

IS4PetCare

Information System for Pet Care

-Projeto de Base de dados, Parte 1

Rafael dos Reis Ferreira 20210773

Teresa Maria Silva Salcedas Cruz 20210772

**Abril 2022**

**Índice de capítulos**

[1. Introdução 1](#_Toc104757315)

[2. Modelo de Casos de Uso 2](#_Toc104757316)

[3. Modelo ER 4](#_Toc104757317)

[4. Modelo Relacional 6](#_Toc104757318)

[5. Tabelas em SQL 7](#_Toc104757319)

[6. Interrogações 11](#_Toc104757320)

[2.5 11](#_Toc104757321)

[2.6 13](#_Toc104757322)

[7. Conclusão 17](#_Toc104757323)

Índice de figuras

[Figura 1- Modelo de Casos de Uso 2](#_Toc104757308)

[Figura 2- Legenda das relações do modelo ER 4](file:///C:\Users\teres\Downloads\PBD_Trabalho_Grupo3_Relatorio.docx#_Toc104757309)

[Figura 3- Modelo ER 4](file:///C:\Users\teres\Downloads\PBD_Trabalho_Grupo3_Relatorio.docx#_Toc104757310)

1. Introdução

O seguinte relatório do trabalho prático da cadeira de projeto de base de dados, tem como objetivo mostrar como foi elaborado o trabalho, como foi interpretado o enunciado, mostrar tabelas, diagramas, bem como responder às questões que nos foram apresentadas. Para isto a metodologia de trabalho utilizada foi a aplicação do conhecimento adquirido nas aulas durante o semestre, consulta das fichas de trabalho disponibilizadas juntamente com os respetivos exercícios e também a consulta de fichas disponibilizadas no semestre passado relativas à cadeira de modelação de base de dados, que nos ajudaram bastante na parte de construção dos modelos E.R e relacional.

Para este trabalho, como SGBD foi selecionado o Microsoft SQL Server Management Studio 18.

2. Modelo de Casos de Uso

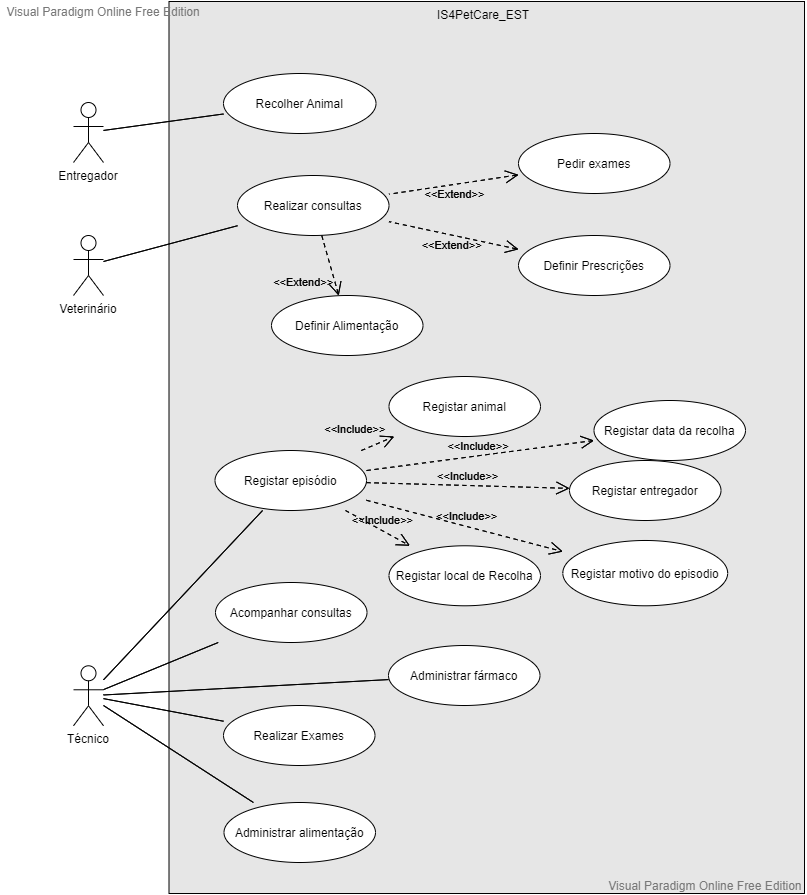


Figura 1- Modelo de Casos de Uso

Como podemos observar na nossa figura 1 acima, a nossa organização tem 3 atores, o entregador, o veterinário e o técnico.

