

Projeto MATA60 - Elaboração, implantação, governança e uso de banco de dados em estudo de caso

Discente: Robespierre Pita

Docente: Laisa da Silva Pereira

Curso: Sistemas de Informação

Matrícula: 221116328

[Drive para acessar os arquivos do projeto:](#)

1. 1 Contexto e Problemática

Eliosvaldo é um apaixonado por animais exóticos que vive em uma cidade grande. De suas muitas aves e bichinhos legalmente possíveis de adquirir, Eli cuida de um pequena família de passarinhos-lóris, que resgatou há alguns anos. Certo dia, Zuzu, o integrante caçula dessa família, depois de um passeio pela vizinhança, começou a apresentar sinais de doença: febre alta, zumbidos de dor, falta de apetite, palidez e apatia para aprender novos truques, como era de costume. Será que era contagioso? Será que

havam o evenenado? Preocupado, Eli decidiu levá-los ao centro de controle de zoonoses local para receber atendimento veterinário.

Ao chegar ao centro, Eli foi informado de que o sistema de registros dos atendimentos e incidentes era feito manualmente, utilizando formulários em papel. Ele percebeu que o processo era lento e ineficiente. Além disso, Dra. Gilmara, a veterinária responsável, teve dificuldade em acessar informações anteriores sobre outros atendimentos ou campanhas de vacinação das quais tanto Zuzu quanto o restante da família havia participado. A situação se complicou ainda mais quando, ao voltar pra casa, Eliosvaldo encontrou Reco-Reco e Brilha-Brilha, irmãos de Zuzu, com marcas de ferida e sintomas de pânico, o que não havia sido a primeira vez. Eli suspeitou mais uma vez de seus vizinhos, mas não havia um registro consolidado desde a última vez que ele denunciara o problema no sistema, demonstrando a ineficiência e falta de centralização das informações no centro de controle de zoonoses.

Nesse sentido, propõe-se a prototipação de um sistema de informações para o contexto acima, sendo alimentado por um sistema de banco de dados.

1.2 Sobre o sistema de informação (SI)

a. Requisitos

R1: Controle de Animais

- O sistema deve gerenciar todos os animais atendidos ou monitorados pelo centro de zoonoses, obtendo-se por exemplo a quantidade cada animal por espécie. Cada animal deve estar associado a um conjunto de informações detalhadas, como espécie, raça, idade, peso, e status atual (tipos: Saudável, Em Recuperação, Em tratamento, Em Observação).
- Animais devem ser identificados por um código único e associados a um responsável (chamado “Guardião”). Assim, o sistema deve permitir o rastreamento da relação guardião-animal, como todos os animais que estão em tutela do determinado guardião.

R2: Registro de Incidentes

- O sistema deve permitir visualizar o registro de todos os incidentes que envolvam animais, como atropelamentos, maus-tratos, e **infestações**.
- Cada incidente deve ser documentado com detalhes como o tipo de incidente, data, localização, animais envolvidos, descrição do ocorrido, e se foi registrado por um agente de campo e quem foi esse agente.
- Deve-se conseguir visualizar a quantidade para cada tipo de incidente, e os distritos sanitários em que mais ocorrem, permitindo melhor gerenciamento e medidas práticas para os incidentes.

R3: Histórico de Maus-Tratos

- O sistema deve manter um registro detalhado de animais com histórico de maus-tratos, registrando intervenções anteriores e tratamentos recebidos. Deve ser possível identificar facilmente animais com histórico de maus-tratos ao inserir um novo incidente. Além disso, deve-se poder identificar quem são os guardiões desses animais com histórico de maus-tratos.

R4: Gestão de Campanhas

- O sistema deve gerenciar campanhas de saúde pública, como vacinação e castração, permitindo a associação de múltiplos animais a cada campanha.
- As campanhas devem ser registradas com informações como tipo de campanha, data de início e término, local (distrito sanitario), e agentes responsáveis.

R5: Controle de Atendimentos Veterinários

- O sistema deve gerenciar os atendimentos veterinários realizados no centro, armazenando informações detalhadas sobre o diagnóstico, tratamento, veterinário responsável e resultados de cada atendimento.
- Deve ser possível associar múltiplos atendimentos a um único animal, rastreando o histórico de saúde de animais, como **Zuzu e Reco-Reco**.
- Deve ser possível, ainda, visualizar os atendimentos realizados e suas informações básicas, em um intervalo específico de tempo, fornecendo uma visão geral dos atendimentos durante esse período.

R6: Inspeções Zoossanitárias

- O sistema deve controlar as inspeções realizadas em locais de risco, como áreas infestadas por pragas ou locais de alta incidência de zoonoses. Cada inspeção deve ser documentada com informações sobre a data, local, resultado da inspeção, e o agente responsável.

R7: Gestão de Agentes de Campo e Veterinários

- O sistema deve gerenciar as informações dos agentes de campo e veterinários, permitindo o rastreamento das atividades realizadas por cada um.
- Cada agente ou veterinário deve estar associado aos incidentes, atendimentos, e inspeções que realizaram e suas quantidades.

