

UrbLixo

Versão 2.0

SUMÁRIO

1. Introdução.....	1
1.1. Finalidade.....	1
1.2. Objetivo.....	1
1.3. Definições acrônimos a abreviações.....	1
1.4. Visão Geral.....	1
2. Requisitos do Sistema.....	1
3. Entidades.....	2
4. Relações.....	2
5. Diagramas.....	3

1. Introdução

1.1. Finalidade

Este documento especifica oficialmente o escopo do Sistema UrbLixo, desenvolvido pelos alunos Elysson Alves de Lacerda e Gustavo Almeida Monteiro, da disciplina de Fundamento de Banco de Dados. Este documento será versionado, servindo de base para a construção da aplicação.

1.2. Objetivo

A aplicação desenvolvida a partir deste documento é o UrbLixo. Seu objetivo é informar o processo de descarte de lixo dentro dos centros urbanos, com foco na consulta de dados, proporcionando informações sobre os resíduos produzidos pelos centros urbanos, permitindo acessar dados do descarte, pontos de coleta e outras consultas. Portanto, fornecer informações relevantes para uma aprendizagem eficiente é seu objetivo principal.

1.3. Definições acrônimos a abreviações

Esta seção fornece as definição do termo, acrônimo e abreviação requerida para interpretar este documento de requisitos.

As abreviação usada no documento é:

RF00X – Requisito Funcional;

1.4 Visão Geral

Além desta seção introdutória, as seções seguintes descrevem as funcionalidades e entidades envolvidas na aplicação.

Requisitos funcionais: apresenta os requisitos funcionais do sistema juntamente com a descrição de cada requisito;

2. Requisitos Funcionais do Sistema

RF001 – Consultar Dados do Lixo: O sistema deve permitir o usuário consultar dados de descarte do lixo da cidade;

RF002 – Consultar Localização do Ponto de Coleta: O sistema deve permitir o usuário consultar a localização de pontos de coleta;

RF003 – Consultar Rotas de Coleta: O sistema deve permitir o usuário consultar as rotas dos caminhões de coleta;

RF004 – Consultar a Entrada e Saída de resíduos: O sistema deve permitir ao usuário o monitoramento de entrada e saída de resíduos;

RF005 – Consultar Dados dos Caminhões de Coleta: O sistema deve permitir o usuário consultar dados sobre os caminhões de coleta;

RF006 – Consultar Dados sobre Locais de Descarte: O sistema deve permitir ao usuário consultar dados sobre os locais de descarte;

3. Entidades

Usuário:

Atributos → id_usuario (PK) SERIAL, nome VARCHAR(70) UNIQUE NOT NULL, senha VARCHAR(6) UNIQUE NOT NULL, e-mail VARCHAR(70) NOT NULL;

Lixo:

Atributos → id_lixo (PK) SERIAL, classificação VARCHAR(20), id_ponto (FK) INT;

Pontos de Coleta:

Atributos → id_ponto (PK) SERIAL, endereço ATRIBUTO COMPOSTO → (bairro VARCHAR(50), rua VARCHAR(50), numero INT) NOT NULL;

Rotas de Coleta:

Atributos → id_rota (PK) SERIAL, ponto_partida ATRIBUTO COMPOSTO → (bairro VARCHAR(100), rua VARCHAR(100)), id_ponto (FK) INT;

Caminhão:

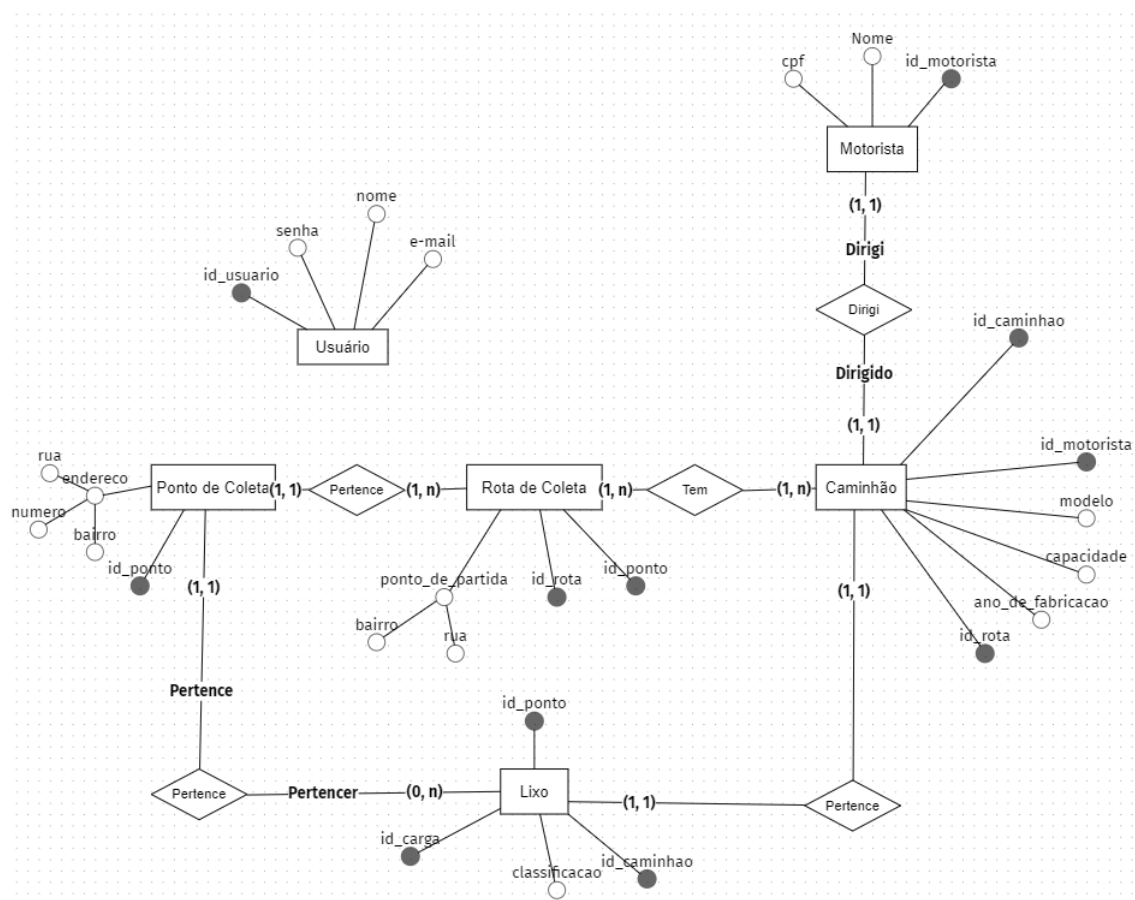
Atributos → id_caminhao (PK) SERIAL, id_rota (FK) INT, modelo VARCHAR(20), capacidade INT, ano_de_operacao DATE;