Seguindo a lógica do nosso modelo relacional foi decidido que o entregador tem a funcionalidade de recolher o animal para a nossa organização, onde lá o nosso técnico regista um novo episódio com a chegada do animal, nesse registo ele tem que registar o animal, o entregador que o trouxe, o local onde foi recolhido, a sua data de recolha e o motivo. Com isso temos depois o nosso veterinário que tem como funcionalidade realizar consultas, onde após isso tem a opção de pedir exames, definir prescrições e ainda definir a alimentação para o animal. Voltando para o técnico, este também pode acompanhar certas consultas, realizar exames, e administrar os fármacos prescritos e a alimentação.

3. Modelo ER

Uma imagem com texto

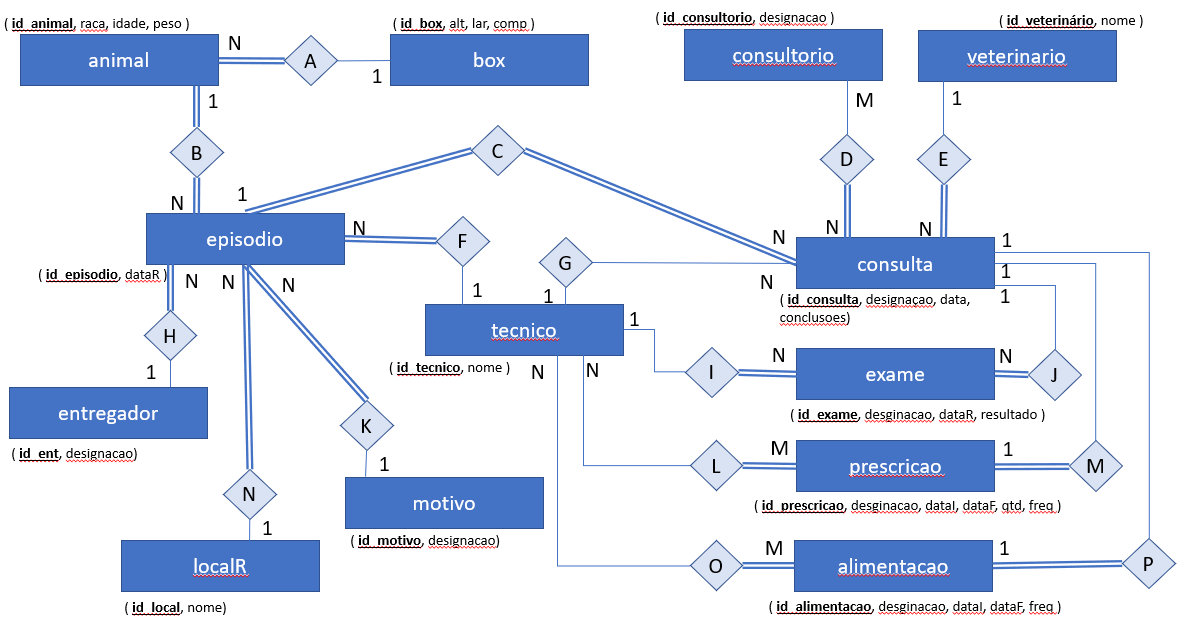
Descrição gerada automaticamente

Figura 2- Legenda das relações do modelo ER

Figura 3- Modelo ER

A figura acima (Figura 3) representa o modelo ER, onde nos é apresentado todas as entidades com os respetivos dados, bem como as relações entre as mesmas.

Na figura ao lado (Figura 2), estão representados os nomes das ligações do modelo ER.

Começou-se por definir quais tabelas existiriam no nosso sistema e chegámos á conclusão de que teríamos as tabelas animal, box, episodio, técnico, entregador, localR, motivo, consulta, consultório, veterinário, exame, prescrição e alimentação.

