

# Resumo das Regras do Banco de Dados

## Sistema de Adoção de Pets Resgatados

Equipe de Desenvolvimento

27 de agosto de 2025

### Conteúdo

<b>1</b>	<b>Regras Gerais</b>	<b>2</b>
<b>2</b>	<b>Inserts</b>	<b>2</b>
<b>3</b>	<b>Updates</b>	<b>2</b>
<b>4</b>	<b>Deletes</b>	<b>4</b>
<b>5</b>	<b>Triggers Principais</b>	<b>4</b>
<b>6</b>	<b>Constraints e Regras de Integridade</b>	<b>4</b>

# 1 Regras Gerais

- O sistema segue o princípio de consistência: campos críticos são gerenciados por **triggers**, e não podem ser alterados manualmente.
- Usuários recém-criados assumem, por padrão, o perfil de **adotante**.
- Restrições de integridade impedem duplicidade de solicitações de adoção em aberto para o mesmo usuário e o mesmo pet.
- As triggers garantem coerência entre tabelas relacionadas (ex.: adoção aprovada altera status do pet para **adotado** e encerra hospedagem ativa).

# 2 Inserts

- Inserir usuários com perfil padrão **adotante** se não especificado.
- Inserir pets exige atributos básicos: nome, sexo, porte, data de nascimento, status inicial **disponivel**.
- Inserir lar temporário requer um **responsavel\_id** válido (usuário existente).
- Inserir uma adoção marca o status inicial como **pendente** e registra **solicitado\_em**.
- Inserir respostas no formulário atualiza automaticamente o campo **formulario\_respondido** da tabela **adocao**, se todas as obrigatórias estiverem preenchidas.

# 3 Updates

## Usuário

- Permite atualizar blocos separados:
  - Endereço (cep, cidade, estado, bairro, etc.).
  - Informações pessoais (nome, e-mail, telefone, celular, cpf).
  - Senha (via id ou via e-mail).
  - Update completo (admin) permite alterar tipo de usuário e se está ativo.

## Adoções

- Admin pode mudar **situacao** (**pendente**, **aprovada**, **rejeitada**), sempre registrando data em **atualizado\_em**.

- Campos `usuario_id`, `pet_id`, `solicitado_em` não podem ser alterados após inserção.
- Se a adoção for aprovada, triggers garantem:
  - Status do pet passa para `adotado`.
  - Hospedagem ativa do pet é encerrada.
  - O campo `formulario_respondido` deve estar = 1, senão a aprovação falha.

## Lar Temporário

- Admin pode alterar qualquer informação (responsável, descrição, limite de pets e endereço).
- O limite de pets não pode ser inferior ao número de animais já ativos no lar.
- Existe opção de copiar endereço automaticamente do usuário responsável.

## Pet

- Admin pode atualizar qualquer dado do pet (nome, descrição, porte, saúde etc.).
- O campo `status` só pode ser alterado manualmente para `disponivel` ou `resgatado`.
- O status `adotado` é atualizado exclusivamente pela trigger de adoção.

## Lar Temporário Pet

- Atualizar `data_saida` torna o registro automaticamente inativo.
- Atualizar `ativo = 0` preenche `data_saida` se não informada.
- Atualizar `ativo = 1` zera `data_saida` e valida exclusividade e capacidade.
- Trocas de lar ou pet disparam validações de capacidade e unicidade.

## Respostas de Formulário

- Usuário pode editar suas respostas; cada update registra `ultima_alteracao = NOW()`.
- A trigger revalida automaticamente se todas obrigatórias foram respondidas, mantendo `adocao.formulario_respondido`.

## 4 Deletes

- Apenas administradores podem executar exclusões.
- Permissões de exclusão:
  - Usuários.
  - Pets.
  - Lares temporários.
  - Registros de hospedagem (`lar_temporario_pet`).
  - Adoções que ainda não tenham sido aprovadas.
- Se a adoção estiver **aprovada**, a exclusão é bloqueada.

## 5 Triggers Principais

- **trg\_status\_pet**: ao aprovar/rejeitar adoção, atualiza status do pet e hospedagem.
- **trg\_capacidade\_lar\_pet**: impede ultrapassar limite de capacidade de um lar temporário e impede que um pet esteja em dois lares ativos simultaneamente.
- **Triggers de coerência em lar\_temporario\_pet**: garantem que `ativo` e `data_saida` sempre estejam sincronizados.
- **Triggers de formulário**: carimbam `ultima_alteracao`, atualizam `formulario_respondido` e impedem aprovação de adoção sem questionário completo.

## 6 Constraints e Regras de Integridade

- Constraint garante que não haja duas adoções em aberto (pendente) para o mesmo usuário e pet.
- Constraints de enumeração asseguram que campos como **status** e **situacao** aceitem apenas valores válidos.
- O campo `tipo_usuario` possui valor padrão `adotante`.