

Aqui pode ser inserida
uma figura ou logo que
caracterize o software a
ser desenvolvido

Adoção de Árvore: Sistema de controle

Documentação do Projeto

Versão 0.1

Elaborado por: Lucas Pasin

Matheus Salgado

Rafael Rees

Curso: Engenharia de Computação
Disciplina: BANCO DE DADOS
Professora: Fabrícia Damando Santos

Guaíba / RS
UERGS
2024

SUMÁRIO (a ser alterado)

1. INTRODUÇÃO – APRESENTAÇÃO DO PROBLEMA	3
2. SOLUÇÃO – Software a ser desenvolvido	3
3. LEVANTAMENTO DE REQUISITOS	3
3.1 Requisitos Funcionais	3
3.2 Requisitos Não-Funcionais	3
3.3 Regras de Negócio	3
4. MODELAGEM COMPORTAMENTAL	4
4.1 Diagramas de caso de uso	4
5. MODELAGEM DO BANCO DE DADOS	4
5.1 Diagrama ER	4
5.2 Código de criação das tabelas	4
5.3 Print das tabelas no MySQL	4
5.4 Código das inserções	5
5.5 Código das consultas solicitadas	5
6. TELAS DESENVOLVIDAS (N3)	5
7. IMPLEMENTAÇÃO (N3)	5
8. REFERÊNCIAS BIBLIOGRÁFICAS	5

1. INTRODUÇÃO – APRESENTAÇÃO DO PROBLEMA

Muitas espécies emblemáticas da Mata Atlântica foram classificadas como espécies ameaçadas de extinção, chegando a uma taxa de desmatamento de 221 árvores/minuto [1]

Devido a esse problema crítico, a empresa <empresa fictícia> decidiu lutar contra esta ameaça. Gerando um projeto de Adoção de Árvore, onde qualquer pessoa poderia contatá-la e pedir para plantar uma árvore em seu nome.

Porém, essa empresa encontrou problemas em seu gerenciamento, requisitando ajuda para desenvolver um sistema para ajudá-los a gerenciar melhor o projeto.

2. SOLUÇÃO – Software a ser desenvolvido

AdotaArvore TM – logo

A solução para o problema proposto é o desenvolvimento de um sistema de gerenciamento, o qual auxiliará a empresa a controlar e monitorar as informações e as suas atividades neste cenário.

3. LEVANTAMENTO DE REQUISITOS

A empresa vai precisar cadastrar os clientes no sistema, com informações de nome, endereço, CPF e número de telefone.

As árvores plantadas devem ser cadastradas, uma árvore, vai possuir informações como nome da pessoa que adotou, coordenadas, tipo de árvore, entre mais. Para buscar a árvore, será utilizado um ID próprio, o mesmo vale para o cliente. Será necessário também ver quais brotos no software estão disponíveis para serem plantados, possuindo as mesmas informações que a árvore, porém, quando o broto é plantado, é necessário colocar as coordenadas de onde foi plantado e a identificação do cliente que adotou ele.

Também devemos cadastrar as espécies de árvores disponíveis, e em que local cada espécie pode ser plantada, além da expectativa de vida de cada uma das espécies.

Será necessário ver qual funcionário plantou a árvore, pois eles recebem uma comissão por árvore plantada, então deveremos ver qual funcionário plantou qual árvore e quanto ele recebeu dependendo da espécie da árvore.

O cliente pode visitar o local da árvore, porém, dependendo do local onde foi plantando deverá requisitar uma permissão, como em caso de reservas ecológicas.

Apenas um funcionário que está próximo do local ao qual foi pedido a adoção poderá plantar, caso não tenham funcionários próximos, será pedido uma transferência de funcionário, que deverá ter uma comissão extra.

O local deverá ter o bioma dele, e o seu endereço

Gostaríamos que tivesse uma hud para manipular e ver as informações, para facilitar o uso do software.

A doação poderá ser feita por pix, cartão de débito, cartão de crédito ou boleto.

A importância do protocolo se deve ao fato de que quando um cliente pedir uma árvore gere um protocolo possuindo as seguintes informações: id protocolo, data de criação do protocolo, o id do funcionário e o id da árvore. Um protocolo é necessário para o acompanhamento dessa árvore e uma melhor organização a nível de gestão e gerenciamento.

3.1 Requisitos Funcionais

*endereço = informações sobre cidade, bairro, número, complemento e rua e a uf.