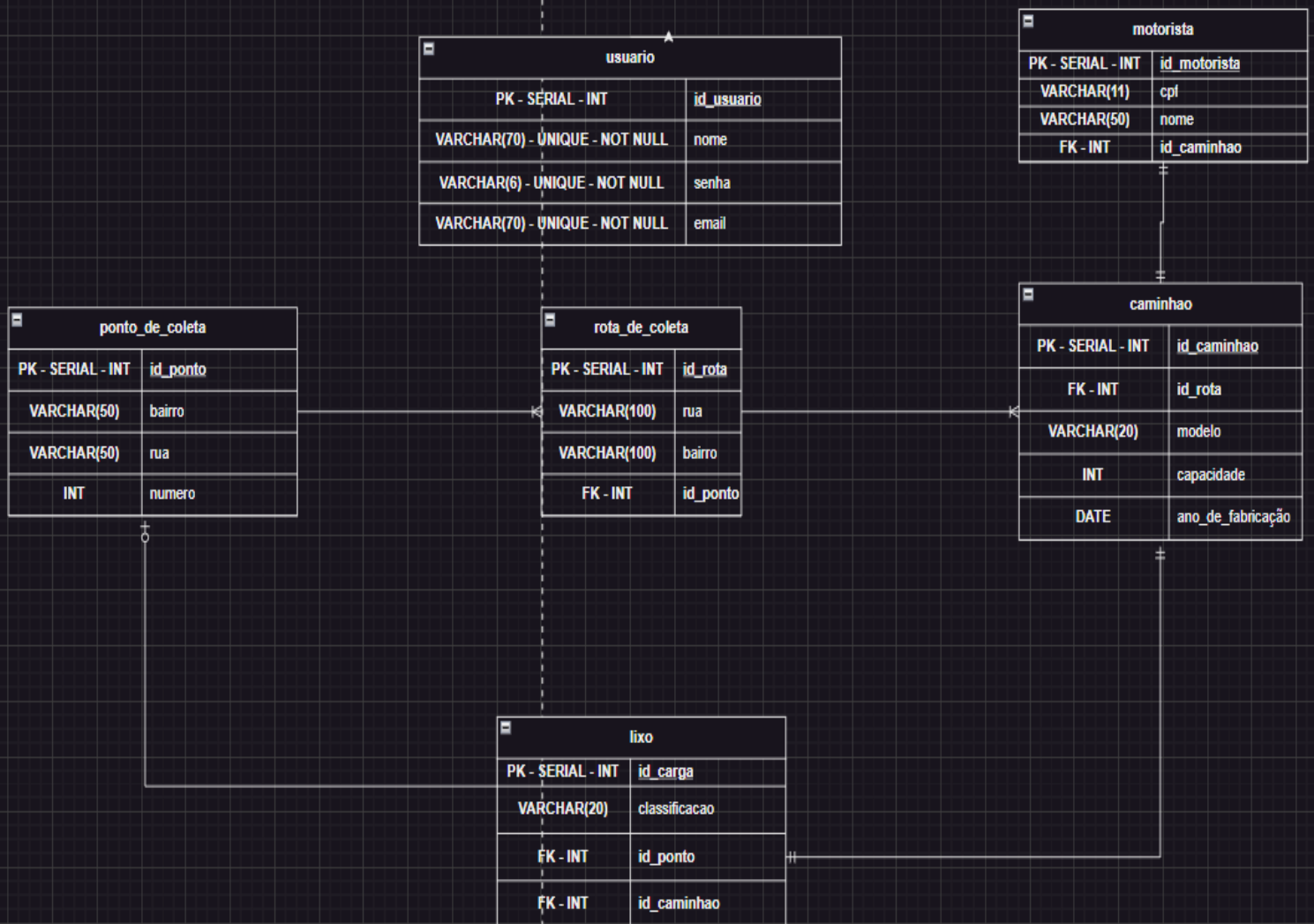
4. Relação entre as entidades

Nesse tópico será apresentado um resumo sobre as relações entre as entidades envolvida :

- A relação de cardinalidade entre lixo e o ponto de coleta seria (0,n), onde no mínimo não se tem lixo e no máximo várias cargas lixo pertencem a um ponto de coleta;
- A relação de cardinalidade entre lixo e caminhão seria (1,1), onde no mínimo e no máximo 1 caminhão tem carga de lixo ou uma carga pertence a um caminhão;
- A relação de cardinalidade entre o ponto de coleta e a rota de coleta seria (1,n), onde no mínimo 1 e no máximo vários pontos de coleta podem pertencer a uma rota de coleta;
- A relação de cardinalidade entre a rota de coleta e o caminhão seria (1,n), onde no mínimo 1 uma rota tem um caminhão ou vários caminhões que transitam por uma rota;
- A relação de cardinalidade entre caminhão e motorista seria (1,1), onde no mínimo 1 e no máximo 1, um motorista dirige um caminhão e um caminhão é dirigido por um motorista;

5. Diagramas





5. *SQL e Python*

AS consultas SQL, Visões, Triggers e Python estarão no final do documento.