Tudo começa na nossa tabela episodio onde temos como campos (id\_episodio, dataR- data de recolha). É nesta tabela onde todos os dados da recolha do animal serão processados, portanto a esta tabela temos ligadas a tabela entregador que corresponde à entidade que entrega o animal, a tabela localR que é o local onde foi recolhido o animal, a tabela motivo onde é indicado o motivo porque o animal foi para á organização e a tabela técnico que é quem toma registo da ocorrência, nestas tabelas a relação que temos é de 1 para vários com obrigatoriedade do lado do episodio, pois sem nenhum destes elementos não existe nenhuma ocorrência. Temos também ligada ao episodio a tabela animal que corresponde ao animal que foi entregue na organização, aqui a nossa relação com a tabela episodio varia um pouco sendo de 1 para vários pois um episodio corresponde a um só animal, mas com obrigatoriedade dos dois lados, pois a nosso entender só existe animal registado se já tiver tido algum episodio e cada episodio tem de ter um animal associado. Ligada à tabela animal está também a tabela box que é onde este fica até que consiga ser visto por um veterinário, aqui a relação é de vários para 1, visto que um animal fica numa box mas uma box poderá servir para diversos animais.

Tendo o episódio sido registado passamos para a área de análise, onde para isso temos uma ligação da nossa tabela episódio para a nossa tabela consulta com os campos (id\_consulta, designação –motivo da consulta, data –data da consulta, conclusões –a conclusão da consulta, se terminou em abate, operação etc.. ), a relação entre estas é de 1 para vários com obrigatoriedade dos dois lados, visto que apesar de uma consulta ser especifica de um episodio, o episodio pode originar várias consultas, por exemplo, um animal que deu entrada por atropelamento pode necessitar de uma nova consulta para modificar o tratamento que lhe foi indicado na primeira consulta.

Já ligada á nossa tabela consulta temos a tabela consultório que será onde as consultas são realizadas, aqui temos uma relação de vários para vários com obrigatoriedade do lado da consulta, pois dependendo do tratamento ou situações que vão surgindo a meio poderá ser necessário outro tipo de consultório, por exemplo se tenho um consultório para fazer o check up de um animal não vou efetuar o abate do mesmo nesse consultório, para isso teremos que mover o animal para um consultório especifico para isso. Também ligada á consulta temos a tabela veterinário que será o medico que cuidará do animal, e aqui temos relação de 1 para vários com obrigatoriedade do lado do episodio. Temos a tabela técnico ligada à consulta, mas com uma relação de 1 para vários sem obrigatoriedade de nenhum dos lados, pois aqui já é uma questão opcional se a consulta é ou não acompanhada por um técnico. De seguida temos a tabela exame que se encontra ligada à tabela consulta e á tabela técnico, ambas as relações ocorrem de 1 para vários com obrigatoriedade do lado do exame, tendo em consideração que nenhuma consulta precisa obrigatoriamente de pedir exames e nem todos os técnicos precisam obrigatoriamente de realizar exames. Por fim temos as tabelas prescrição que corresponde à medicação prescrita pelo veterinário, e a tabela alimentação que corresponde à alimentação prescrita pelo veterinário. Ambas as tabelas estão ligadas à tabela consulta com uma relação de 1 para 1 com obrigatoriedade do lado da prescrição e da alimentação, e à tabela técnico com uma relação de vários para vários com obrigatoriedade do lado da prescrição e da alimentação, pois cada dose de cada tabela pode ser administrada por técnicos diferentes.

4. Modelo Relacional

Seguindo as relações impostas no nosso capítulo anterior temos o seguinte modelo relacional simplificado.

