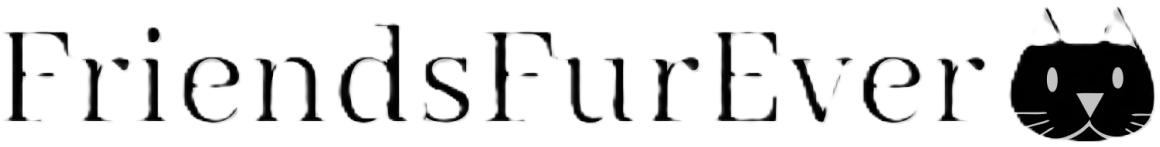
universidade de aveiro

Relatório de BD-FriendsFurEver



Gonçalo Lima- 108254 Francisco Murcela- 108815

Introdução

Para o nosso projeto de Base de Dados decidimos criar um sistema de serviços de adoção de animais, chamado FriendsFurEver.

Desenvolvemos uma webapp, com uma interface criada por nós, usando html, css, js, php, flask e python.

Na nossa aplicação, podemos registar-nos como empregados ou como adotantes, tendo roles diferentes e funcionalidades diferentes quando fazemos o login.

Na interface do adotante, podemos pesquisar por cães e gatos, filtrando pelas características pretendidas. Para adotarmos um animal, temos que preencher as nossas informações no perfil e depois clicamos na imagem do animal que pretendemos adotar.

Já na interface do empregado, quando fazemos login, encontramos uma homepage com as estatísticas do shelter. Podemos adicionar, remover ou atualizar informações sobre os animais e atualizar os seus tratamentos e vacinas.

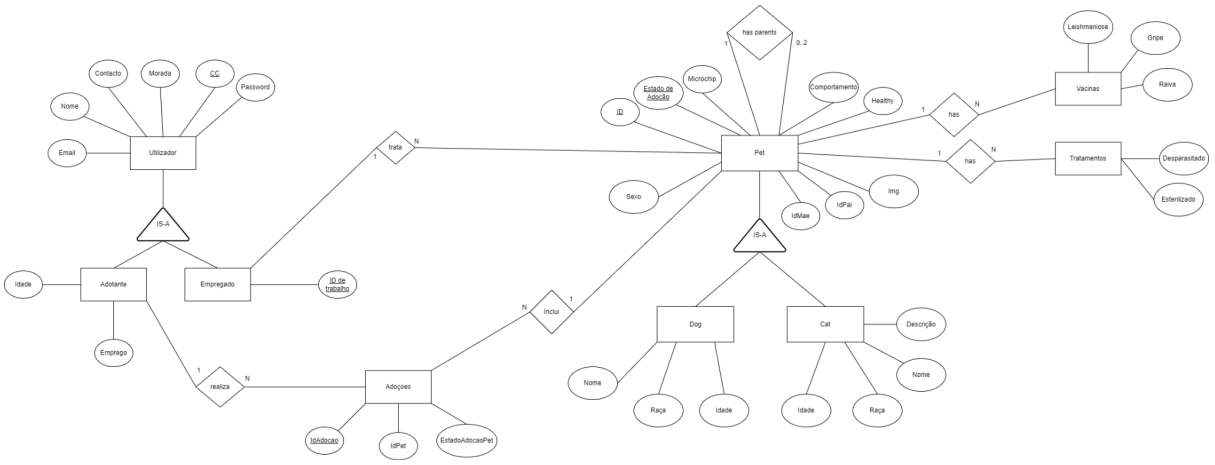
Ficheiros entregáveis

DLL.sql- ficheiro com o todo o sql implementado
Inserts.sql- ficheiro com os inserts implementados
Triggers.sql- ficheiro com os triggers implementados
Functions.sql- ficheiro com os UDF’s implementados
Procedures.sql- Ficheiro com os SP’s implementados
FriendsFurEver- pasta com a app

