



**UNIVERSIDADE ESTADUAL PAULISTA  
“JÚLIO DE MESQUITA FILHO”**

**Leonardo Yudi Higuti**

**RA: 171252608**

**Otávio Leite Ribeiro dos Santos**

**RA: 171251814**

**Onodrim**

**Ciência da Computação – Banco de Dados I**

**Professor Dr. Ronaldo Celso Messias Correia**

**24 de Junho de 2019**

## Índice

<b>1. Especificações do problema</b>	<b>3</b>
1.1 Objetivo	3
1.2 Entidades	3
1.3 Relacionamentos	5
1.4 Principais consultas	5
<b>2. Esquema conceitual</b>	<b>6</b>
<b>3. Esquema relacional</b>	<b>6</b>
<b>4. Normalização</b>	<b>7</b>
<b>5. Especificação de consultas</b>	<b>7</b>
5.1. Álgebra relacional	7
5.2. SQL	8
<b>6. Implementação da base de dados</b>	<b>8</b>
6.1. Criação da base de dados	8
6.2. Alimentação inicial	11

# **1. Especificações do problema**

## **1.1 Objetivo**

Neste documento, será descrita a modelagem do banco de dados utilizado pelo aplicativo Onodrim, aplicativo este, que permite que o usuário realize o cadastro de árvores, assim como seus detalhes e localizações, para que esses dados possam ser gerenciados por algum órgão especializado, utilizados em pesquisas, entre diversos outros usos.

## **1.2 Entidades**

A entidade “Usuario” representa os utilizadores do aplicativo. Os usuários deverão fornecer, obrigatoriamente, seus nomes completos, CPF, senha (que será criptografada no banco) e e-mail, este último será o identificador primário das contas dos usuários, não havendo a possibilidade de dois cadastros com o mesmo. Essa entidade também possui um atributo com o grau de permissão aos recursos do programa do usuário, possuindo valor “1” para usuários comuns, que podem cadastrar árvores e sugerir novas espécies para o banco, “2” para administradores menores, espécies de “supervisores” que possuem a capacidade de inserir uma espécie sem a necessidade de uma aprovação, além das funções básicas de um usuário comum, e finalmente “3” para desenvolvedores, que poderão adicionar, remover e acessar quaisquer dados que julgaram necessários (com exceção das senhas de outros usuários).

A entidade “Arvore” modela a instância de uma árvore qualquer, possuindo atributos que armazenam suas fotos, altura, largura, espécie, ano de plantio (da qual se infere à idade), data em que foram adicionadas ao banco e se são protegidas por algum decreto de tombamento (explicado em detalhes mais a frente). Além disso, possuem um identificador, que lhes é atribuído pelo programa e coordenadas de latitude e longitude, usadas para recuperar a localização onde são encontradas. Caso seja necessário, novos atributos poderão ser adicionados posteriormente.

A entidade “Tipo” descreve o modelo de uma espécie de árvore, possuindo obrigatoriamente um nome científico da espécie como identificador, um nome na qual se é geralmente chamada, tipo de fruto que possui, e um campo com informações básicas sobre suas utilidades para a fauna e para a humanidade, como por exemplo, fonte de alimento, folhas e flores medicinais, troncos utilizados para construções, entre outros.

A entidade “Local” armazena dados de localização de uma árvore, sendo identificada por uma latitude e longitude únicas. Além disso, possuem informações mais palpáveis de localização para os usuários: cidade, bairro, rua e CEP de onde se encontram.

A entidade fraca “Tombamento” possui dados sobre decretos de tombamento de árvores, possuindo dois atributos: “motivo”, que armazena uma breve descrição do porquê a árvore foi tombada, sua importância, etc; “decreto”, onde se encontra o número do decreto que descreve o tombamento.

A entidade agregada “Sugestão” engloba uma instância de um relacionamento “Sugere” entre um “Usuario” e um “Tipo”. Cada sugestão realizada possui um número identificador único, além das datas em que foram sugeridas e aprovadas (ou não) por um administrador, caso a sugestão ainda não tenha sido analisada, uma data futura estará inserida no banco. Possuem também um atributo de checagem que armazena “0” para aquelas sugestões que ainda não receberam um veredito de um administrador, “1” para sugestões que foram aprovadas, e “2” para sugestões negadas.