Animal (id\_animal (PK), raca, idade, peso, id\_box (FK) **--A**)

Episodio (id\_episodio (PK), datar, id\_animal (FK) **--B**, id\_tecnico (FK) **--F**, id\_ent (FK) **--H**, id\_motivo (FK) **--K**, id\_local (FK) **--N**)

Consulta (id\_consulta (PK), designação, data, conclusões, id\_veterinario (FK) **--E**, id\_episodio (FK) **--C**)

Consultório\_consulta (id\_consultorio (PK)(FK), id\_consulta (PK)(FK)) **--D**

Técnico\_consulta (id\_consulta (PK)(FK), id\_tecnico (PK)(FK)) **--G**

Exame (id\_exame (PK), designação, dataR, resultado, id\_tecnico (FK) **--I**)

Consulta\_exame (id\_consulta (PK)(FK), id\_exame (PK)(FK)) **--J**

Técnico\_prescricao (id\_tecnico (PK)(FK), id\_prescricao (PK)(FK)) **--L**

Prescrição (id\_prescricao (PK), designação, dataI, dataF, qtd, freq, id\_consulta (FK) **--M**)

Técnico\_alimentacao (id\_tecnico (PK)(FK), id\_alimentacao (PK)(FK)) **--O**

alimentacao (id\_alimentacao (PK), designação, dataI, dataF, freq, id\_consulta (FK) **--P**)

5. Tabelas em SQL

Começamos por criar a base das nossas tabelas com a ajuda do nosso modelo relacional. Na nossa criação identificamos logo as chaves primárias das tabelas adicionando uma constraint nas mesmas. Para a criação das tabelas usamos a sintaxe CREATE TABLE.

CREATE TABLE tecnico (id\_tecnico INT, nome VARCHAR(20),

CONSTRAINT tb\_tecnico\_pk PRIMARY KEY(id\_tecnico));

CREATE TABLE entregador (id\_ent INT, designacao VARCHAR(20),

CONSTRAINT tb\_ent\_pk PRIMARY KEY(id\_ent));

CREATE TABLE motivo (id\_motivo INT, designacao VARCHAR(20),

CONSTRAINT tb\_motivo\_pk PRIMARY KEY(id\_motivo));

CREATE TABLE localR (id\_local INT, nome VARCHAR(20),

CONSTRAINT tb\_local\_pk PRIMARY KEY(id\_local));

CREATE TABLE veterinario (id\_veterinario INT, nome VARCHAR(20),

CONSTRAINT tb\_veterinario\_pk PRIMARY

KEY(id\_veterinario));

CREATE TABLE consultorio (id\_consultorio INT, designacao VARCHAR(20),

CONSTRAINT tb\_consultorio\_pk PRIMARY

KEY(id\_consultorio));

CREATE TABLE consulta (id\_consulta INT, designacao VARCHAR(70), id\_veterinario INT, id\_episodio INT, dataA DATE, conclusoes VARCHAR(100),

CONSTRAINT tb\_consulta\_pk PRIMARY KEY(id\_consulta));

CREATE TABLE consultorio\_consulta(id\_consultorio INT, id\_consulta INT,

CONSTRAINT tb\_consultorioConsulta\_pk PRIMARY KEY(id\_consultorio, id\_consulta));

CREATE TABLE exame (id\_exame INT, designacao VARCHAR(20), dataR DATE, resultado VARCHAR(70), id\_tecnico INT, id\_consulta INT,

CONSTRAINT tb\_exame\_pk PRIMARY KEY(id\_exame));

CREATE TABLE prescricao (id\_prescricao INT, designacao VARCHAR(20), dataI DATE, dataF DATE, qtd INT, freq INT, id\_consulta INT,

CONSTRAINT tb\_prescicao\_pk PRIMARY KEY(id\_prescricao));

CREATE TABLE alimentacao (id\_alimentacao INT, designacao VARCHAR(20), dataI DATE, dataF DATE, freq INT, id\_consulta INT,

CONSTRAINT tb\_alimentacao\_pk PRIMARY KEY(id\_alimentacao));

CREATE TABLE animal (id\_animal INT, raca VARCHAR(20), idade INT, peso DECIMAL(5,2), id\_box INT,

CONSTRAINT tb\_animal\_pk PRIMARY KEY(id\_animal));

CREATE TABLE box (id\_box INT, alt INT, lar INT, comp INT,

CONSTRAINT tb\_box\_pk PRIMARY KEY(id\_box));

CREATE TABLE episodio (id\_episodio INT, dataR DATE, id\_animal INT, id\_tecnico INT, id\_ent INT, id\_motivo INT, id\_local INT,