R8: Geração de Relatórios e Dashboards

1. o sistema deve ser capaz de gerar relatórios e/ou gráficos que informem:
 - 1.1. Detalhamento de incidentes registrados, incluindo o código do incidente, tipo de incidente, data e hora, descrição do ocorrido, agente responsável e códigos dos animais envolvidos;
 - 1.2. O desempenho dos agentes de campo em termos de incidentes registrados, segmentados pelos tipos de incidentes;
 - 1.3. O desempenho dos agentes de campo em termos de campanhas realizadas, segmentos pelos tipos de campanha;
 - 1.4. Identificação dos animais com registros de maus-tratos, acompanhados de seus históricos de intervenções e tratamentos, detalhando as ocorrências e atendimentos realizados
 - 1.5. Atendimentos veterinários realizados, mostrando um histórico completo dos atendimentos para cada animal;
 - 1.6. Distribuição dos atendimentos veterinários realizados por cada veterinário, segmentada por especialidade e tipo de incidente
 - 1.7. Análise temporal dos atendimentos veterinários realizados dentro de um período específico, segmentados pelo status (“resultado”) do atendimento (saúdável, em recuperação, em observação, etc)
 - 1.8. Participação dos veterinários nos tipos de campanhas realizadas nos distritos sanitários;

- 1.9. Análise do quantitativo da diversidade de incidentes por distrito sanitário:
- 1.10. Eficácia das inspeções zoossanitárias para cada distrito, considerando o status/resultado (resolvida, em andamento, em análise, a ser feita) das inspeções;

1.3 Delimitação do mini-mundo para o banco de dados

O banco de dados relacional para suportar as operações deste sistema de informação deve contar com entidades e atributos coerentes ao contexto apresentado. Abaixo, segue:

● Entidades

1. Entidade: Animal

- **Descrição:** Representa os animais atendidos ou monitorados pelo centro de zoonoses;
- **Atributos:**
 - **PK_id_animal:** Identificador único do animal. (inteiro, incremental, PK)
 - **no_animal:** Nome do animal. (string, 50 caracteres)
 - **tp_especie:** Espécie do animal, como "Cão", "Gato", "Macaco". (string, 30 caracteres)
 - **tp_raca:** Raça do animal. (string, 50 caracteres)
 - **dt_nascimento:** Data de nascimento do animal (DATETIME)
 - **qnt_peso:** Peso do animal em quilogramas. (decimal, 5.2)
 - **st_animal:** Status atual do animal (Resgatado, Tratado, Devolvido, Eutanasiado). (string, 20 caracteres)
 - **id_guardiao:** Referência ao responsável/guardiao pelo/do animal. (inteiro, FK)

2. Entidade: Incidente

- **Descrição:** Registra eventos adversos ou situações que envolvem animais ou pragas urbanas, como maus-tratos ou infestações.
- **Atributos:**

- **PK_id_incidente**: Identificador único do incidente. (inteiro, incremental, PK)
- **tp_incidente**: Tipo de incidente (Atropelamento, Maus-tratos, Infestação). (string, 50 caracteres)
- **dh_incidente**: Data e hora do incidente. (datetime)
- **ds_incidente**: Descrição detalhada do incidente. (text)
- **FK_id_animal**: Referência ao animal envolvido no incidente, se aplicável. (inteiro, FK)
- **FK_id_agente**: Referência ao agente que registrou o incidente. (inteiro, FK)

3. Entidade: Campanha

- **Descrição**: Gerencia campanhas de saúde pública, como campanhas de vacinação e castração.
- **Atributos**:
 - **PK_id_campanha**: Identificador único da campanha. (inteiro, incremental, PK)
 - **no_campanha**: Nome da campanha (Ex: "Vacinação Antirrábica 2024"). (string, 100 caracteres)
 - **tp_campanha**: Tipo de campanha (Vacinação, Castração, etc.). (string, 50 caracteres)
 - **dt_inicio**: Data de início da campanha. (date)
 - **dt_fim**: Data de término da campanha. (date)
 - **FK_id_agente**: Referência ao agente responsável pela campanha. (inteiro, FK)
 -

4. Entidade: Atendimento

- **Descrição**: Documenta os atendimentos veterinários prestados aos animais, incluindo diagnósticos e tratamentos.
- **Atributos**:
 - **PK_id_atendimento**: Identificador único do atendimento. (inteiro, incremental, PK)
 - **dh_atendime**: Data do atendimento. (datetime)
 - **ds_diagnostico**: Diagnóstico realizado no animal. (text)
 - **ds_tratamento**: Tratamento prescrito ou realizado. (text)

- **st_resultado**: Resultado do tratamento (Curado, Em Tratamento, Eutanasiado). (string, 50 caracteres)
- **FK_id_animal**: Referência ao animal atendido. (inteiro, FK)
- **FK_id_veterinario**: Referência ao veterinário que realizou o atendimento. (inteiro, FK)

5. Entidade: Inspeção Zoossanitaria

- **Descrição**: Armazena informações sobre inspeções zoossanitárias realizadas em áreas de risco.
- **Atributos**:
 - **PK_id_inspeção**: Identificador único da inspeção. (inteiro, incremental, PK)
 - **dh_inspecao**: Data da inspeção. (datetime)
 - **st_resultado_inspecao**: Resultado da inspeção (resolvida, em andamento, em análise, critica, muito critica). (string, 50 caracteres)
 - **ds_inspecao**: detalhes da inspeção (detalhes, como foi aplicado, e o que foi identificado) (text)
 - **FK_id_agente**: Referência ao agente que realizou a inspeção. (inteiro, FK)

6. Entidade: Agente

- **Descrição**: Registra informações sobre os agentes que atuam em campo realizando inspeções e registrando incidentes.
- **Atributos**:
 - **id_agente**: Identificador único do agente. (inteiro, incremental, PK)
 - **no_agente**: Nome completo do agente. (string, 100 caracteres)
 - **no_cargo**: Cargo ou função do agente (Ex: Inspetor, Veterinário). (string, 50 caracteres)
 - **nu_telefone**: Número de telefone do agente. (string, 15 caracteres)
 - **ds_email**: Email de contato do agente. (string, 50 caracteres)
 - **FK_id_distrito_sanitario**: referência ao distrito sanitário ao qual o agente faz parte(inteiro, FK)

7. Entidade: Veterinário

- **Descrição:** Armazena dados dos veterinários de referência que atuam no centro, aqueles que respondem pelos atendimentos dos animais e participam das campanhas.
- **Atributos:**
 - **PK_id_veterinario:** Identificador único do veterinário. (inteiro, incremental, PK)
 - **no_veterinario:** Nome completo do veterinário. (string, 100 caracteres)
 - **tp_especialidade:** Especialidade do veterinário (Ex: Cirurgião, Clínico Geral). (string, 50 caracteres)
 - **nu_telefone:** Número de telefone do veterinário. (string, 15 caracteres)
 - **ds_email:** Email de contato do veterinário. (string, 50 caracteres)
 - **cd_crmv:** Código do CRMV (Conselho Regional de Medicina Veterinária) do veterinário. (string, 10 caracteres, UNIQUE)