A entidade fraca “Foto” armazena um caminho (local ou online) para o acesso a uma fotografia de uma árvore, assim como o ID da árvore em questão.

### **1.3 Relacionamentos**

O relacionamento “Sugere” dá ao usuário comum a capacidade de fazer uma sugestão de uma espécie de árvore. Essa sugestão estará sujeita a aprovação ou negação de um administrador.

O relacionamento “Cadastra” permite que qualquer usuário cadastre árvores quaisquer no banco, podendo adicionar fotos e dados da mesma para enriquecer o banco.

O relacionamento “Pertence” impõe que uma árvore pertença a um tipo (espécie).

O relacionamento “Possui” faz a ligação entre uma árvore e um decreto de tombamento, que obrigatoriamente deverá estar associado a alguma árvore.

O relacionamento “Localiza” conecta uma árvore à sua localização física, representada por uma entidade “Local”

O relacionamento “Checa” define a capacidade de um administrador, seja ele de nível 2 ou 3, a aprovar ou negar uma sugestão de tipo de um usuário.

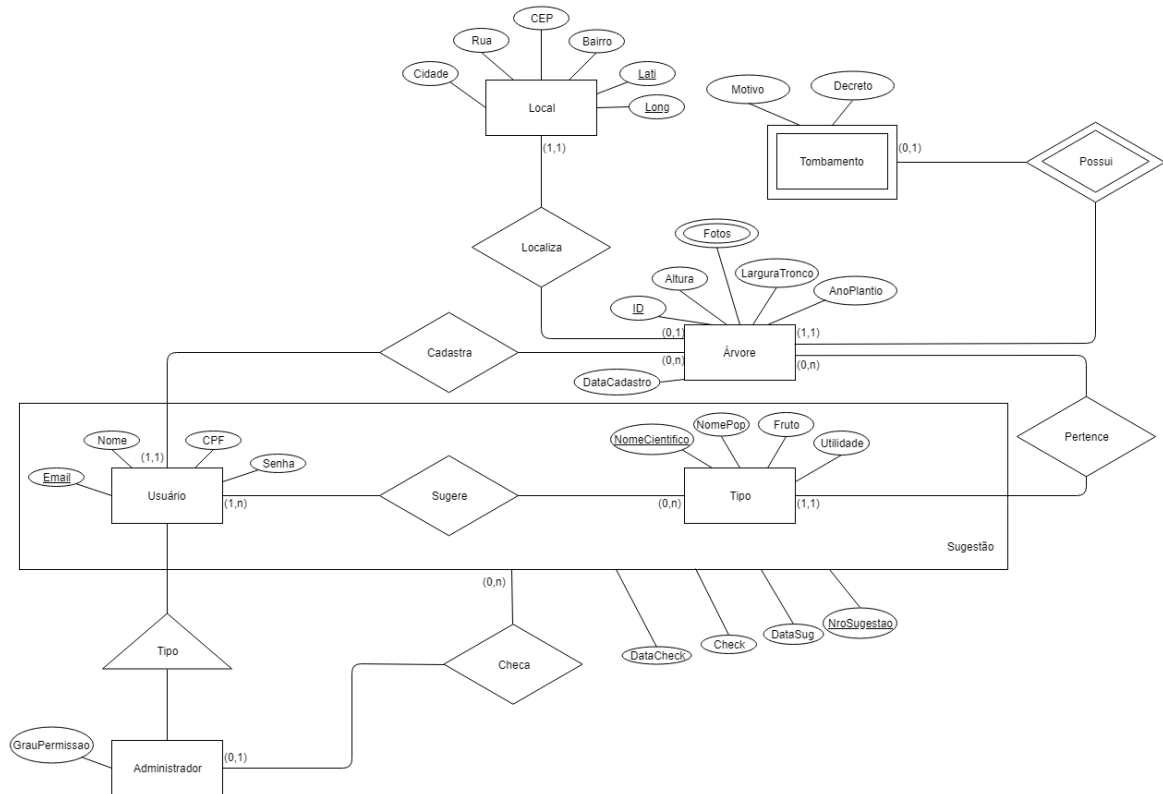
### **1.4 Principais consultas**

Alguns exemplos de consultas que podem ser realizadas neste modelo:

- Lista de árvores tombadas
- Lista de árvores cadastradas por um usuário específico
- Lista das espécies de árvores cadastradas
- Lista de sugestões de espécies negadas
- Lista de árvores de uma determinada espécie
- Lista de árvores cadastradas em um mesmo ano
- Lista de diferentes fotos de uma mesma árvore
- Lista de árvores localizadas em um bairro

## 2. Esquema conceitual

Abaixo está o Modelo Entidade Relacionamento, as chaves primárias se encontram sublinhadas.



## 3. Esquema relacional

Abaixo está o Modelo Relacional, as chaves primárias estão destacadas em negrito, e as chaves estrangeiras estão sublinhadas.

- Usuario (**Email**, cpf, nome, senha, grauPermissao);
- Arvore (**ID**, altura, larguraTronco, anoPlantio, usuarioCadastrador, dataCadastro, latitude, longitude, tipo);
- Local (**Latitude**, **Longitude**, cep, bairro, rua, cidade)
- Tombamento (**IDArvore**, decreto, motivo);
- Tipo (**NomeCientifico**, nomePopular, fruto, utilidade);
- Foto (IDArvore, caminho);
- Sugestao (**NroSugestao**, dataSugestao, check, dataCheck, Usuario, TipoArvore);

## 4. Normalização

O modelo relacional descrito anteriormente já se encontra normalizado na Terceira Forma Normal, as justificativas se encontram abaixo:

- 1FN: Não existem atributos multivalorados ou compostos.
- 2FN: Além de já se encontrar na 1FN, todos os atributos não-chave de todas as tabelas são completamente dependentes da chave primária, não havendo nenhuma dependência parcial.
- 3FN: Além de já se encontrar na 2FN, não existem atributos não-chave dependentes de outros atributos não-chave, eliminando a existência de dependências transitivas.

## 5. Especificação de consultas

A seguir, serão simuladas 5 consultas ao banco de dados. Estas consultas serão demonstradas em álgebra relacional, e em seguida com a própria linguagem SQL.

### 5.1. Álgebra Relacional

1. Busca dos IDs das árvores tombadas:  
 $\pi(\text{id\_arvore}) \text{ Arvore } |x| \text{ Tombamento};$
2. Busca de sugestões feitas após uma data X:  
 $\sigma(\text{data\_sug} > X) \text{ Sugestao};$
3. Busca do nome dos administradores que aprovaram ao menos uma sugestão:  
 $\pi(\text{nome}) \sigma(\text{id\_usuario} = \text{email}) \text{ Sugestao } X (\sigma(\text{grau\_permissao} > 1) \text{ Usuario});$
4. Lista dos caminhos de todas as fotos de uma árvore X:  
 $\pi(\text{caminho}) \sigma(\text{id\_arvore} = X) \text{ Arvore};$
5. Lista dos nomes populares dos tipos de árvore encontrados em uma rua X:  
 $\pi(\text{nome\_pop}) \sigma(\text{rua} = X) \text{ Localizacao } |x| \text{ Arvores};$

## 5.2. SQL

1. Busca dos IDs das árvores tombadas:

- `SELECT id_arvore FROM Arvore, Tombamento WHERE Arvore.id_arvore = Tombamento.id_arvore;`

2. Busca de sugestões feitas após uma data X:

- `SELECT nro_sug, data_sug, checado, data_checagem FROM Sugestao WHERE data_sug > X;`

3. Busca do nome dos administradores que aprovaram ao menos uma sugestão:

- `SELECT nome FROM Usuario, Sugestao WHERE email = id_usuario;`

4. Lista dos caminhos de todas as fotos de uma árvore X:

- `SELECT caminho FROM Foto, Arvore WHERE Foto.id_arvore = Arvore.id_arvore`

5. Lista dos nomes populares dos tipos de árvore encontrados em uma rua X:

- `SELECT nome_pop FROM Tipo, Arvore, Localizacao WHERE (rua = X) AND (Arvore.latitude = Localizacao.latitude AND Arvore.longitude = Localizacao.longitude);`

## 6. Implementação da base de dados

Finalmente, será apresentada a implementação da base de dados com a linguagem SQL, bem como uma breve alimentação inicial do banco com alguns INSERTs.