CONSTRAINT tb\_episodio\_pk PRIMARY KEY(id\_episodio));

CREATE TABLE tecnico\_consulta (id\_consulta INT, id\_tecnico INT,

CONSTRAINT tb\_tecnicoConsulta\_pk PRIMARY KEY(id\_consulta, id\_tecnico));

CREATE TABLE tecnico\_prescricao (id\_tecnico INT, id\_prescricao INT,

CONSTRAINT tb\_tecnicoPresc\_pk PRIMARY KEY(id\_tecnico, id\_prescricao));

CREATE TABLE tecnico\_alimentacao (id\_tecnico INT, id\_alimentacao INT,

CONSTRAINT tb\_tecnicoAlim\_pk PRIMARY KEY(id\_tecnico, id\_alimentacao));

Feitas as tabelas avançamos para a criação das chaves externas onde para isso utilizaremos a sintaxe ALTER TABLE. Para cada chave externa criamos uma constraint para assim ser mais fácil de perceber os possíveis erros e ser mais fácil de eliminar.

ALTER TABLE consulta

ADD CONSTRAINT tb\_consulta\_FK1

FOREIGN KEY (id\_veterinario) REFERENCES veterinario (id\_veterinario);

ALTER TABLE consulta

ADD CONSTRAINT tb\_consulta\_FK2

FOREIGN KEY (id\_episodio) REFERENCES episodio (id\_episodio);

ALTER TABLE consultorio\_consulta

ADD CONSTRAINT tb\_consultorioConsulta\_FK1

FOREIGN KEY (id\_consultorio) REFERENCES consultorio (id\_consultorio);

ALTER TABLE consultorio\_consulta

ADD CONSTRAINT tb\_consultorioConsulta\_FK2

FOREIGN KEY (id\_consulta) REFERENCES consulta (id\_consulta);

ALTER TABLE exame

ADD CONSTRAINT tb\_exame\_FK1

FOREIGN KEY (id\_tecnico) REFERENCES tecnico (id\_tecnico);

ALTER TABLE exame

ADD CONSTRAINT tb\_exame\_FK2

FOREIGN KEY (id\_consulta) REFERENCES consulta (id\_consulta);

ALTER TABLE prescricao

ADD CONSTRAINT tb\_prescricao\_FK1

FOREIGN KEY (id\_consulta) REFERENCES consulta(id\_consulta);

ALTER TABLE alimentacao

ADD CONSTRAINT tb\_alimentacao\_FK1

FOREIGN KEY (id\_consulta) REFERENCES consulta(id\_consulta);

ALTER TABLE animal

ADD CONSTRAINT tb\_animal\_FK1

FOREIGN KEY (id\_box) REFERENCES box(id\_box);

ALTER TABLE episodio

ADD CONSTRAINT tb\_episodio\_FK1

FOREIGN KEY (id\_animal) REFERENCES animal(id\_animal);

ALTER TABLE episodio

ADD CONSTRAINT tb\_episodio\_FK2

FOREIGN KEY (id\_tecnico) REFERENCES tecnico(id\_tecnico);

ALTER TABLE episodio

ADD CONSTRAINT tb\_episodio\_FK3

FOREIGN KEY (id\_ent) REFERENCES entregador(id\_ent);

ALTER TABLE episodio

ADD CONSTRAINT tb\_episodio\_FK4

FOREIGN KEY (id\_motivo) REFERENCES motivo(id\_motivo);

ALTER TABLE episodio

ADD CONSTRAINT tb\_episodio\_FK5

FOREIGN KEY (id\_local) REFERENCES localR(id\_local);

ALTER TABLE tecnico\_consulta

ADD CONSTRAINT tb\_tecnicoConsulta\_FK1

FOREIGN KEY (id\_consulta) REFERENCES consulta(id\_consulta);

ALTER TABLE tecnico\_consulta

ADD CONSTRAINT tb\_tecnicoConsulta\_FK2

FOREIGN KEY (id\_tecnico) REFERENCES tecnico(id\_tecnico);

