

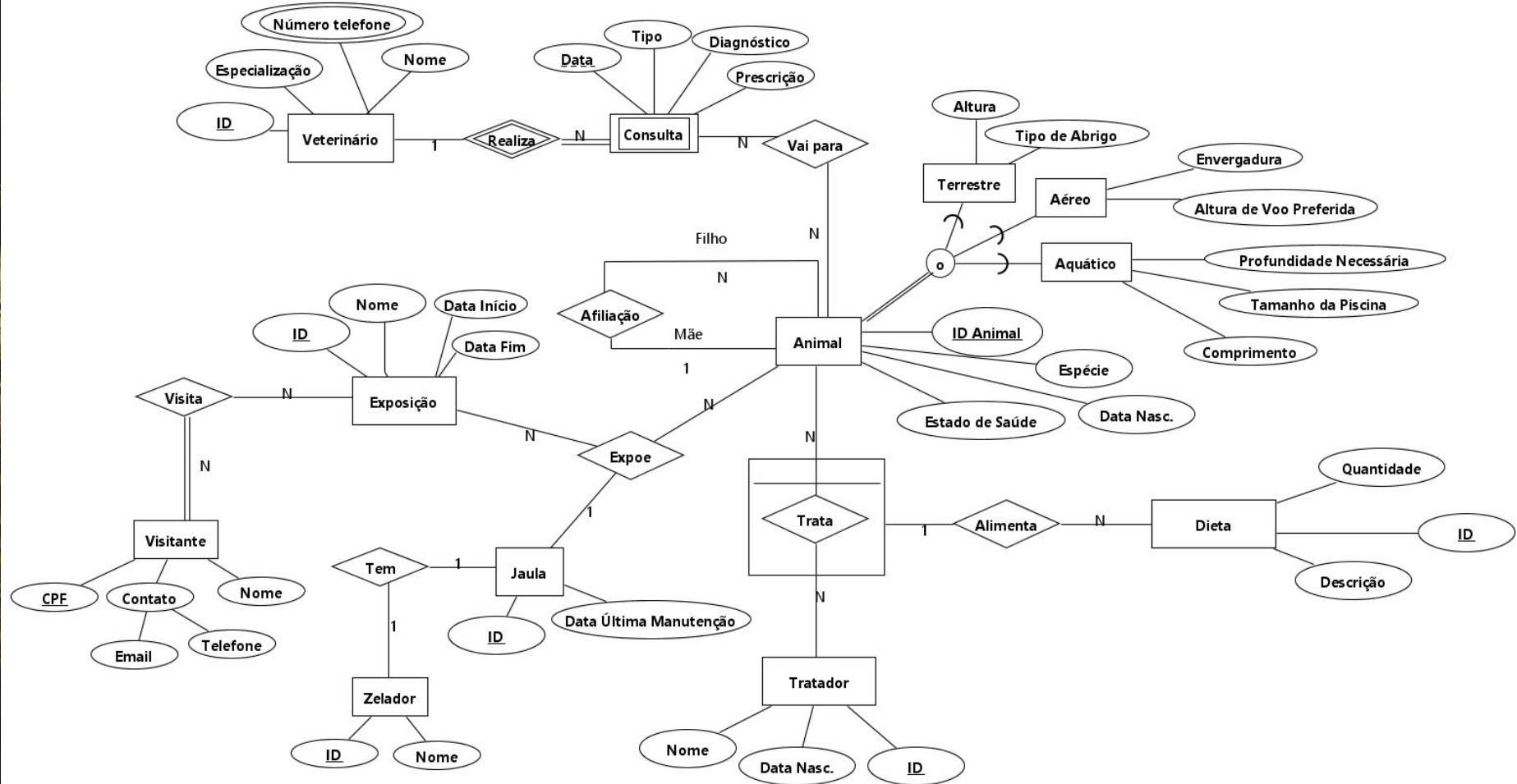


Minimundo

Este é um universo organizacional que mistura o deleite de um funcionamento de um zoológico desde as visitas até os cuidados essenciais de pessoas que se dedicam para com os animais.



Projeto Conceitual



The background of the image is a lush, dark green jungle scene. It features various types of tropical plants, including large, broad leaves and ferns. On the left side, there is a prominent, light-colored rock or tree trunk with a rough, textured surface. A semi-transparent green rectangular box is centered in the middle of the image, containing the text "Projeto Lógico" in a black, sans-serif font.