8. Entidade: Guardião

- **Descrição:** Representa o responsável por um ou mais animais,, que pode ser uma pessoa física ou uma organização.
- **Atributos:**
 - **PK_id_guardiao:** Código único do responsável (inteiro, incremental, PK)
 - **no_guardiao:** Nome do responsável (pessoa física ou organização) (string, 100 caracteres)
 - **tp_guardiao:** Tipo de responsável (Física, Jurídica) (string, 20 caracteres)
 - **nu_cpf_cnpj_guardiao:** Número do CPF (para pessoa física) ou CNPJ (para organização) (string, 14 caracteres)
 - **ds_distrito_sanitario_guardiao:** Endereço do responsável (string, 150 caracteres)
 - **nu_telefone:** Número de telefone do responsável (string, 15 caracteres)
 - **ds_email:** Email de contato do responsável (string, 50 caracteres)

9. Entidade: Distritos Sanitários

Descrição: Representa os distritos sanitários cobertos pelo sistema, incluindo as áreas de abrangência, meios de contato, e localização física do distrito. Cada

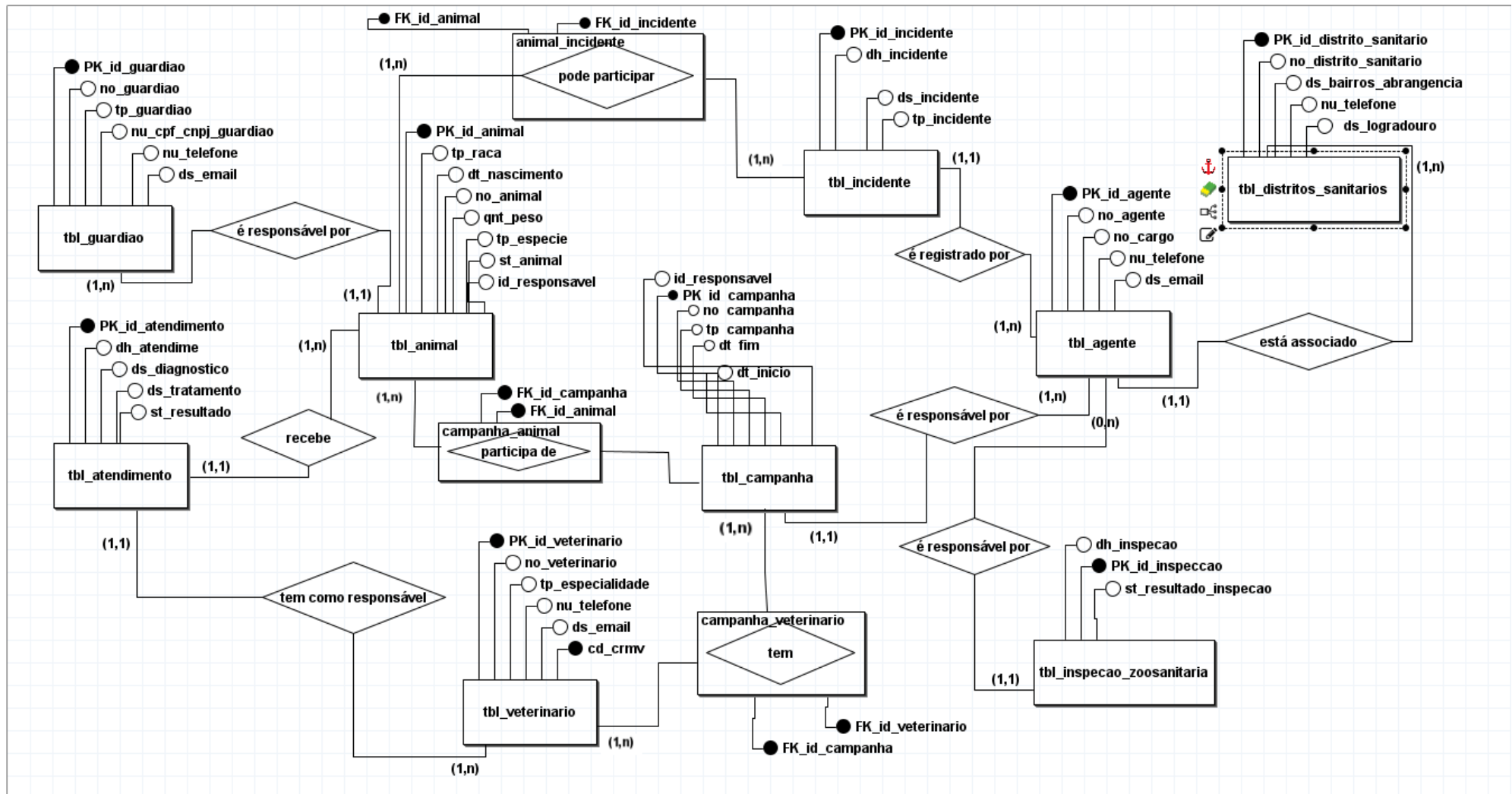
distrito sanitário pode estar associado a vários agentes que operam dentro dessa jurisdição.