Data	Entrega	Tarefa	Responsável
06/05/2023	06/05/2023	Requisitos Funcionais	Ambos
25/05/2023	25/05/2023	Diagramas da Aplicação	Ambos
10/07/2023	10/07/2023	Interface da Aplicação	Ambos

Link para acesso ao vídeo de explicação dos diagramas

[Vídeo explicação](#)

Aperte CTRL + enter para abrir

```

-- Criação Urblixo

SET SCHEMA 'urblixo';

-- Criação da tabela sobre os Pontos de coleta, local onde o lixo será
coletado
CREATE TABLE ponto_de_coleta (
    id_ponto SERIAL PRIMARY KEY,
    bairro VARCHAR(50),
    rua VARCHAR(50),
    numero INT
);

-- Criação da tabela sobre o Lixo, entidade que julgamos ser a mais
importante
CREATE TABLE lixo (
    id_carga SERIAL PRIMARY KEY,
    classificacao VARCHAR(20),
    id_ponto INT,
    id_caminhao INT
);

-- Criação da tabela sobre o Usuário, onde temos os atributos do usuário
que irá acessar o sistema
CREATE TABLE usuario (
    id_usuario SERIAL PRIMARY KEY,
    nome VARCHAR(70) UNIQUE NOT NULL,
    senha VARCHAR(6) UNIQUE NOT NULL,
    email VARCHAR(70) NOT NULL
);

-- Criação da tabela sobre o Motorista do caminhão
CREATE TABLE motorista (
    id_motorista SERIAL PRIMARY KEY,
    cpf VARCHAR(11),
    nome VARCHAR(50),
    id_caminhao INT
);

-- Criação da tabela sobre as Rotas de coleta, onde os caminhões passarão
para coletar o lixo
CREATE TABLE rota_de_coleta (
    id_rota SERIAL PRIMARY KEY,
    rua VARCHAR(100),
    bairro VARCHAR(100),
    id_ponto INT
);

-- Criação da tabela Caminhão, onde o lixo será armazenado
CREATE TABLE caminhao (
    id_caminhao SERIAL PRIMARY KEY,
    id_rota INT,
    modelo VARCHAR(20),
    capacidade INT,

```

```

        ano_de_fabricacao DATE
    );

-- Conexão entra o caminhao a tabela lixo e o ponto de coleta que ele
veio
ALTER TABLE lixo ADD CONSTRAINT fk_lixo_ponto_de_coleta FOREIGN KEY
(id_ponto) REFERENCES ponto_de_coleta(id_ponto);

-- Conexão entre o lixo e o caminhao de onde ele veio
ALTER TABLE lixo ADD CONSTRAINT fk_lixo_caminhao FOREIGN KEY
(id_caminhao) REFERENCES caminhao(id_caminhao);

-- Conexão entre o motorista e o o carro que ele dirige
ALTER TABLE motorista ADD CONSTRAINT fk_motorista_caminhao FOREIGN KEY
(id_caminhao) REFERENCES caminhao(id_caminhao);

-- Conexão entre a rota de coleta e o ponto onde o lixo será descartado
ALTER TABLE rota_de_coleta ADD CONSTRAINT
fk_rota_de_coleta_ponto_de_coleta FOREIGN KEY (id_ponto) REFERENCES
ponto_de_coleta(id_ponto);

-- Conexão entre o caminhão e a rota que ele faz
ALTER TABLE caminhao ADD CONSTRAINT fk_caminhao_rota_de_coleta FOREIGN
KEY (id_rota) REFERENCES rota_de_coleta(id_rota);