ALTER TABLE tecnico\_prescricao

ADD CONSTRAINT tb\_tecnicoPres\_FK1

FOREIGN KEY (id\_tecnico) REFERENCES tecnico(id\_tecnico);

ALTER TABLE tecnico\_prescricao

ADD CONSTRAINT tb\_tecnicoPres\_FK2

FOREIGN KEY (id\_prescricao) REFERENCES prescricao(id\_prescricao);

ALTER TABLE tecnico\_alimentacao

ADD CONSTRAINT tb\_tecnicoAlim\_FK1

FOREIGN KEY (id\_tecnico) REFERENCES tecnico(id\_tecnico);

ALTER TABLE tecnico\_alimentacao

ADD CONSTRAINT tb\_tecnicoAlim\_FK2

FOREIGN KEY (id\_alimentacao) REFERENCES alimentacao(id\_alimentacao);

6. Interrogações

## 2.5

--i. Faça a listagem geral de todas as tabelas;

SELECT \* FROM tecnico;

SELECT \* FROM entregador;

SELECT \* FROM motivo;

SELECT \* FROM localR;

SELECT \* FROM veterinario;

SELECT \* FROM consultorio;

SELECT \* FROM exame;

SELECT \* FROM prescricao;

SELECT \* FROM alimentacao;

SELECT \* FROM animal;

SELECT \* FROM box;

SELECT \* FROM tecnico\_consulta;

SELECT \* FROM tecnico\_prescricao;

SELECT \* FROM tecnico\_alimentacao;

SELECT \* FROM consulta;

SELECT \* FROM episodio;

SELECT \* FROM consultorio\_consulta;

--ii. Liste toda a informação sobre as pessoas, que tenham ‘Rita’ no nome, e que tenham entregado animais no primeiro semestre de 2018;

SELECT ent.id\_ent, ent.designacao FROM entregador ent INNER JOIN episodio ep

ON ent.id\_ent=ep.id\_ent

WHERE ent.designacao LIKE '%Rita%' AND dataR BETWEEN '2018-1-1' AND '2018-6-30';

--iii. Liste os nomes dos animais, a sua raça e as datas das suas entregas, que terminaram em abate no matador ‘Muerte’. Ordene o resultado listando do mais recente para o mais antigo;

SELECT raca, dataR FROM animal a INNER JOIN episodio e ON a.id\_animal=e.id\_animal INNER JOIN consulta c ON e.id\_episodio=c.id\_episodio

WHERE c.id\_consulta IN (SELECT cc.id\_consulta FROM consultorio\_consulta cc

INNER JOIN consultorio co ON co.id\_consultorio=cc.id\_consultorio

WHERE c.id\_consulta=cc.id\_consulta AND c.conclusoes='Abate' AND co.designacao='Muerte')

ORDER BY dataR DESC;

--iv. Liste os animais e respetivas localizações (jaulas/box) que estão em tratamento e que ainda não fOram devolvidos ao seu habitat. Ordene pela data de admissão.

SELECT raca, id\_box FROM animal a INNER JOIN episodio ep ON a.id\_animal=ep.id\_animal

WHERE ep.id\_episodio IN (SELECT c.id\_episodio FROM consulta c

WHERE c.id\_episodio=ep.id\_episodio AND c.conclusoes='tratamento')

ORDER BY dataR;

--v. Indique as consultas realizadas pelo médico “Rui”, em que foi solicitado a realização do exame “RX”.

SELECT DISTINCT c.id\_consulta, c.designacao FROM consulta c INNER JOIN veterinario v ON c.id\_veterinario=v.id\_veterinario INNER JOIN exame e ON e.id\_consulta=c.id\_consulta

WHERE e.designacao='RX' AND nome='Rui';

/\*vii. Liste para cada consulta a quantidade de exames que foram realizados. Tenha em consideração que podem ter existido consultas onde não foram realizados quaisquer exames.