Atributos:

- **PK_id_distrito_sanitario:** Código único do distrito sanitário (inteiro, incremental, PK).
- **no_distrito_sanitario:** Nome ou descrição do distrito sanitário (string, 100 caracteres).
- **ds_bairros_abrangencia:** Lista de bairros que estão sob a abrangência do distrito sanitário, armazenada como um array de strings (array de strings, cada string com até 100 caracteres).
- **nu_telefone:** Número de telefone de contato do distrito sanitário (string, 15 caracteres).
- **ds_logradouro:** Endereço completo ou parcial do distrito sanitário, incluindo rua, número, e demais informações relevantes para localização (string, 255 caracteres)

- Modelo Conceitual

[Clique aqui](#) para visualizar com mais definição e detalhes (colocar elementos no relacionamento)



● Relacionamentos

Relacionamento entre Animal e Incidente:

- **Tipo:** Muitos-para-muitos
- **Descrição:** Um animal pode estar envolvido em múltiplos incidentes. Por exemplo, Reco-Reco, um dos passarinhos-lóris de Eliosvaldo, pode ter registros de múltiplos incidentes de maus-tratos. Assim como o incidente pode estar associado a vários animais (Reco-Reco e Zuzu brigaram)

Relacionamento entre Animal e Atendimento:

- **Tipo:** Um-para-muitos
- **Descrição:** Um animal pode receber múltiplos atendimentos veterinários ao longo do tempo, mas aquele atendimento só pode estar associado a um animal.
- **Chaves Estrangeiras (FK):** O relacionamento está representado com a chave estrangeira `FK_id_animal` na tabela `tbl_atendimento`, referenciando `tbl_animal`.

Relacionamento entre Veterinário e Atendimento:

- **Tipo:** Um-para-muitos
- **Descrição:** Um veterinário pode ser responsável por realizar múltiplos atendimentos. Por exemplo, a Dra. Ana pode ter sido **responsável** por ter atendido vários animais, incluindo Reco-Reco. Mas um atendimento só tem um veterinário responsável.
- **Chaves Estrangeiras (FK):** A entidade `Atendimento` possui uma chave estrangeira `ID_Veterinario`, que referencia `ID_Veterinario` da entidade `Veterinário`.

Relacionamento entre Campanha e Veterinário:

- **Tipo:** Muitos-para-muitos (acaba gerando uma entidade associativa)
- **Descrição:** Uma campanha pode envolver vários veterinários, e um veterinário pode participar de várias campanhas.
- **Tabela Intermediária:** `Campanha_Veterinario` com as chaves estrangeiras `ID_Campanha` e `ID_Veterinario`.

Relacionamento entre Campanha e Animal:

- **Tipo:** Muitos-para-muitos (acaba gerando uma entidade associativa)
- **Descrição:** Um animal pode participar de várias campanhas, e uma campanha pode envolver vários animais.
- **Tabela Intermediária:** `Campanha_Animal` com as chaves estrangeiras `PK_ID_Campanha` e `PK_ID_Animal`.

Relacionamento entre Inspeção_Zoossanitaria e Agente:

- **Tipo:** Um-para-muitos
- **Descrição:** Um agente pode ser responsável por várias inspeções zoossanitárias, mas uma inspeção só tem 1 agente responsável;
- **Chaves Estrangeiras (FK):** A entidade `Inspeção_Zoossanitaria` possui uma chave estrangeira `PK_ID_Agente`, que referencia `ID_Agente` da entidade `Agente`.

Relacionamento entre Agente e Incidente:

- **Tipo:** Um-para-muitos
- **Descrição:** Um agente pode ser responsável por registrar vários incidentes, mas cada incidente é registrado por apenas um agente.
- **Chaves Estrangeiras (FK):** A entidade `Incidente` possui uma chave estrangeira `ID_Agente`, que referencia `ID_Agente` da entidade `Agente`.

Relacionamento entre Agente e Campanha:

- **Tipo:** Um-para-muitos
- **Descrição:** Um agente pode ser responsável por várias campanhas, mas cada campanha fica sob responsabilidade de apenas um agente.
- **Chaves Estrangeiras (FK):** A entidade `Campanha` possui uma chave estrangeira `FK_id_agente`, que referencia `id_agente` da entidade `Agente`.

Relacionamento entre Guardiã e Animal:

- **Tipo:** Um-para-Muitos
- **Descrição:** Um responsável pode ser responsável por vários animais, mas cada animal tem apenas um responsável.

- ## Relacionamento entre Agente e Distrito Sanitário

- **Modelo lógico**



2. Criando a estrutura relacional do SGBD

DDL (Data definition language)

- Criação das tabelas e definição de constraints
 - [Acesse aqui o arquivo de criação de tabela e constraints](#)
- Índices

Considerando as entidades/tabelas que mais aparecem nas queries e a frequência das referidas nas operações de agregação, agrupamento e filtragem, seleciona-se os seguintes índices abaixo para serem criados.

Query para criação dos Índices	Entidade(s) envolvida (s)	Objetivo
CREATE INDEX IDX_tbl_animal_tp_especie_st_animal ON tbl_animal (tp_especie, st_animal);	Animal	otimizar buscas por espécie e status do animal
CREATE INDEX IDX_tbl_animal_fk_guardiao_hash ON tbl_animal USING HASH (FK_id_guardiao);	Animal e Guardião	otimizar queries de agregação entre animal e guardião.
CREATE INDEX IDX_tbl_incidente_tp_incidente ON tbl_incidente (tp_incidente);	Incidente	Otimizar consultas que filtram por tipo de incidente
CREATE INDEX IDX_tbl_incidente_dh_incidente ON tbl_incidente (dh_incidente);	Incidente	consultas que ordenam ou filtram por data do incidente
CREATE INDEX IDX_tbl_agente_fk_distrito_sanitario ON tbl_agente (FK_id_distrito_sanitario);	Agente e Distritos Sanitários	otimizar queries que agregam distritos sanitários e agentes
CREATE INDEX IDX_tbl_atendimento_fk_animal ON tbl_atendimento (FK_id_animal);	Atendimentos e Animais	consultas sobre atendimentos associados a animais específicos

CREATE INDEX IDX_tbl_atendimento_fk_veterinario ON tbl_atendimento (FK_id_veterinario);	Atendimentos e Veterinário	Consultas de atendimento que filtram por veterinário responsável
CREATE INDEX IDX_tbl_distritos_sanitarios_no_distrito_sa nitario ON tbl_distritos_sanitarios (no_distrito_sanitario);	Distritos Sanitários	consultas que filtram por distrito sanitário

Por padrão, o PostgreSQL automaticamente utiliza o tipo de índice B-Tree. Apenas para a relação Animal e Guardião, especifiquei o uso do index hash, considerando que é o índice recomendado e útil para consultas que utilizam igualdade (=), como nas queries utilizadas para essas agregações.

