

Smart Pet Feeder – Segunda Fase

Projeto de Base de Dados – 2022/2023

Aluno 1:	202001449	Eduardo Silva			
Aluno 2:	202002147	Manuel Lagarto			
Aluno 3:	199100317	Rui Colaço			
Docente:	Prof. Cláudio Miguel Sapateiro				
Fase:	02	Data:	2022/2023	Turma:	LEEC-EC-02

1. Índice

2. Introdução.....	2
3. Especificação de Requisitos.....	3
3.1. Funcionalidades a Implementar no Projeto	3
3.1.1. Comedouro	3
3.1.2. App Móvel	3
3.1.3. Hub Central	3
3.2. Identificação das Entidades e Atributos	4
3.3. Relações entre Entidades (MER)	6
4. Diagrama Entidade-Relação (DER)	7
4.1. Proposta Inicial	7
4.2. Revisões ao DER	7
5. Modelo Relacional (MR).....	8
6. Scripts Desenvolvidos.....	8
6.1. Criação das Tabelas	8
6.2. Inserção de Dados	9
6.3. Inserção de Erros	9
6.4. Consulta das Tabelas	9
7. Conclusão.....	10
8. Anexos.....	11

2. Introdução

No âmbito da Unidade Curricular Projeto em Internet das Coisas, será desenvolvido um projeto cuja componente de base de dados será incluída na avaliação contínua da presente UC, Projeto de Base de Dados.

O projeto tem como objetivo o desenvolvimento de um comedouro inteligente para animais, denominado por *Smart Pet Feeder*.

Este comedouro realizará a gestão automática das quantidades de ração e água nas tigelas de alimentação com base nas parametrizações estabelecidas numa App Móvel, dadas pelo utilizador responsável pelo animal.

Existirá adicionalmente um Hub Central composto por um Raspberry Pi, o qual, será responsável pelo controlo de um ou mais comedouros a ele conectados, de acordo com os dados na base de dados interna e com a lógica programada para o processamento dos mesmos. Desta forma, o Hub Central será um dispositivo indispensável para a correta operação do comedouro.

3. Especificação de Requisitos

3.1. Funcionalidades a Implementar no Projeto

3.1.1. Comedouro

Cada comedouro será composto pelos sensores e atuadores responsáveis pela gestão das quantidades de ração e água nas tigelas. As quantidades serão verificadas através de sensores de nível por ultrassons que, após o sensor de presença ter detetado que o animal terminou a sua alimentação, estes reportam os níveis para o Hub Central. Existirão três sensores de nível para cada tigela da ração e água, e para o depósito interno da ração.

3.1.2. App Móvel

Através de uma App Móvel um ou mais utilizadores poderão gerir todos os comedouros conectados ao Hub Central, permitindo assim, definir um nome para cada comedouro, associar um animal ao mesmo, definir o local onde este se encontra instalado, monitorizar os níveis das tigelas e do depósito interno, parametrizar tabelas horárias para o enchimento da tigela de ração, visualizar um histórico da dieta do animal e receber alertas/notificações sobre eventos diversos, tais como, depósito interno quase vazio, falta de alimentação do animal, entre outros.

Cada perfil de utilizador criado na *app* será definido pelos dados de registo, assim como, níveis de acesso diferentes para a utilização de certas funcionalidades da *app*.

Será possível também criar um perfil para cada animal que será associado a um comedouro. Cada perfil será composto pelo nome do animal, peso, tipo (cão ou gato), idade e a dosagem de ração para os períodos de alimentação.

3.1.3. Hub Central

O Hub Central será responsável por estabelecer uma conexão entre cada comedouro e a App Móvel, assim como, por guardar, com recurso a uma base de dados interna, todos os dados descritos nos tópicos anteriores.

No Hub Central também será implementada a lógica de controlo responsável por fazer a leitura dos dados recebidos de cada comedouro, pelo processamento lógico dos dados, e por enviar os dados para os atuadores de cada comedouro.