Projeto Lógico

Animal(IDAnimal, Espécie, EstadoDeSaúde, DataNasc, IDMãe)
IDMãe -> Animal(IDAnimal)

Terrestre(IDAnimal, Altura, TipoDeAbrigo)
IDAnimal -> Animal(IDAnimal)

Aéreo(IDAnimal, Envergadura, AlturaDeVooPreferida)
IDAnimal -> Animal(IDAnimal)

Aquático(IDAnimal, ProfundidadeNecessária, TamanhoDaPiscina, Comprimento)
IDAnimal -> Animal(IDAnimal)

Veterinário(IDVet, Nome, Especialização)

NúmeroTelefone(Telefone, IDVet!)
IDVet-> Veterinário(IDVet)

Consulta(IDVet, Data, Diagnóstico, Prescrição, Tipo)
IDVet-> Veterinário(IDVet)

VaiPara(IDVet, Data, IDAnimal)
(IDVet, Data) -> Consulta(IDVet, Data)
IDAnimal -> Animal(IDAnimal)

Tratador(IDTrat, Nome, DataNasc)

Trata(IDTrat, IDAnimal)

IDTrat -> Tratador(IDTrat)

IDAnimal -> Animal(IDAnimal)

Dieta(IDDieta, Quantidade, Descrição, IDTrat, IDAnimal)
(IDTrat, IDAnimal) -> Trata(IDTrat, IDAnimal)

Zelador(IDZel, Nome)

Jaula(IDJaula, [IDZel], DataManutenção)
IDZel -> Zelador(IDZel)

Exposição(IDExpo, Nome, DataInício, DataFim)

Visitante(CPF, Nome, ContatoEmail, ContatoTelefone)

Visita(CPF, IDExpo)

CPF -> Visitante(CPF)

IDExpo -> Exposicao(IDExpo)

Expõe(IDExpo, IDAnimal, IDJaula)

IDExpo -> Exposição(IDExpo)

IDJaula -> Jaula(IDJaula)

IDAnimal -> Animal(IDAnimal)

The background of the image is a lush, dark green jungle scene. It features various types of tropical plants, including large, broad leaves and feathery ferns. On the left side, a portion of a rough, light-colored rock is visible. A semi-transparent green rectangular box is centered in the middle of the image, containing the text "Projeto Físico" in a black, sans-serif font.