3. OLTP: Interação cotidiana com o SI

Considerando as operações transacionais do dia a dia, onde as consultas são mais focadas em manter o funcionamento regular do sistema, temos:

3.1 Inserções

- [Acesse aqui](#) o arquivo dml para popular o banco de dados, simulando o que seriam os cadastros via SI dos novos registros cotidianos para o centro de zoonoses.

4. Protótipo e Código SQL Associado

Considerando os requisitos da aplicação, em termos de análises sistema-banco , teríamos:

Descrição da Query	Pergunta Analítica	Query	Justificativa da Query
Obter quantidade de animais por espécie	“Como é a distribuição das espécies de animais presentes no sistema?”	SELECT tp_especie, COUNT(*) AS num_animal FROM tbl_animal GROUP BY tp_especie ORDER BY num_animal DESC;	Associado ao RF1 , visando monitoramento e controle da dos animais.
Obter todos os tipos de incidentes registrados, mostrando a quantidade de cada tipo, agrupadamente	“Como é a distribuição dos tipos de incidentes registrados no sistema?”	SELECT tp_incidente, COUNT(*) AS quantidade FROM tbl_incidente GROUP BY tp_incidente;	Associado ao RF2 , em termos de controle de incidentes, seus tipos e quantidades no sistema
Obter o nome, a espécie e o status de todos os animais que são da espécie 'X' e que estão com o status 'Y'.	Quais animais da espécie 'x' estão atualmente em “x status”?	SELECT no_animal, tp_especie, st_animal FROM tbl_animal WHERE tp_especie = 'Pássaro Lóris" AND st_animal = 'Em tratamento';	Associado ao RF1 , visando monitoramento e controle da situação dos animais.
Procurar por status atual do animal de nome: [Nome_do_Animal] e a qual espécie ele pertence	Qual é o status atual e a espécie do animal chamado 'Nome'?	SELECT no_animal, tp_especie, st_animal FROM tbl_animal WHERE no_animal = 'Reco-Reco';	Associado ao RF1 , visando monitoramento e controle da situação dos animais.
Pesquisando pelo nome do guardião, obtém-se o(s) nome(s) do(s) animal(is), a espécie e o status do(s) animal(is)	“Quais são os animais sob a responsabilidade de um guardião específico, e qual é o status atual de cada animal que ele é tutor?”	SELECT a.no_animal, a.tp_especie, a.st_animal FROM tbl_animal a INNER JOIN tbl_guardiao g ON a.FK_id_guardiao = g.PK_id_guardiao WHERE g.no_guardiao = 'Eliosvaldo Souza';	Associado ao RF1 , visando monitoramento e controle da situação dos animais.

Obter o número de incidentes ocorridos em cada distrito sanitário	Qual é a distribuição dos incidentes registrados em cada distrito sanitário?	<pre>SELECT ds.no_distrito_sanitario, COUNT(*) AS num_incidentes FROM tbl_incidente i JOIN tbl_agente a ON i.FK_id_agente = a.PK_id_agente JOIN tbl_distritos_sanitarios ds ON a.FK_id_distrito_sanitario = ds.PK_id_distrito_sanitario GROUP BY ds.no_distrito_sanitario;</pre>	Associado ao RF2 , em termos de controle de incidentes, seus tipos, relação com os distritos sanitários e quantidades no sistema
Obter os animais que possuem um histórico de maus-tratos e listar quem são os seus guardiões, incluindo o nome e o código de cada guardião.	Quais animais possuem histórico de maus-tratos e quem são seus guardiões?	<pre>SELECT an.PK_id_animal, an.no_animal, g.PK_id_guardiao, g.no_guardiao FROM tbl_animal an JOIN tbl_guardiao g ON an.FK_id_guardiao = g.PK_id_guardiao WHERE an.PK_id_animal IN (SELECT ai.FK_id_animal FROM animal_incidente ai WHERE ai.FK_id_incidente IN (SELECT i.PK_id_incidente FROM tbl_incidente i WHERE i.tp_incidente ILIKE '%Maus-Tratos));</pre>	Associado ao RF3
Obter uma lista de todos os animais, suas espécies, o nome da campanha, e o local (distrito sanitário) onde a campanha foi realizada,	"Quais são os animais e suas respectivas espécies que participaram da campanha específica com ID 'x', onde essa campanha foi realizada?"	<pre>SELECT c.no_campanha,a.no_animal,a.tp_espe cie,ds.no_distrito_sanitario FROM tbl_campanha c JOIN campanha_animal ca ON c.pk_id_campanha = ca.fk_id_campanha JOIN tbl_animal a ON ca.fk_id_animal = a.pk_id_animal JOIN tbl_agente ag ON c.fk_id_agente = ag.pk_id_agente JOIN tbl_distritos_sanitarios ds ON ag.fk_id_distrito_sanitario = ds.pk_id_distrito_sanitario WHERE c.pk_id_campanha = '2' ORDER BY a.no_animal;</pre>	Associado ao RF4 , demonstrando a associação múltipla de animais a cada campanha
Obter lista de campanhas que o animal participou, incluindo o tipo de campanha e as datas de início	"Quais são as campanhas, tipos de campanha, e datas em que o animal específico com código/id 'y' e nome 'nome' participou?"	<pre>SELECT a.no_animal, c.no_campanha, c.tp_campanha, c.dt_inicio, c.dt_fim FROM tbl_animal a JOIN</pre>	Associado ao RF4 , demonstrando a associação múltipla de animais a cada campanha