3.2. Identificação das Entidades e Atributos

Foi dado início ao desenvolvimento do Modelo Entidade-Relação da base de dados do projeto com a identificação das seguintes entidades:

1. Utilizador – dados de login do utilizador;
2. Comedouro – dados de identificação do comedouro, animal atribuído e níveis de referência para as tigelas da água e da ração;
3. Animal – dados do perfil de cada animal associado a um comedouro.
4. Depósito – níveis de referência máximo e mínimo do depósito interno de ração do comedouro;
5. Horários – periodicidade e hora de cada refeição;
6. Histórico – dados com o *time stamp* da leitura do nível de cada sensor de cada comedouro;
7. Mensagens – mensagens modelo com nome, descrição e níveis de criticidade para cada notificação na *app*;

Para a entidade **Utilizador** foram definidos os seguintes atributos:

1. id – chave primária;
2. *nome* – nome do utilizador;
3. *password* – password de login do utilizador;
4. *perfil* – nível de controlo de acesso na *app*;
5. *telefone* – número de telemóvel/telefone do utilizador;
6. *email* – endereço de e-mail do utilizador;

Para a entidade **Comedouro** foram definidos os seguintes atributos:

1. id – chave primária;
2. *nome* – nome atribuído ao comedouro;
3. *local* – local da habitação onde se encontra o comedouro;
4. *nivelref_água* – nível máximo de referência para a tigela da água;
5. *nivelref_racao* – nível máximo de referência para a tigela da ração;
6. ~~Utilizador~~ – um ou mais utilizadores aos quais a um ou mais comedouros estão atribuídos.

Para a entidade **Animal** foram definidos os seguintes atributos:

1. *id* – chave primária;
2. *nome* – nome do animal;
3. *tipo* – cão ou gato;
4. *peso* – peso do animal;
5. *idade* – idade do animal;
6. *dosagem_racao* – dose de ração para cada período de alimentação.
7. ~~*Comedouro*~~ – um único comedouro ao qual um único animal está atribuído.

Para a entidade **Depósito** foram definidos os seguintes atributos:

1. *id* – chave primária;
2. *nivelref_max* – nível máximo de referência do depósito interno do comedouro;
3. *nivelref_min* – nível mínimo de referência do depósito interno do comedouro;
4. ~~*Comedouro*~~ – um único comedouro ao qual um único depósito está atribuído.

Para a entidade **Horários** foram definidos os seguintes atributos:

1. *id* – chave primária;
2. *periodicidade* – frequência a que se repete o horário;
3. *hora* – hora definida para a alimentação do animal;
4. ~~*Comedouro*~~ – um ou mais comedouros aos quais um ou mais horários estão atribuídos.

Para a entidade **Histórico** foram definidos os seguintes atributos:

1. *id* – chave primária;
2. *time_stamp* – registo da data e hora da recolha de dados do sensor;
3. *sensor* – id do sensor ao qual corresponde o nível recolhido;
4. *nível* – valor do nível recolhido;
5. ~~*Comedouro*~~ – um ou mais comedouros aos quais poderá ou não um histórico estar atribuído.

Para a entidade **Mensagens** foram definidos os seguintes atributos:

1. *id* – chave primária;
2. *criticidade* – nível de criticidade da mensagem (alerta ou notificação);
3. *nome* – título da mensagem;
4. *descrição* – descrição da mensagem;
5. ~~*Comedouro*~~ – zero ou mais comedouros aos quais uma ou mais mensagens estão atribuídas.

3.3. Relações entre Entidades (MER)

Através das entidades e atributos apresentados no tópico anterior, foi possível estabelecer os relacionamentos entre entidades e as respectivas restrições.

Para o relacionamento entre *Utilizador* e *Comedouro*, um *Utilizador* pode ter zero ou vários *Comedouros* (0N) e um *Comedouro* pode ter um ou vários *Utilizadores* (1N).

Para o relacionamento entre *Comedouro* e *Animal*, um *Comedouro* apenas pode ter um único *Animal* (1) e um *Animal* apenas pode ter um único *Comedouro* (1).