Estas consultas devem também constar na resposta, com a indicação de quantidade zero;\*/

SELECT c.id\_consulta, c.designacao, COUNT(id\_exame) AS 'Qtd Exames' FROM consulta c LEFT OUTER JOIN exame e ON c.id\_consulta=e.id\_consulta

GROUP BY c.id\_consulta, c.designacao;

--ix. Indique a raça dO primeiro animal a ser entregue;

SELECT raca FROM animal a INNER JOIN episodio e ON a.id\_animal=e.id\_animal

WHERE dataR=(SELECT MIN(dataR) FROM episodio);

/\*x. Crie uma cOnsulta que demostre a participação dOs técnicOs nas cOnsultas. Tenha em cOnsideração O fato de poderem existirem cOnsultas sem técnicOs, bem cOmo técnicos que não tenham participadO em qualquer cOnsulta. Ambos Os casos deverão ter reflexo na resposta;\*/

SELECT nome, COUNT(tc.id\_consulta) AS 'Qtd consultas' FROM tecnico t FULL OUTER JOIN tecnico\_consulta tc ON t.id\_tecnico=tc.id\_tecnico FULL OUTER JOIN consulta c ON tc.id\_consulta=c.id\_consulta

GROUP BY nome;

--xii. EncOntre episódiOs distintos que tenham tidO cOnsultas Onde fOi realizadO O mesmo exame cOm o mesmo valOr no resultadO;

select DISTINCT ep.id\_episodio

from episodio ep INNER JOIN consulta c ON ep.id\_episodio=c.id\_episodio INNER JOIN exame e ON e.id\_consulta=c.id\_consulta

WHERE EXISTS (SELECT designacao, resultado, count(\*) as quantidade

FROM exame e1

GROUP BY designacao, resultado

HAVING count(\*) > 1)

--xiii. Indique quais as prescrições que têm fármacOs prescritos em quantidade superiOr à média das quantidades nessa mesma prescrição;

SELECT id\_prescricao FROM prescricao

WHERE qtd>=(SELECT AVG(qtd) FROM prescricao);

--xiv. Indique os principais motivos porque um animal é recolhido e quantas consultas já foram realizados para cada um;

SELECT m.designacao, COUNT(id\_consulta) AS 'Número de consultas por motivo'

FROM motivo m INNER JOIN episodio e ON m.id\_motivo=e.id\_motivo INNER JOIN consulta c ON e.id\_episodio=c.id\_episodio

WHERE c.id\_episodio IN (SELECT c1.id\_episodio FROM consulta c1

WHERE c.id\_episodio=c1.id\_episodio)

GROUP BY m.designacao

HAVING COUNT(m.id\_motivo)>1;

--xv. Crie uma vista que indique os locais de recolha (sem repetições) cujas consultas deram em abate;

CREATE VIEW abate

AS

SELECT DISTINCT l.nome

FROM localR l INNER JOIN episodio e ON l.id\_local=e.id\_local

WHERE l.id\_local IN (SELECT e1.id\_local

FROM episodio e1 INNER JOIN consulta c ON e1.id\_episodio=c.id\_episodio

WHERE l.id\_local=e1.id\_local AND c.conclusoes LIKE 'abate');

SELECT \* FROM abate;

## 2.6

/\*Liste os animais que tenham a letra 'e'\*/

SELECT \* FROM animal

WHERE raca LIKE '%e%';

/\*Liste as box cuja largura esteja entre 7 e 10\*/

SELECT \* FROM Box

WHERE lar BETWEEN 7 AND 10;

/\*Liste os veterinários cujo nome não termine em o\*/

SELECT \* FROM veterinario

WHERE nome NOT LIKE '%o';

/\*Liste os locais onde os animais com id=5 e id=9 foram recolhidos com o nome dos locais\*/

SELECT e.id\_animal, l.nome

FROM episodio e INNER JOIN localR l ON e.id\_local=l.id\_local

WHERE e.id\_animal IN (5,9);

/\*Liste as consultas resultadas de episódios em que a sua data não esteja entre 2016 e 2018\*/

SELECT c.\* FROM consulta c INNER JOIN episodio e ON c.id\_episodio=e.id\_episodio

WHERE YEAR(dataR) NOT BETWEEN 2016 AND 2018;

/\*Liste as consultas que não foram realizadas nos consultórios com id=201 e id=202\*/

SELECT c.\* FROM consulta c INNER JOIN consultorio\_consulta cc ON c.id\_consulta=cc.id\_consulta