		campanha_animal ca ON a.pk_id_animal = ca.fk_id_animal JOIN tbl_campanha c ON ca.fk_id_campanha = c.pk_id_campanha WHERE a.pk_id_animal = '2' ID do animal específico AND a.no_animal = 'Mimi' específico ORDER BY c.dt_inicio DESC;	
Obter registros de campanhas com informações como tipo de campanha, data de início e término, local (distrito sanitario que ocorreu), e agentes responsáveis.	Quais foram as campanhas realizadas, seus detalhes e os agentes responsáveis?	SELECT c.no_campanha, c.tp_campanha, c.dt_inicio, c.dt_fim, ds.no_distrito_sanitario AS local, a.no_agente AS agente_responsavel FROM tbl_campanha c JOIN tbl_agente a ON c.FK_id_agente = a.PK_id_agente JOIN tbl_distritos_sanitarios ds ON a.FK_id_distrito_sanitario = ds.PK_id_distrito_sanitario ORDER BY c.dt_inicio;	Associado ao RF4 , em termos de informação e gerenciamento de campanhas
Obter as inspeções realizadas, com detalhes sobre data, local, resultado e agente responsável	Quais inspeções zoossanitárias foram realizadas, quando, onde, e quais foram os resultados?	SELECT TO_CHAR(i.dh_inspecao,'DD/MM/YYYY HH24:MI:SS') AS data_inspecao, ds.no_distrito_sanitario AS local, i.st_resultado_inspecao, a.no_agente FROM tbl_inspecao_zoosanitaria i JOIN tbl_agente a ON i.FK_id_agente = a.PK_id_agente JOIN tbl_distritos_sanitarios ds ON a.FK_id_distrito_sanitario = ds.PK_id_distrito_sanitario ORDER BY i.dh_inspecao DESC;	Associado ao RF6 , considerando que sobre cada inspeção deve-se ter informações sobre a data, local, resultado da inspeção, e o agente responsável.
Obter as inspeções registradas por cada agente	Quais campanhas foram registradas por um agente específico?	SELECT TO_CHAR(i.dh_inspecao, 'DD/MM/YYYY HH24:MI:SS') AS data_inspecao, i.st_resultado_inspecao, ds.no_distrito_sanitario AS local FROM tbl_inspecao_zoosanitaria i JOIN tbl_agente a ON i.FK_id_agente = a.PK_id_agente JOIN tbl_distritos_sanitarios ds ON a.FK_id_distrito_sanitario = ds.PK_id_distrito_sanitario WHERE a.no_agente = 'Roberto Fernandes' ORDER BY i.dh_inspecao DESC;	Associado ao RF7

Obter os incidentes registradas por cada agente	Quais incidentes foram registradas por um agente específico?	<pre>SELECT TO_CHAR(i.dh_incidente, 'DD/MM/YYYY HH24:MI:SS') AS data_incidente, i.tp_incidente, i.ds_incidente FROM tbl_incidente i JOIN tbl_agente a ON i.FK_id_agente = a.PK_id_agente WHERE a.no_agente = 'Roberto Fernandes' ORDER BY i.dh_incidente DESC;</pre>	Associado ao REFZ
Obter os atendimentos que cada veterinário realizou, através do seu código único, e informações gerais sobre o atendimento	Quais atendimentos foram realizados pelo veterinário com código único 1, incluindo diagnósticos, tratamentos e resultados?	<pre>SELECT TO_CHAR(a.dh_atendimento,'DD/MM/ YYYY HH24:MI:SS') AS data_atendimento,a.ds_diagnostico,a. ds_tratamento,a.st_resultado,an.no_an imal, an.tp_especie, an.pk_id_animal FROM tbl_atendimento a JOIN tbl_animal an ON a.FK_id_animal = an.PK_id_animal JOIN tbl_veterinario v ON a.FK_id_veterinario = v.PK_id_veterinario WHERE v.PK_id_veterinario = 1 ORDER BY a.dh_atendimento DESC;</pre>	Associado ao REFZ

4.1 Geração de relatórios e dashboards

É sabido ainda que o centro de zoonoses precisa gerar relatórios quinzenais detalhados para auditorias governamentais. Esses relatórios devem incluir dados agregados sobre o número total de incidentes de maus-tratos, o desempenho de agentes de campo, e o histórico de saúde de todos os animais, baseando-se no [RF8](#). Visando não impactar a performance e o desempenho do sistema, ao invés de executar essas consultas diretamente nas tabela base, armazenaremos os resultados desses relatórios em views "tabeladas" (materialized views), e utilizaremos stored procedures periodicamente, para

garantir que os dados estejam sempre atualizados sem sobrecarregar o sistema durante a geração dos relatórios.

Pergunta Analítica	Query	Tipo de visualização do resultado
“Quais são os detalhes dos incidentes registrados, incluindo os animais envolvidos e o agente responsável?”	<pre> SELECT i.PK_id_incidente,i.tp_incidente,i.dh_incidente,i.ds_ incidente,a.no_agente,STRING_AGG(CAST(an.PK_ id_animal AS VARCHAR),',') AS codigos_animais_envolvidos FROM tbl_incidente i JOIN tbl_agente a ON i.FK_id_agente=a.PK_id_agente LEFT JOIN animal_incidente ai ON i.PK_id_incidente=ai.FK_id_incidente LEFT JOIN tbl_animal an ON ai.FK_id_animal=an.PK_id_animal GROUP BY i.PK_id_incidente,i.tp_incidente,i.dh_incidente,i.ds_ incidente,a.no_agente; </pre>	Nesse caso, trata-se de uma análise mais textual, funciona melhor no formato de relatório
“Agentes e seus números de registros de incidentes, segmentados pelos tipos de incidente mais frequentes”	<pre> SELECT i.tp_incidente,a.no_agente,COUNT(*) AS num_registrados FROM tbl_incidente i JOIN tbl_agente a ON i.FK_id_agente=a.PK_id_agente WHERE i.tp_incidente IN (SELECT tp_incidente FROM tbl_incidente GROUP BY tp_incidente ORDER BY COUNT(*) DESC LIMIT 2) GROUP BY i.tp_incidente,a.no_agente ORDER BY num_registrados DESC; </pre>	Apoiaria um gráfico de barras, em que Eixo X (Horizontal): Nomes dos agentes; Eixo Y (Vertical): Número de registros de incidentes e haveria uma Segmentação (Cor das Barras) para os tipos de incidentes
“Agentes e seu número de campanhas, segmentados pelos tipos de campanha mais frequentes?”	<pre> SELECT a.no_agente, c.tp_campanha, COUNT(c.PK_id_campanha) AS num_campanhas FROM tbl_campanha c JOIN tbl_agente a ON c.FK_id_agente = a.PK_id_agente GROUP BY a.no_agente, c.tp_campanha ORDER BY COUNT(c.PK_id_campanha) DESC, a.no_agente; </pre>	Apoiaria um gráfico de barras, em que Eixo X (Horizontal): Nomes dos agentes; Eixo Y (Vertical): Número de campanhas realizadas e haveria uma Segmentação (Cor das Barras) para os tipos de campanhas?
	SELECT	