Para o relacionamento entre *Comedouro* e *Depósito*, um *Comedouro* apenas pode ter um único *Depósito* (1) e um *Depósito* apenas um único *Comedouro* (1).

Para o relacionamento entre *Comedouro* e *Horários*, um *Comedouro* pode ter um ou vários *Horários* (1N) e um *Horário* pode ter um ou vários *Comedouros* (1N).

Para o relacionamento entre *Comedouro* e *Histórico*, um *Comedouro* pode ter um ou vários *Históricos* (1N) e um *Histórico* pode ter zero ou vários *Comedouros* (0N).

Para o relacionamento entre *Comedouro* e *Mensagens*, um *Comedouro* pode ter uma ou vários *Mensagens* (1N) e uma *Mensagem* pode zero ou vários *Comedouros* (0N).

4. Diagrama Entidade-Relação (DER)

4.1. Proposta Inicial

Na Figura 1 em anexo encontra-se representado o Diagrama ER desenvolvido com recurso ao *software* draw.io.

No diagrama encontram-se representadas as entidades com os respetivos atributos analisados nos tópicos anteriores, assim como, a simbologia para a identificação das relações e restrições entre entidades.

4.2. Revisões ao DER

Após a realização da discussão da fase anterior do projeto, foi necessário realizar alterações ao Diagrama Entidade-Relação. As alterações corresponderam à eliminação do atributo *dosagem_racao* na entidade *Animal*, e à alteração do atributo *periodicidade* para *dosagem_racao* na entidade *Horários*.

Esta alteração deveu-se ao facto do atributo *periodicidade* ser redundante num horário visto este ser baseado numa hora definida pelo utilizador. Em relação ao atributo *dosagem_racao*, concluiu-se que seria mais lógico a quantidade de dosagens estar associada a um horário ao invés de um animal.

O novo Diagrama Entidade-Relação encontra-se representado na Figura 2 em anexo.

5. Modelo Relacional (MR)

Em seguida, com recurso ao *software ERDPlus*, foi desenvolvido o Modelo Relacional de acordo com as relações estabelecidas entre entidades e o Diagrama Entidade-Relação apresentado no tópico anterior.

Este modelo encontra-se representado na Figura 3 em anexo e permite representar como serão compostas cada tabela, correspondentes a cada entidade da base de dados do projeto, com os respetivos atributos, chaves primárias e chaves estrangeiras representadas.

Neste modelo também foram representadas as tabelas resultantes das relações de muitos para muitos, correspondentes às tabelas *Utilizador_Comedouro*, *Historico_Comedouro* e *Mensagens_Comedouro*, onde foram incluídas como chaves estrangeiras as chaves primárias de cada entidade da relação. Nas restantes entidades *Horarios*, *Deposito* e *Animal*, foi incluída como chave estrangeira a chave primária da entidade *Comedouro*.

6. Scripts Desenvolvidos

Em anexo ao ficheiro do relatório seguem quatro ficheiros texto desenvolvidos, com recurso ao *software MySQL*, para a criação de tabelas (*create.txt*), para a inserção de dados nas tabelas (*populate.txt*), para a inserção de dados errados para que falhem a introdução nas tabelas (*populateErros.txt*) e para as consultas dos dados inseridos (*consultas.txt*).

6.1. Criação das Tabelas

Para a criação das tabelas foi utilizada a linguagem de definição de dados DDL, através da instrução “*CREATE TABLE*”. A definição das chaves primárias em cada tabela foi dada através da instrução “*PRIMARY KEY (atributo)*”, sendo que cada chave primária apresenta como restrições “*NOT NULL*” e “*AUTO_INCREMENT*”. A definição das chaves estrangeiras foi dada através da instrução “*FOREIGN KEY (atributo) REFERENCES (...)*”. Nas tabelas de entidades referenciadas, foi implementada a restrição “*ON DELETE CASCADE*” e “*ON UPDATE CASCADE*”. De notar que, o processo de criação das tabelas teve como base o modelo relacional apresentado no tópico 5.