RF01 – Cadastro do cliente no sistema, com informações de nome, endereço, CPF ou CNPJ , número de telefone, ID do cliente, Data de nascimento.

RF02 – Cadastrar as árvores plantadas no sistema, com informações de id da pessoa que adotou, coordenadas, espécie da árvore, um ID próprio e o id do local que foi plantada.

RF03 – Cadastrar as espécies de árvores disponíveis, com informações de nome da espécie, expectativa de vida e uma breve descrição.

RF04 - Será necessário ver qual funcionário plantou a árvore. Para fazer o cadastro deverá possuir as seguintes informações: Nome, CPF, endereço, ID, data de nascimento.

RF05 - Cadastro do Bioma, contendo informações sobre seu ID, nome do bioma e descrição.

RF06 - Cadastro de UF contendo informações do nome do estado, a sigla e seu ID.

RF07 - Permitir a criação de protocolos, com informação sobre seu id , data de criação , data prevista para plantar, id do funcionário, id árvore e se ele foi finalizado ou não .

RF08 - Cadastrar os locais que as árvores são plantadas, contendo informações sobre seu ID, nome e endereço.

RF09 - Ver quais espécies estão em quais biomas.

3.2 Requisitos Não-Funcionais

RNF01 – Gostaríamos que tivesse uma hud para manipular e ver as informações, para facilitar o uso do software.

RNF02 – Será necessário também ver no software quais brotos estão disponíveis para serem plantados.

3.3 Regras de Negócio

RN01 – O cliente pode visitar o local da árvore, porém, dependendo do local onde foi plantando deverá requisitar uma permissão, como em caso de reservas ecológicas.

RN02 – Apenas um funcionário que está próximo do local ao qual foi pedido a adoção poderá plantar, caso não tenham funcionários próximos, será pedido uma transferência de funcionário, que deverá ter uma comissão extra.

RN03 – A doação poderá ser feita por pix, cartão de débito, cartão de crédito ou boleto.

RN04- Funcionários recebem comissão de diferentes valores conforme a espécie da árvore que foi plantada.

4. MODELAGEM COMPORTAMENTAL

4.1 Diagramas de caso de uso

Atores:

Cliente: A pessoa que acessa o site da empresa para adotar uma árvore.

Administrador: O responsável por gerenciar o sistema e verificar os pedidos de plantio.

Sistema de Plantio: Serviço externo que realiza o plantio das árvores.

Casos de Uso:

Visualizar Árvores Disponíveis: Cliente visualiza as árvores disponíveis para adoção.

Selecionar Árvore para Adoção: Cliente escolhe uma árvore para adotar.

Preencher Protocolo de Adoção: Cliente preenche um formulário com informações necessárias para a adoção.

Enviar Protocolo de Adoção: Cliente envia o protocolo preenchido.

Confirmar Adoção: Sistema confirma a adoção e gera uma solicitação de plantio.

Solicitar Plantio da Árvore: Sistema envia uma request ao Sistema de Plantio para realizar o plantio.

Gerenciar Solicitações de Adoção: Administrador visualiza e gerencia as solicitações de adoção e plantio.

Atualizar Status do Plantio: Sistema de Plantio atualiza o status do plantio da árvore.