<p>“Quais são os detalhes dos animais com registros de maus-tratos e qual o histórico completo de intervenções e tratamentos realizados?”</p>	<pre> an.PK_id_animal,an.no_animal,an.tp_especie,an.dt _nascimento,an.qnt_peso,an.st_animal, (SELECT STRING_AGG(i.tp_incidente ' em ' TO_CHAR(i.dh_incidente,'YYYY-MM-DD') ':' i.ds_incidente,',')) FROM tbl_incidente i JOIN animal_incidente ai ON i.PK_id_incidente=ai.FK_id_incidente WHERE ai.FK_id_animal=an.PK_id_animal AND i.tp_incidente ILIKE '%Maus-Tratos%' AS historico_maus_tratos, (SELECT STRING_AGG(TO_CHAR(a.dh_atendimento,'YYYY- MM-DD') ':' a.ds_diagnostico ' - ' a.ds_tratamento ' (' a.st_resultado ');') FROM tbl_atendimento a WHERE a.FK_id_animal=an.PK_id_animal) AS historico_tratamentos FROM tbl_animal an WHERE an.PK_id_animal IN (SELECT ai.FK_id_animal FROM animal_incidente ai JOIN tbl_incidente i ON ai.FK_id_incidente=i.PK_id_incidente WHERE i.tp_incidente ILIKE '%Maus-Tratos%') ORDER BY an.PK_id_animal; </pre>	<p>Nesse caso, trata-se de uma análise mais textual, funcionando melhor no formato de relatório</p>
<p>Qual é o histórico detalhado dos atendimentos veterinários para um animal específico, incluindo o diagnóstico, tratamento e resultados?</p>	<pre> SELECT a.dh_atendimento, a.PK_id_atendimento, a.ds_diagnostico, a.ds_tratamento, a.st_resultado, an.tp_especie, an.no_animal, AGE(an.dt_nascimento)::TEXT AS idade_animal, v.no_veterinario, v.tp_especialidade FROM tbl_atendimento a JOIN tbl_animal an ON a.FK_id_animal = an.PK_id_animal JOIN tbl_veterinario v ON a.FK_id_veterinario = v.PK_id_veterinario WHERE PK_id_animal = '29' ORDER BY a.dh_atendimento DESC; </pre>	<p>Nesse caso, trata-se de uma análise mais textual, funciona melhor no formato de relatório</p>
<p>“ Qual é a distribuição dos atendimentos veterinários realizados por cada veterinário, segmentada por especialidade e tipo de incidente?”</p>	<pre> SELECT v.no_veterinario,v.tp_especialidade,i.tp_incidente,C OUNT(a.PK_id_atendimento) AS num_atendimentos FROM tbl_atendimento a JOIN tbl_animal an ON a.FK_id_animal = an.PK_id_animal JOIN tbl_veterinario v ON a.FK_id_veterinario = v.PK_id_veterinario JOIN animal_incidente ai ON an.PK_id_animal = ai.FK_id_animal JOIN tbl_incidente i ON ai.FK_id_incidente = i.PK_id_incidente </pre>	<p>Apoiaria um gráfico de barras, em que Eixo X (Horizontal): Nomes dos veterinários; Eixo Y (Vertical): número dos atendimentos e incidentes e haveria uma Segmentação (Cor das Barras) para os tipos de incidentes. Usando uma legenda/cor mais específica, pode-se indicar as especialidades dos veterinários</p>