### 6.1. Criação da base de dados

```
CREATE DATABASE Onodrim;
```

```
CREATE TABLE Usuario (  
    email varchar(30) NOT NULL,
```



```
    cpf varchar(11) NOT NULL,  
    nome varchar(50) NOT NULL,  
    senha varchar(50) NOT NULL,  
    grau_permissao int(2) NOT NULL,  
    CONSTRAINT pk_email PRIMARY KEY (email)  
);
```

```
CREATE TABLE Tipo (  
    nome_cientifico varchar(30) NOT NULL,  
    nome_pop varchar(20) NOT NULL,  
    tipo_fruto varchar(30),  
    utilidade varchar(512),  
    CONSTRAINT pk_nome PRIMARY KEY (nome_cientifico),  
);
```

```
CREATE TABLE Localizacao (  
    latitude float NOT NULL,  
    longitude float NOT NULL,  
    cep varchar(8),  
    rua varchar(50),  
    bairro varchar(50),  
    cidade varchar(50),  
    CONSTRAINT pk_latlon PRIMARY KEY (latitude, longitude)  
);
```

```
CREATE TABLE Arvore (  
    id_arvore varchar(20) NOT NULL,  
    altura float,  
    largura_tronco float,  
    ano_plantio int(4),  
    data_cadastro varchar(8) NOT NULL,  
    usuario varchar(11) NOT NULL,
```

```

    tipo varchar(30) NOT NULL,
    latitude float NOT NULL,
    longitude float NOT NULL,
    CONSTRAINT fk_usuarioA FOREIGN KEY (usuario) REFERENCES
        Usuario(email),
    CONSTRAINT fk_tipo FOREIGN KEY (tipo) REFERENCES Tipo(nome_cientifico),
    CONSTRAINT fk_local FOREIGN KEY (latitude,longitude) REFERENCES
        Localizacao(latitude,longitude),
    CONSTRAINT pk_id PRIMARY KEY (id_arvore)
);

```

```

CREATE TABLE Foto (
    caminho varchar(100) NOT NULL,
    id_arvore varchar(20) NOT NULL,
    CONSTRAINT fk_idarvore FOREIGN KEY (id_arvore) REFERENCES Arvore
        (id_arvore)
);

```

```

CREATE TABLE Tombamento (
    id_arvore varchar(20) NOT NULL,
    motivo varchar(512),
    decreto varchar(20),
    CONSTRAINT fk_idarvoreT FOREIGN KEY (id_arvore) REFERENCES Arvore
        (id_arvore)
);

```

```

CREATE TABLE Sugestao (
    nro_sug varchar(20) NOT NULL,
    data_sug DATE NOT NULL,
    checado bool NOT NULL,
    data_checagem DATE NOT NULL,
    id_usuario varchar(11) NOT NULL,

```

```

nome_cientifico varchar(30) NOT NULL,
CONSTRAINT fk_usuarioS FOREIGN KEY (id_usuario) REFERENCES Usuario
(email),
CONSTRAINT fk_arvore FOREIGN KEY (nome_cientifico) REFERENCES Tipo
(nome_cientifico),
CONSTRAINT pk_nro PRIMARY KEY (nro_sug)
);