6.2. Inserção de Dados

Em cada tabela foram inseridos diversos dados de teste com a utilização da linguagem de manipulação de dados DML, através da instrução “*INSERT INTO*”.

Como o projeto *Smart Pet Feeder* não foi finalizado até à data de entrega do presente projeto, a base de dados ainda não estava integrada com o restante sistema, logo, os dados inseridos correspondem a uma aproximação da realidade com o objetivo de serem simuladas as consultas da base de dados.

6.3. Inserção de Erros

Foram também introduzidas falhas na inserção, remoção e atualização dos dados, por forma a testar as falhas relacionadas com o desrespeito às restrições definidas no modelo relacional e na criação das tabelas.

Ao todo foram introduzidas cinco falhas, sendo que, o código de cada uma encontra-se devidamente comentado no script “*populateErros.txt*”, em anexo ao presente relatório.

6.4. Consulta das Tabelas

Para as consultas da base de dados foi utilizada a linguagem DML em que, com base nos requisitos impostos pelo guia do projeto, foram desenvolvidas doze consultas.

As consultas foram desenvolvidas com o objetivo de simular as consultas realizadas pela aplicação móvel e pela lógica de controlo implementada no ambiente de desenvolvimento *Node-RED*, em operação no Hub Central.

7. Conclusão

Com a conclusão do Projeto de Base de Dados referente ao projeto *Smart Pet Feeder* inserido na UC de Projeto em Internet das Coisas, foi possível desenvolver uma base de dados pronta a ser integrada no sistema de um comedouro inteligente para animais.

Este projeto foi composto por várias etapas de desenvolvimento, das quais, a especificação de requisitos e o desenvolvimento do Diagrama Entidade-Relação que foram inseridas na primeira fase, e na segunda fase o desenvolvimento do Modelo Relacional e dos *scripts* com recurso ao *software MySQL*. Entre a primeira fase e a segunda existiu um momento de autoavaliação do projeto que permitiu a identificação de possíveis ajustes ao mesmo.

Posteriormente ao desenvolvimento da primeira iteração do Modelo e do Diagrama Entidade-Relação da base de dados, foram identificadas lacunas relacionadas com a falta de atributos importantes em certas entidades, dos quais, na entidade *Depósito* a ausência de atributos para níveis de referência nos depósitos internos de cada comedouro, e a substituição do atributo *periodicidade* para *dosagem_racao* na entidade *Horários*. Após a correção destas lacunas procedeu-se à continuação do desenvolvimento das restantes etapas.

Como não foi possível finalizar o projeto *Smart Pet Feeder* até à data de entrega do presente Projeto de Base de Dados, não foi possível integrar a mesma com o restante sistema do comedouro inteligente. Em falta, ficaram a realização de testes em contexto prático, os quais, foram realizados durante as semanas seguintes à data de entrega.