	<p>GROUP BY v.no_veterinario,v.tp_especialidade,i.tp_incidente ORDER BY v.no_veterinario,i.tp_incidente;</p>	
<p>“Quais foram os atendimentos veterinários realizados dentro de um período específico, segmentando por status?”</p>	<p>SELECT TO_CHAR(a.dh_atendimento, 'YYYY-MM-DD') AS data_atendimento, v.no_veterinario, a.st_resultado, COUNT(*) AS num_atendimentos FROM tbl_atendimento a JOIN tbl_veterinario v ON a.FK_id_veterinario = v.PK_id_veterinario WHERE a.dh_atendimento BETWEEN TO_DATE('01/01/2024', 'DD/MM/YYYY') AND TO_DATE('31/12/2024', 'DD/MM/YYYY') GROUP BY TO_CHAR(a.dh_atendimento, 'YYYY-MM-DD'), v.no_veterinario, a.st_resultado ORDER BY data_atendimento;</p>	<p>Apoiaria um gráfico de linhas, em que Eixo X (Horizontal): datas; Eixo Y (Vertical): número dos atendimentos e haveria uma Segmentação (Cor das linhas): cada linha no gráfico representa um resultado de atendimento diferente</p>
<p>“Qual é a distribuição dos tipos de incidentes em cada distrito sanitário, e quais distritos apresentam maior diversidade e quantidade de incidentes?”</p>	<p>SELECT ds.no_distrito_sanitario, i.tp_incidente, COUNT(i.PK_id_incidente) AS num_incidentes FROM tbl_incidente i JOIN tbl_agente a ON i.FK_id_agente = a.PK_id_agente JOIN tbl_distritos_sanitarios ds ON a.FK_id_distrito_sanitario = ds.PK_id_distrito_sanitario GROUP BY ds.no_distrito_sanitario, i.tp_incidente ORDER BY ds.no_distrito_sanitario, num_incidentes DESC;</p>	<p>Apoiaria um gráfico de barras empilhadas (horizontal), em que Eixo X (Horizontal): Nome dos distritos sanitários ; Eixo Y (Vertical): Número de incidentes. Usando uma legenda/cor mais específica, Cada cor representará um tipo de incidente diferente. Compondo cada “barra” dos distritos sanitários</p>
<p>“Como está distribuída a participação dos veterinários em diferentes tipos de campanhas, e quais veterinários estão mais envolvidos em cada tipo de campanha?”</p>	<p>SELECT v.no_veterinario, c.tp_campanha, COUNT(cv.fk_id_campanha) AS num_campanhas FROM campanha_veterinario cv JOIN tbl_campanha c ON cv.fk_id_campanha = c.pk_id_campanha JOIN tbl_veterinario v ON cv.fk_id_veterinario = v.pk_id_veterinario GROUP BY v.no_veterinario, c.tp_campanha ORDER BY v.no_veterinario, num_campanhas DESC;</p>	<p>Apoiaria um gráfico de barras onde o eixo x (Horizontal): Nomes dos veterinários e eixo Y (Vertical): número de campanhas realizada, com uma segmentação (Cores das Barras), em que Cada cor representaria um tipo de campanha</p>
<p>“Qual é a distribuição e o status de eficácia das inspeções zoossanitárias para cada distrito sanitário específico, com base nos estados de resolução</p>	<p>SELECT ds.no_distrito_sanitario, i.st_resultado_inspecao, COUNT(i.PK_id_inspeção) AS num_inspecoes FROM tbl_inspecao_zoosanitaria i JOIN tbl_agente a ON i.FK_id_agente =</p>	<p>Apoiaria um gráfico de pizza, onde os segmentos do gráfico de pizza representariam um tipo de status de inspeção e o tamanho de cada fatia</p>

(resolvida, em andamento, em análise, ou a ser feita)”	<pre> a.PK_id_agente JOIN tbl_distritos_sanitarios ds ON a.fk_id_distrito_sanitario = ds.pk_id_distrito_sanitario WHERE ds.pk_id_distrito_sanitario = '2' GROUP BY ds.no_distrito_sanitario, i.st_resultado_inspecao ORDER BY i.st_resultado_inspecao; </pre>	seria proporcional ao número de inspeções que têm aquele status
--	---	---

● Views e Stored Procedures

Acesse aqui [o link](#) para as stored procedures a serem realizadas

5. Política e rotina de backup

Considerando o contexto do centro de zoonoses, em que:

- Infraestrutura não tem robustez adequada ainda, visto que se trata de um projeto de início/exemplo de centro de zoonoses, e portanto, não conta com locais apropriados, com controle de fatores ambientais sensíveis, como umidade, temperatura, etc., e com acesso restrito a pessoas autorizadas para acessar;
- Localização do centro de zoonoses é próxima à praia, o que facilita corrosão e deterioração da estrutura física de possíveis servidores de hospedagem para backup;
- Além disso, trata-se de uma região em que notifica-se queda energética de modo mensal;

Tem-se a seguinte proposta para a rotina de backup para as tabelas tbl_animal, tbl_guardiao, tbl_atendimento, tbl_veterinario e tbl_agente, consideradas críticas à nível de importância para o funcionamento do sistema:

Tipo de Backup	Frequência de	Tempo de	Local de
----------------	---------------	----------	----------

	Execução	Retenção	Armazenamento
Full	A cada 7 dias	4 semanas	Servidor em nuvem
Incremental	Diariamente [no final do expediente padrão, depois das 20:00]	3 semanas	Servidor em nuvem
Diferencial	Semanalmente [Nas quarta-feiras, às 19:00, pois são os momentos identificados com menos alterações de dados]	3 semanas	Servidor em nuvem

Para as demais tabelas, consideradas não críticas, tem-se a seguinte proposta de rotina de backup:

Tipo de Backup	Frequência de Execução	Tempo de Retenção	Local de Armazenamento
Full	A cada 15 dias	3 semanas	Servidor em nuvem
Incremental	Dia sim dia não [no final do expediente padrão, depois das 20:00]	2 semanas	Servidor em nuvem
Diferencial	De 10 em 10 dias [Nas quarta-feiras, às 19:00, pois são os momentos identificados com menos alterações de dados]	2 semanas	Servidor em nuvem

Em termos de políticas e diretrizes para o backup dos dados nesse centro de zoonoses, temos:

Responsável :	Liliana, administradora de TI do Centro de Zoonoses
----------------------	---

Aprovado por:	Marcelo, diretor-geral do Centro de Zoonoses
Localização de armazenamento:	Serviço externo de hospedagem em nuvem
Data da Aprovação :	_____
Data de revisão:	Anualmente

Para descarte da mídia de backup, tem-se que o time administrador de tecnologia deverá configurar políticas de retenção coerentes com o tempo estabelecido acima, em relação às classes de armazenamento em nuvem que vão conter os backups. Essas políticas devem incluir regras de ciclo de vida, que determinam quando os backups devem ser movidos para classes de armazenamento de longo prazo, adequadas para dados que raramente são acessados, mas ainda precisam ser preservados. Além disso, essas regras devem prever o descarte seguro dos backups quando eles não forem mais necessários, garantindo que os dados sejam devidamente excluídos ou destruídos. A escolha em nuvem permite também um controle mais discriminado de acesso do armazenamento e sua privacidade, uma vez que somente administradores e pessoas previamente discriminadas terão acesso a esses dados.