Diagrama de Caso de Uso

Ator: Cliente

Visualizar Árvores Disponíveis

Selecionar Árvore para Adoção

Preencher Protocolo de Adoção

Enviar Protocolo de Adoção

Ator: Administrador

Gerenciar Solicitações de Adoção

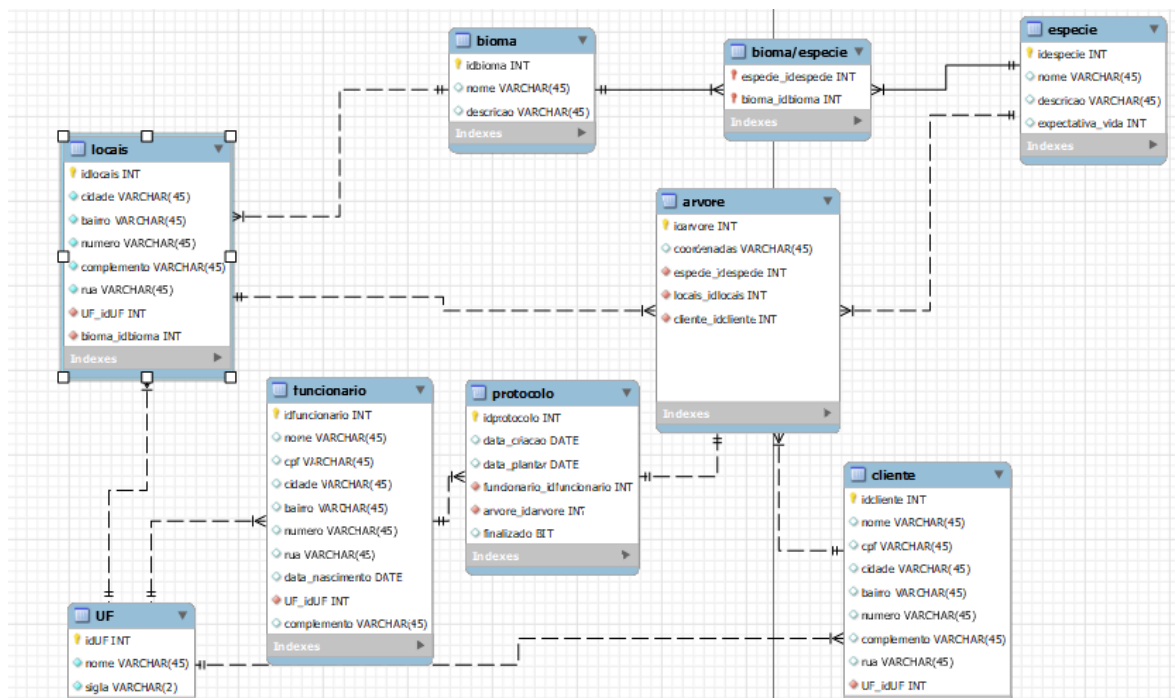
Ator: Sistema de Plantio

Solicitar Plantio da Árvore

Atualizar Status do Plantio

5. MODELAGEM DO BANCO DE DADOS

5.1 Diagrama ER



5.2 Código de criação das tabelas


```
CREATE TABLE uf (  
    id INTEGER UNSIGNED NOT NULL AUTO_INCREMENT,  
        nome VARCHAR(40) NOT NULL,  
    sigla VARCHAR(2) NOT NULL,  
    PRIMARY KEY (id)  
)
```

```
CREATE TABLE bioma (  
    id INTEGER UNSIGNED NOT NULL AUTO_INCREMENT,  
        nome VARCHAR(40) NOT NULL,  
    descricao VARCHAR(45) NOT NULL,  
    PRIMARY KEY (id)  
)
```

```
CREATE TABLE especie (  
    id INTEGER UNSIGNED NOT NULL AUTO_INCREMENT,  
        nome VARCHAR(40) NOT NULL,  
    descricao VARCHAR(45) NOT NULL,  
    expectativa_vida INTEGER UNSIGNED NOT NULL,  
    PRIMARY KEY (id)  
)
```

```
CREATE TABLE bioma_especie (  
    id_especie INTEGER UNSIGNED NOT NULL,  
    id_bioma INTEGER UNSIGNED NOT NULL,  
    CONSTRAINT id_especie FOREIGN KEY id_especie (id_especie)  
        REFERENCES especie (id)  
        ON DELETE RESTRICT  
        ON UPDATE RESTRICT,  
    CONSTRAINT id_bioma FOREIGN KEY id_bioma (id_bioma)  
        REFERENCES bioma (id)
```

```
        ON DELETE RESTRICT
        ON UPDATE RESTRICT
    )
```

```
CREATE TABLE locais (
    id INTEGER UNSIGNED NOT NULL AUTO_INCREMENT,
    cidade VARCHAR(40) NOT NULL,
    bairro VARCHAR(40) NOT NULL,
    rua VARCHAR(40) NOT NULL,
    numero INTEGER UNSIGNED NOT NULL,
    complemento VARCHAR(40) NOT NULL,
    id_UF INTEGER UNSIGNED NOT NULL,
    id_bioma INTEGER UNSIGNED NOT NULL,
    PRIMARY KEY (id),
    CONSTRAINT locais_UF FOREIGN KEY id_UF (id_UF)
        REFERENCES uf (id)
        ON DELETE RESTRICT
        ON UPDATE RESTRICT,
    CONSTRAINT locais_bioma FOREIGN KEY id_bioma (id_bioma)
        REFERENCES bioma (id)
        ON DELETE RESTRICT
        ON UPDATE RESTRICT
)
```

```
CREATE TABLE funcionario (
    id INTEGER UNSIGNED NOT NULL AUTO_INCREMENT,
    nome VARCHAR(40) NOT NULL,
    cpf VARCHAR(11) NOT NULL UNIQUE,
    cidade VARCHAR(40) NOT NULL,
    bairro VARCHAR(40) NOT NULL,
    rua VARCHAR(40) NOT NULL,
```

```
numero INTEGER UNSIGNED NOT NULL,  
complemento VARCHAR(40) NOT NULL,  
data_nascimento DATE NOT NULL,  
id_UF INTEGER UNSIGNED NOT NULL,  
PRIMARY KEY (id),  
CONSTRAINT funcinario_UF FOREIGN KEY id_UF (id_UF)  
REFERENCES uf (id)  
ON DELETE RESTRICT  
ON UPDATE RESTRICT  
)
```

```
CREATE TABLE cliente (  