8. Anexos

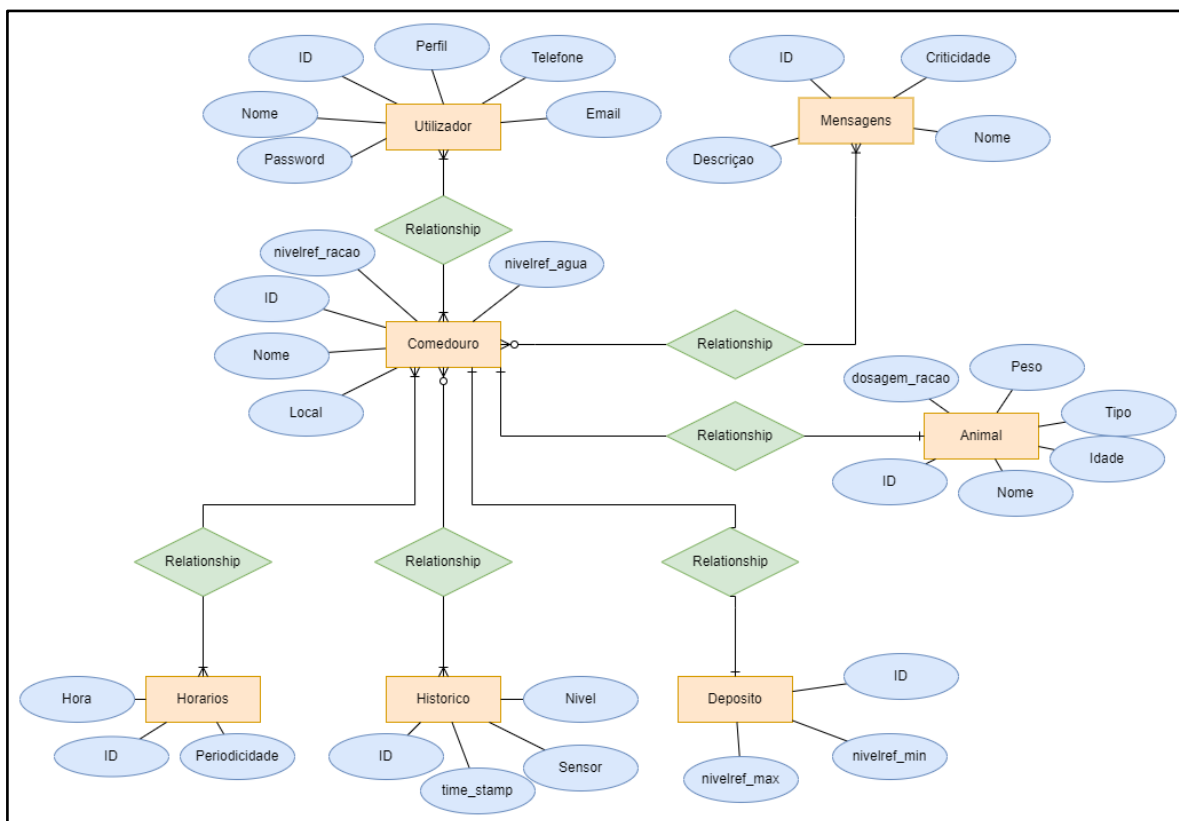


Figura 1 - Primeira iteração do Diagrama Entidade-Relação.