```

## 6.2. Alimentação inicial

- INSERT INTO Usuario VALUES ('leonardo.higuti@unesp.br', '12345678910', 'Leonardo Yudi Higuti', MD5('leo123'), 3);
- INSERT INTO Usuario VALUES ('otavio.leite@unesp.br', '23456789101', 'Otavio Leite dos Santos', MD5('otavio123'), 3);
- INSERT INTO Usuario VALUES ('ronaldo.correia@unesp.br', '34567891011', 'Ronaldo Celso Messias Correia', MD5('ronaldo123'), 2);
- INSERT INTO Usuario VALUES ('rogerio.garcia@unesp.br', '45678910111', 'Rogerio Eduardo Garcia', MD5('rogerio123'), 1);
- INSERT INTO Usuario VALUES ('danilo.eler@unesp.br', '56789101112', 'Danilo Medeiros Eler', MD5('danilo123'), 1);
- INSERT INTO Tipo VALUES ('Caesalpinia leiostachya', 'Pau-ferro', 'legume', null);
- INSERT INTO Tipo VALUES ('Caesalpinia peltophoroides', 'Sibipiruna', 'legume', 'Reflorestamento');

- INSERT INTO Tipo VALUES ('Campomanesia phaea', 'Cambuci', 'cápsula', 'Bom para o plantio em calçadas');
- INSERT INTO Tipo VALUES ('Cordia superba', 'Guanhuma', 'baga', 'Consumo e reflorestamento');
- INSERT INTO Tipo VALUES ('Erythrina speciosa', 'Mulungu', null, null);
- INSERT INTO Localizacao VALUES (-22.894768, -43.223303, '20921420', 'Rua General Bruce', null, 'Rio de Janeiro');
- INSERT INTO Localizacao VALUES (-22.128757, -51.400190, null, 'Av. Quatorze de Setembro', null, null);
- INSERT INTO Localizacao VALUES (-22.120681, -51.404003, null, null, 'Conjunto Habitacional Ana', null);
- INSERT INTO Localizacao VALUES (-22.135709, -51.411780, null, null, null, 'Presidente Prudente');
- INSERT INTO Localizacao VALUES (-22.118214, -51.433864, '19067350', 'Rua Altair de Senna', 'Parque Cedral', 'Presidente Prudente');
- INSERT INTO Arvore VALUES ('1', 15.76, 4.23, 2005, '2019-06-15', 'leonardo.higuti@unesp.br', 'Caesalpinia leiostachya', -22.894768, -43.223303);
- INSERT INTO Arvore VALUES('2', 12, 0.35, 2008, '2019-06-15', 'leonardo.higuti@unesp.br', 'Caesalpinia peltophoroides', -22.128757, -51.400190);

- INSERT INTO Arvore VALUES ('3', 5, 0.2, 2013, '2019-06-15', 'leonardo.higuti@unesp.br', 'Campomanesia phaea', -22.120681, -51.404003);
- INSERT INTO Arvore VALUES ('4', 10, 0.3, 2010, '2019-06-15', 'otavio.leite@unesp.br', 'Cordia superba', -22.135709, -51.411780);
- INSERT INTO Arvore VALUES ('5', null, null, null, '2019-06-15', 'otavio.leite@unesp.br', 'Erythrina speciosa', -22.118214, -51.433864);
- INSERT INTO Foto VALUES ('Untitled.png', '4');
- INSERT INTO Foto VALUES ('arvore.png', '4');
- INSERT INTO Foto VALUES ('ed9fe362-63a1-4fd7-acd7-e1fb63a9c98d.png', '1');
- INSERT INTO Foto VALUES ('Passeio(34).png', '4');
- INSERT INTO Foto VALUES ('FotoBonita.png', '3');
- INSERT INTO Tombamento VALUES('1', null, 'Decreto 30.631');
- INSERT INTO Sugestao VALUES ('ronaldo.correia@unesp.br', 'Cordia superba', '1', '2100-01-30', '2019-06-15', 0);
- 
- INSERT INTO Sugestao VALUES ('ronaldo.correia@unesp.br', 'Caesalpinia peltophoroides', '2', '2100-01-30', '2 19-06-15', 0);
- INSERT INTO Sugestao VALUES ('ronaldo.correia@unesp.br', 'Erythrina speciosa', '3', '2100-01-30', '2019-06-15', 0);

- INSERT INTO Sugestao VALUES ('danilo.eler@unesp.br', 'Campomanesia phaea', '4', '2100-01-30', '2019-06-15', 0);
- INSERT INTO Sugestao VALUES ('danilo.eler@unesp.br', 'Erythrina speciosa', '5', '2100-01-30', '2019-06-15', 0);