id INTEGER UNSIGNED NOT NULL AUTO_INCREMENT,  
nome VARCHAR(40) NOT NULL,  
cpf VARCHAR(11) NOT NULL UNIQUE,  
cidade VARCHAR(40) NOT NULL,  
bairro VARCHAR(40) NOT NULL,  
rua VARCHAR(40) NOT NULL,  
numero INTEGER UNSIGNED NOT NULL,  
complemento VARCHAR(40) NOT NULL,  
id_UF INTEGER UNSIGNED NOT NULL,  
PRIMARY KEY (id),  
CONSTRAINT cliente_UF FOREIGN KEY id_UF (id_UF)  
REFERENCES uf (id)  
ON DELETE RESTRICT  
ON UPDATE RESTRICT  
)
```

```
CREATE TABLE arvore (  
id INTEGER UNSIGNED NOT NULL AUTO_INCREMENT,  
cord_latitude FLOAT NOT NULL,
```

```
cord_longitude FLOAT NOT NULL,  
id_especie INTEGER UNSIGNED NOT NULL,  
id_locais INTEGER UNSIGNED NOT NULL,  
id_cliente INTEGER UNSIGNED NOT NULL,  
PRIMARY KEY (id),  
CONSTRAINT arvore_especie FOREIGN KEY id_especie (id_especie)  
    REFERENCES especie (id)  
    ON DELETE RESTRICT  
    ON UPDATE RESTRICT,  
CONSTRAINT arvore_locais FOREIGN KEY id_locais (id_locais)  
    REFERENCES locais (id)  
    ON DELETE RESTRICT  
    ON UPDATE RESTRICT,  
CONSTRAINT arvore_cliente FOREIGN KEY id_cliente (id_cliente)  
    REFERENCES cliente (id)  
    ON DELETE RESTRICT  
    ON UPDATE RESTRICT  
)
```

```
CREATE TABLE protocolo (  
    id INTEGER UNSIGNED NOT NULL AUTO_INCREMENT,  
        data_criacao DATE NOT NULL,  
    data_plantar DATE NOT NULL,  
    deferido BIT NULL,  
    id_arvore INTEGER UNSIGNED NOT NULL,  
    id_funcionario INTEGER UNSIGNED NOT NULL,  
    PRIMARY KEY (id),  
    CONSTRAINT protocolo_arvore FOREIGN KEY id_arvore (id_arvore)  
        REFERENCES arvore (id)  
        ON DELETE RESTRICT  
        ON UPDATE RESTRICT,
```

```
        CONSTRAINT protocolo_funcionario FOREIGN KEY id_funcionario
(id_funcionario)
        REFERENCES funcinario (id)
        ON DELETE RESTRICT
        ON UPDATE RESTRICT
    )
```

5.3 Print das tabelas no MySQL

5.4 Código das inserções

5.5 Código das consultas solicitadas

Realizar consultas com que use pelos menos INNER JOIN entre 3 tabelas. Faça pelo menos um exemplo de consulta com uso de ORDER BY, outra consulta com LIKE (e suas variações), DISTINCT, >, <, =, .

Para cada consulta, escreva a mesma e de o print na tela do resultado.

6. TELAS DESENVOLVIDAS (N3)

Apresentar aqui as telas desenvolvidas (FIGMA, por exemplo)

Pode colar as telas

Pode colocar o link de onde estão:

7. IMPLEMENTAÇÃO (N3)

Aqui você deve apresentar o que realmente foi implementado de acordo com as funcionalidades definidas no item 7.

8. REFERÊNCIAS BIBLIOGRÁFICAS

Aqui devem constar todas as referências bibliográficas utilizadas para a construção deste documento: livros, revistas, sites da internet, apostilas, e publicações em geral.