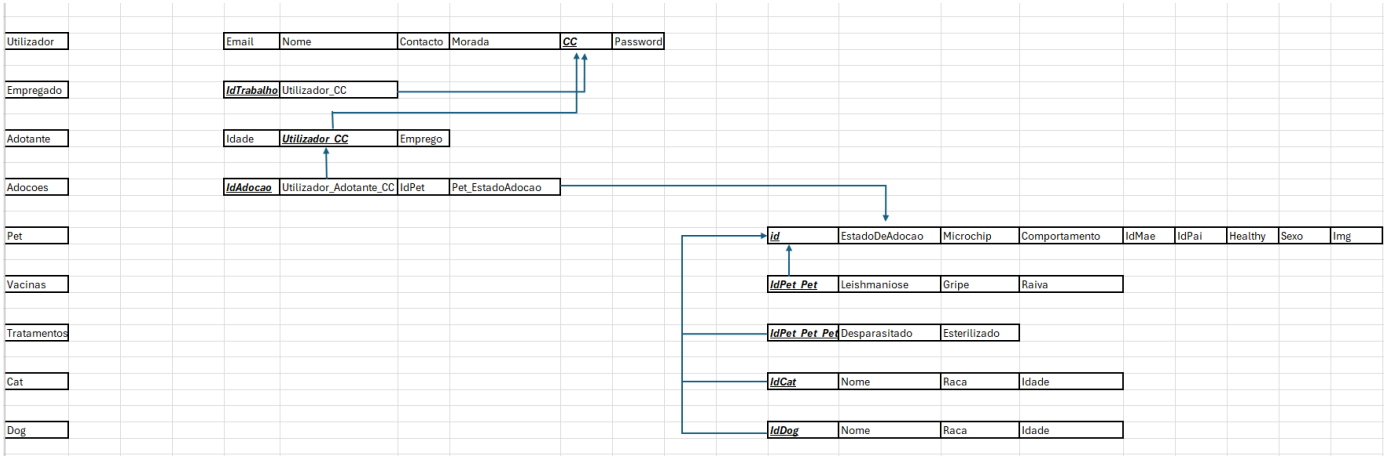
Gonçalo Lima- 108254 Francisco Murcela- 108815

Diagrama ER da Base de Dados

Tendo em conta a nossa primeira apresentação sobre o diagrama e esquema relacional, seguimos as sugestões do professor e alterámos ambos de modo a ficarem mais completos e de modo a representar melhor um sistema de adoção de animais.



Esquema Relacional



Gonçalo Lima- 108254 Francisco Murcela- 108815

UDF’s

Nos UDFS implementámos 5 funções importantes, que retornam as estatísticas atualizadas do shelter.

- Uma função que retorna o número total de animais no shelter.
- Outra que retorna a raça mais comum no shelter.
- Outra que retorna a percentagem de animais adotados no shelter.
- Outra que mostra o número total de animais adotados no shelter.
- E uma que mostra o número total de empregados no shelter.

Todas estas UDFs estão presentes na página inicial do empregado e são atualizadas conforme as alterações feitas.

Triggers

Nos triggers criamos:

- Um trigger para verificar se um animal ja foi adotado, quando o tentamos adotar.
- Um trigger para atualizar o estado de adoção de um animal, quando este é adotado.
- Um trigger para atualizar o estado de saúde de um animal, depois de lhe serem realizados os tratamentos necessários.
- Outro trigger para atualizar o estado de saúde de um animal, depois de lhe serem dadas as vacinas necessárias.
-

SP’s

Implementámos vários Stored Procedures, tais como:

- Um para autenticar um empregado
- Outro para autenticar um utilizador
- Outro para ir buscar os detalhes de um Animal(Pet)
- Outro para ir buscar os animais registados.
- Outro para ir buscar os detalhes do utilizador.
- Um que cria uma entrada na tabela de adoções quando um pet é adotado.

E muitos mais que vão constantemente atualizando as tabelas com detalhes novos, tanto dos pets, como das adoções e utilizadores(empregados e adotantes).

Segurança

Com a finalidade de assegurarmos uma maior segurança, utilizamos hashing nas passwords, de modo a serem encriptadas e um procedimento para validar as passwords.

Conclusões

Para finalizar, concluímos todos os objetivos pretendidos, incluindo os que nos propusemos a realizar como trabalho futuro na nossa apresentação, sendo estes a implementação de uma paginação na nossa página dos animais e uma barra de pesquisa na interface do empregado. Este trabalho permitiu-nos desenvolver as nossas capacidades de análise de sistemas e de desenvolvimento de bases de dados com um propósito de realizar um projeto que tenha uma aplicação na vida real, aprendendo mais sobre como funcionam este tipo de sistemas.