Projeto Físico



```
1  /* Junção interna: Quais as espécies dos animais e o nome dos
2     veterinários envolvidos nas consultas na data 09/09/2023? */
3  SELECT A.ESPECIE, V.NOME
4  FROM ANIMAL A INNER JOIN ANIMAL_CONSULTA AC ON A.ID = AC.ID_ANIMAL
5           INNER JOIN VETERINARIO V ON AC.ID_VET = V.ID
6  WHERE AC.DATA_CONSULTA = TO_DATE('09/09/2023')
```



```
1  /* Junção Externa: quais são os nomes dos visitantes
2     que visitaram a exposição com ID 5678? */
3  SELECT VISITANTE.NOME
4  FROM VISITANTE
5  LEFT JOIN VISITA ON VISITANTE.CPF = VISITA.CPF_VISITANTE
6  LEFT JOIN EXPOSICAO ON VISITA.ID_EXPO = EXPOSICAO.ID
7  WHERE EXPOSICAO.ID = '5678';
```



```
1  /* Group By/Having: Agrupar por nome de exposição,  
2     os que terão mais de 6 visitantes */  
3  SELECT E.NOME  
4  FROM EXPOSICAO E INNER JOIN VISITA A ON A.ID_EXPO = E.ID  
5  GROUP BY E.NOME  
6  HAVING COUNT(*) > 6;
```




```
1  /* Subconsulta Escalar: Quais animais aéreos tem altura
2     de voo maior que a média */
3  SELECT ID, ALTURA_VOO
4  FROM AEREO
5  WHERE ALTURA_VOO > (SELECT AVG(ALTURA_VOO) FROM AEREO);
```



```
1  /* Subconsulta Tabela: Quais os veterinarios que
2     realizam consulta no dia X */
3  SELECT V.NOME FROM VETERINARIO V
4  WHERE V.ID IN (SELECT ID_VET FROM CONSULTA
5                  WHERE TO_CHAR(DATA_CONSULTA, 'DD/MM/YYYY') = '16/09/2023');
```



```
1  /* Subconsulta linha: Saber nome das exposições que vão se
2     acontecem entre os mesmos dias da exposição com o id 3456 */
3  SELECT NOME FROM EXPOSICAO
4  WHERE (DATA_INICIO, DATA_FIM) =
5          (SELECT DATA_INICIO, DATA_FIM
6             FROM EXPOSICAO
7             WHERE ID = 3456);
```




```
1  /* Semi junção: Listar todos as espécies dos animais
2     que participaram de exposições */
3  SELECT DISTINCT A.ESPECIE FROM ANIMAL A
4  WHERE EXISTS (SELECT * FROM EXPOE E
5                 WHERE A.ID = E.ID_ANIMAL);
```



```
1  /* Anti Junção: Quais são os zeladores que não
2     trabalharam depois de DD/MM/YY */
3  SELECT NOME FROM ZELADOR
4  WHERE ID NOT IN (SELECT ID_ZELADOR FROM JAULA
5                     WHERE DATA_MANUTENCAO > TO_DATE('14/03/21'));
```



```
1  -- Operação de conjunto: Lista espécie e id de todos os animais excetos aquaticos.
2  SELECT ESPECIE AS ANIMAL, ID ID_ANIMAL
3  FROM ANIMAL
4  WHERE ID IN (
5      (SELECT ID FROM ANIMAL)
6      EXCEPT
7      (SELECT ID FROM AQUATICO)
8  )
9  ORDER BY ESPECIE, ID;
```




```
1  -- Function: Para quantas consultas um animal de ID tal já foi?
2  CREATE OR REPLACE FUNCTION qtd_consultas_animal (ID VARCHAR2) RETURN NUMBER IS
3      qtdC NUMBER;
4  BEGIN
5      SELECT COUNT(*) INTO qtdC
6      FROM ANIMAL_CONSULTA AC
7      WHERE AC.ID_ANIMAL = ID
8      GROUP BY AC.ID_ANIMAL;
9
10     RETURN qtdC;
11 END;
12 /
13 BEGIN
14     DBMS_OUTPUT.PUT_LINE('Quantidade de consultas do animal: '||qtd_consultas_animal('0011'));
15 END;
```



```
1  -- Tabela e Gatilho: Registro de todas as visitas a exposição
2  CREATE TABLE HISTORICO_VISITAS_EXPOSICAO (
3      CPF_VISITANTE VARCHAR2(11),
4      ID_EXPOSICAO VARCHAR2(4),
5      TEMPO_SISTEMA TIMESTAMP,
6
7      CONSTRAINT FK_HISTORICO_VISITANTE FOREIGN KEY (CPF_VISITANTE) REFERENCES VISITANTE (CPF),
8      CONSTRAINT FK_HISTORICO_EXPOSICAO FOREIGN KEY (ID_EXPOSICAO) REFERENCES EXPOSICAO (ID),
9      CONSTRAINT PK_HISTORICO PRIMARY KEY (TEMPO_SISTEMA, CPF_VISITANTE, ID_EXPOSICAO)
10 );
11
12 CREATE OR REPLACE TRIGGER TRIGGER_INSERIR_HISTORICO
13 AFTER INSERT ON VISITA
14 FOR EACH ROW
15 BEGIN
16     INSERT INTO HISTORICO_VISITAS_EXPOSICAO (CPF_VISITANTE, ID_EXPOSICAO, TEMPO_SISTEMA)
17     VALUES (:NEW.CPF_VISITANTE, :NEW.ID_EXPO, SYSTIMESTAMP AT TIME ZONE 'UTC');
18 END;
19 /
```



```
1  -- Procedimento Quantidade de visitantes por exposicao
2  CREATE OR REPLACE PROCEDURE vis_exp(NOME_EXP VARCHAR)IS
3      QTD_VIS NUMBER;
4
5  BEGIN
6      SELECT COUNT(*) INTO QTD_VIS
7      FROM VISITA
8      WHERE ID_EXPO = (SELECT ID FROM EXPOSICAO
9                      WHERE NOME = NOME_EXP);
10
11      DBMS_OUTPUT.PUT_LINE('Quantidade de visitantes no ' || NOME_EXP || ' foi de: ' || QTD_VIS);
12  END;
13  /
14
15  EXEC vis_exp('SEABORN');
```