WHERE id\_consultorio NOT IN (201, 202);

/\*Quantos animais foram recolhidos na avenida\*/

SELECT COUNT(id\_animal) AS 'Animais recolhidos na avenida'

FROM episodio e INNER JOIN localR l ON e.id\_local=l.id\_local

WHERE nome LIKE '%avenida%';

/\*Qual o peso mais baixo entre os animais?\*/

SELECT MIN(peso) AS 'Peso mais baixo' FROM animal;

/\*Liste para cada motivo de recolha de animais qual a recolha mais recente\*/

SELECT designacao, MAX(ep.dataR) AS 'Data Mais Recente'

FROM motivo m INNER JOIN episodio ep ON m.id\_motivo=ep.id\_motivo

GROUP BY designacao;

/\*Calcule a média de frequencia que cada alimentacao tem que ser dada\*/

SELECT designacao, AVG(freq) AS 'Frequencia media dada' FROM alimentacao

GROUP BY designacao;

/\*Calcule a quantidade que tem que ser administrada por prescrição\*/

SELECT designacao, SUM(qtd) AS 'Quantidade Total' FROM prescricao

GROUP BY designacao;

/\*Mostre para cada tecnico quantas consultas este realizou\*/

SELECT nome, COUNT(id\_consulta) AS 'Qtd consultas'

FROM tecnico t LEFT OUTER JOIN tecnico\_consulta c ON t.id\_tecnico=c.id\_tecnico

GROUP BY nome;

/\*Liste para cada consulta a prescrição com maior quantidade de administração que seja maior ou igual que 15\*/

SELECT c.id\_consulta, c.designacao, p.designacao, MAX(qtd) AS 'Maior qtd'

FROM consulta c INNER JOIN prescricao p ON c.id\_consulta=p.id\_consulta

GROUP BY c.id\_consulta, c.designacao, p.designacao

HAVING MAX(qtd) >= 15;

/\*Encontre o veterinário que não realizou nenhuma consulta\*/

SELECT nome FROM veterinario v

WHERE NOT EXISTS (SELECT id\_veterinario FROM consulta c

WHERE v.id\_veterinario=c.id\_veterinario);

/\*Mostre os nomes e o id dos tecnicos que estejam a administrar a alimentacao com id=70\*/

SELECT t.nome, t.id\_tecnico FROM tecnico t

WHERE t.id\_tecnico IN (SELECT t1.id\_tecnico FROM tecnico t1 INNER JOIN tecnico\_alimentacao ta ON t1.id\_tecnico=ta.id\_tecnico

WHERE ta.id\_alimentacao=70);

/\*Crie uma vista contendo para cada episodio a raça do animal, quem recolheu, o motivo e o nome do local\*/

CREATE VIEW InfoRecolha

AS

SELECT a.raca AS 'Animal', e.designacao AS 'Entregador', m.designacao AS 'Motivo', l.nome AS 'Local'

FROM episodio ep INNER JOIN animal a ON ep.id\_animal=a.id\_animal

INNER JOIN entregador e ON e.id\_ent=ep.id\_ent

INNER JOIN motivo m ON m.id\_motivo=ep.id\_motivo

INNER JOIN localR l ON l.id\_local=ep.id\_local

SELECT \* FROM InfoRecolha;

/\*Crie uma vista com os episodios ocorridos no local 117 com o nome do local, data da recolha e o animal recolhido\*/

CREATE VIEW Ep\_local117

AS

SELECT l.nome, e.dataR, a.raca

FROM localR l INNER JOIN episodio e ON l.id\_local=e.id\_local

INNER JOIN animal a ON e.id\_animal=a.id\_animal

WHERE l.id\_local=117;

SELECT \* FROM Ep\_local117;

7. Conclusão

À exceção de algumas questões do 2.5 em que foram sentidas algumas dificuldades a realizar, acredita-se que em geral o trabalho tenha atingido os principais pontos e objetivos.

Todas as decisões foram bem pensadas e justificadas de forma sucinta neste relatório. Em geral estamos satisfeitos com o trabalho realizado, tendo uma boa base para prosseguir com o mesmo na segunda parte do trabalho.