficheiros “.csv” gerados a partir de um ficheiro “Excel”. Desta maneira, a execução do programa será um passo posterior à criação das tabelas e dos procedimentos do sistema. De seguida, estamos livres de executar o programa. A execução pode tanto ser feita de um terminal como em um editor de texto, sendo apenas necessário existir os ficheiros “.csv” com o nome respetivo e na diretoria respetiva. Após estes passos, a Base já possui os dados e já podem ser feitas consultas. Caso ocorra algum erro na importação dos dados, estes são representados no terminal onde foi executado. Geralmente, os erros estão relacionados com dados mal convertidos para “.csv”, ficheiros sem estar nos locais corretos e por tentar colocar entradas repetidas.

## **7. Implementação do Sistema de Painéis de Análise**

### **7.1 Definição e caracterização da vista de dados para análise**

De modo a implementar o sistema de painéis de análise, como já havia sido referido, recorreremos à ferramenta da Microsoft “Power BI”, recomendada pelos docentes e, a nosso ver, uma excelente opção para alcançar a finalidade pretendida, permitindo-nos obter uma representação estruturada dos dados selecionados e extraídos da base de dados. Através da mesma, é nos possível selecionar, agregar e tratar os dados do modo que achemos mais relevante, findando com uma visualização chamativa, útil e interativa, caracterizada por uma exploração dinâmica e de fácil compreensão da informação, personalizada aos gostos dos funcionários da associação.

### **7.2 Povoamento das estruturas de dados para análise**

Para poder obter estatísticas à cerca da Base de Dados, é preciso existir povoamento em primeiro lugar. Para esse efeito, utilizamos o programa desenvolvido anteriormente para alimentar a Base de Dados com a informação atual da associação. Além disso, como queríamos que todas as tabelas contivessem dados, pedimos a um dos voluntários que nos informasse que ações normalmente acontecem no dia a dia da associação. Isto inclui, desde adoções, ao trabalho dos voluntários de, por exemplo, alimentar os animais. Seguidamente, criamos um ficheiro em “MySQL” que utilize os procedimentos criados tal como faria um utilizador da Base de Dados. Por fim, obtivemos então um sistema povoado com dados.



## 7.3 Apresentação e caracterização dos dashboards implementados



Figura 42-Dashboard

Com a ferramenta, como podemos observar na imagem supramencionada, encontramos cinco gráficos que achamos relevantes para o projeto. O primeiro, demonstra a percentagem de animais por categoria, concluindo que a associação, neste momento, possui mais cães do que qualquer outro tipo de animal doméstico. Dispomos também de uma tabela que mostra em data os animais foram adotados. Para facilitar as contas da associação, o “Power BI” é capaz de demonstrar o total dos donativos feitos em dinheiro, como é indicado na figura. Podemos também verificar que é fácil de descobrir quais as competências dos funcionários da associação. Por fim, podemos ver quais os donativos feitos à associação pelo seu tipo, sendo mais predominante comida para os animais do que mantas ou brinquedos.

## 8. Conclusão

Neste trabalho objetivou-se o ciclo de desenvolvimento de uma Base de Dados, sendo o tema escolhido pelos membros do grupo uma associação de animais. Todas as fases revelaram o seu papel na conceção do programa, sendo que os requisitos foram sempre a base para o sistema desenvolvido que, rapidamente, tomaram a forma de um desenho conceptual, modelo útil e minimalista que ilustra as ideias dos desenvolvedores. A modelação lógica, foi o ponto que deu uma forma mais técnica da Base de Dados, sendo depois implementada fisicamente em “MySQL”. Ainda tivemos a oportunidade de explorar algumas ferramentas que interagem com a Base de Dados e que integram o sistema desenvolvido, como por exemplo, um sistema de recolha de dados na linguagem “python”, e também podemos ver algumas estatísticas com a ferramenta “Power BI”.

O desenvolvimento do projeto foi acompanhando as aulas da unidade curricular, servindo como forma de consolidação da matéria. Além disso, serviu para perceber como os conteúdos estudados se interligavam e como podemos ultrapassar obstáculos dentro do ciclo de desenvolvimento.

Deste modo, podemos constatar o que seria uma possível implementação de um sistema de Bases de Dados real, pronto a ser utilizado por associações reais. Com o progresso da nossa educação em engenharia informática, podemos ainda explorar novas formas de melhorar o projeto desenvolvido, cobrindo partes como uma interface gráfica ou um outro tipo de aplicação que interage com o sistema desenvolvido.

Em suma, o trabalho desenvolvido serviu de teste e objeto de consolidação de estudo da unidade curricular Bases de Dados, incluindo todas as etapas previstas para os alunos e cumprindo com todos os objetivos esperados.

## 9.Referências Bibliográficas

ICNF - Instituto da Conservação da Natureza e das Florestas. Disponível em: <https://www.icnf.pt/api/file/doc/23b1dd8b4e8a6356>. Acesso em: 1 jun. 2023.

BLACKBOARD Learn. Disponível em: [https://elearning.uminho.pt/webapps/blackboard/content/listContent.jsp?course\\_id=\\_55895\\_1&content\\_id=\\_1301674\\_1&mode=reset](https://elearning.uminho.pt/webapps/blackboard/content/listContent.jsp?course_id=_55895_1&content_id=_1301674_1&mode=reset). Acesso em: 4 jun. 2023.

ICNF - Instituto da Conservação da Natureza e das Florestas. Disponível em: <https://www.icnf.pt/api/file/doc/23b1dd8b4e8a6356>. Acesso em: 15 jul. 2023.

CRIANDO Diagrama Entidade Relacionamento no brModelo. 11 nov. 2021. 1 vídeo (16 min 1 s). Publicado pelo canal Hugo Feitosa de Figueirêdo. Disponível em: <https://www.youtube.com/watch?v=rpw5bZutAZg>. Acesso em: 4 jul. 2023.

AULA 7 - Criando o Modelo Lógico no MySQL Workbench. 15 abr. 2020. 1 vídeo (20 min 15 s). Publicado pelo canal Carlos Portela. Disponível em: <https://www.youtube.com/watch?v=ILMsvnSKl5I>. Acesso em: 16 jul. 2023.

CURSO SQL Completo 2022 [Iniciantes] + Desafios + Muita Prática. 28 jul. 2022. 1 vídeo (230 min 7 s). Publicado pelo canal Dev Aprender. Disponível em: <https://www.youtube.com/watch?v=G7bMwefn8RQ>. Acesso em: 25 jul. 2023.