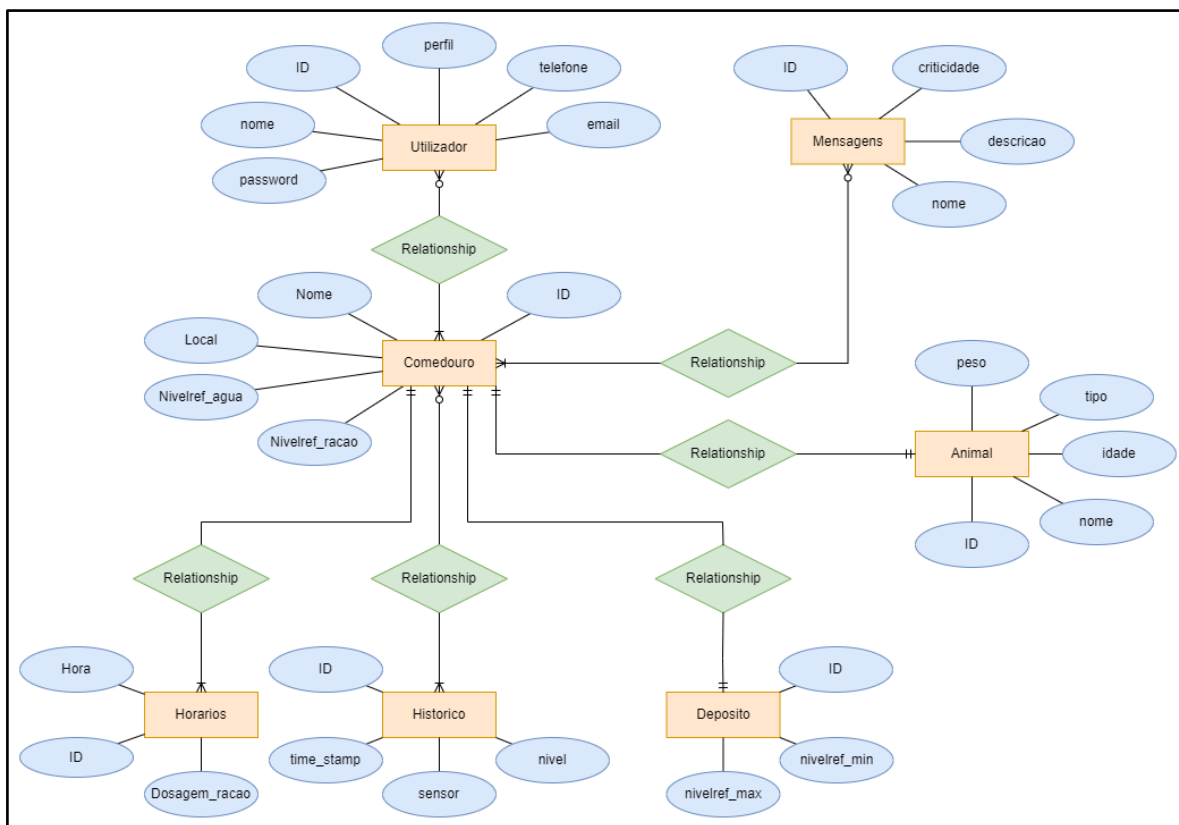


Figura 2 - Segunda iteração do Diagrama Entidade-Relação.

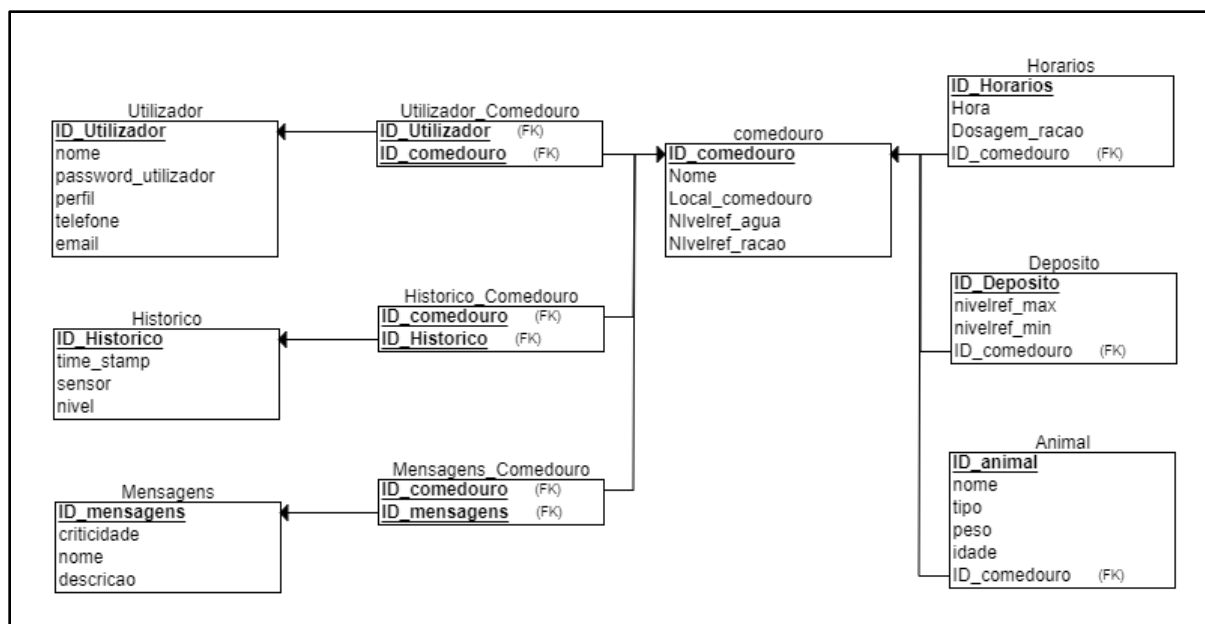


Figura 3 - Modelo Relacional da base de dados do projeto.