```



```
-- Povoamento do Urblixo
```

```
SET SCHEMA 'urblixo';
```

```
-- Inserção de 10 dados em cada tabela
```

```
INSERT INTO ponto_de_coleta (bairro, rua, numero) VALUES
    ('São João', 'Rua Domicio Cassiano Almeida', 204 ),
    ('Centro', 'Rua Laerte Pinheiro', 105),
    ('Baviera', 'Rua Nabor Crebilon de Sousa', 135),
    ('Planalto Universitário', 'Rua José Enéas Monteiro Lessa', 245),
    ('Combate', 'Rua Nossa Senhora da Conceição', 271),
    ('Planato Renascer', 'Rua João Vicente Queiroz', 131),
    ('Carrascal', 'Rua José de Queiroz Pessoa', 247),
    ('Nova Jesusalém', 'Rua José Candido de Sousa', 303),
    ('Putiú', 'Rua Santa Isabel', 285),
    ('Jardim dos Monólitos', 'Rua Wagner Vasconcelos Braga', 140);
```

```
INSERT INTO usuario (nome, senha, email) VALUES
    ('Edson', '044179', 'edsonyuri@gmail.com'),
    ('Betina', '513815', 'betina_pereira@gmail.com'),
    ('Caio', '106606', 'caio_araujo@gmail.com'),
    ('Antonio', '659950', 'antonio_nicolas@gmail.com'),
    ('Pietra', '895819', 'pietra_tatiane@gmail.com'),
    ('Ruan', '449251', 'ruan_farias@gmail.com'),
    ('Otávio', '690140', 'otavio_gomes@gmail.com'),
    ('Lais', '078807', 'lais_mirela@gmail.com'),
    ('Bruna', '943969', 'brunanicole@gmail.com'),
    ('Eduarda', '526663', 'duda_castro@gmail.com');
```

```
INSERT INTO lixo (classificacao, id_ponto, id_caminhao) VALUES
    ('Plástico', 1, 1),
    ('Vidro', 2, 2),
    ('Papel', 3, 3),
    ('Lixo Orgânico', 4, 4),
    ('Madeira', 5, 5),
    ('Vidro', 6, 6),
    ('Metal', 7, 7),
    ('Plástico', 8, 8),
    ('Papel', 9, 9),
    ('Lixo Orgânico', 10, 10);
```

```
INSERT INTO motorista (cpf, nome, id_caminhao) VALUES
    ('34229986327', 'Daniel', 1),
    ('17247938374', 'Breno', 2),
    ('77664308356', 'Carlos', 3),
    ('48897924301', 'Pedro', 4),
    ('74321310318', 'João', 5),
    ('57560648347', 'Vitor', 6),
    ('84120283330', 'Carlos', 7),
    ('93233968345', 'Lucas', 8),
    ('00957756330', 'Gustavo', 9),
    ('63061022309', 'Paulo', 10);
```

```
INSERT INTO rota_de_coleta (bairro, rua, id_ponto) VALUES
```

```
('Planalto Renascer','Rua Nova Vida', 1),
('Centro', 'Rua Juvencio Alves de Oliveira', 2),
('Carrascal', 'Travessa Francisco Severino Figueiredo', 3),
('Jardim dos Monólitos', 'Rua Doutor Aldízio Alves Vieira', 4),
('Curicaca', 'Rua Municipalista Doutor Américo Barreira', 5),
('Centro', 'Travessa Clóvis Beviláqua', 6),
('Triângulo', 'Travessa Manoel Pereira de Santana', 7),
('Campo Novo', 'Travessa José Alencar de Macêdo', 8),
('Irajá', 'Rua Pedro Amaro', 9),
('Putiú', 'Rua Edilson Marques dos Santos', 10);
```

```
INSERT INTO caminhao (id_rota, modelo, capacidade, ano_de_fabricacao)
VALUES
```

```
(1, 'Mercedes-Benz', 10, '2020-01-01'),
(2, 'Volvo', 20, '2019-01-01'),
(3, 'Scania', 15, '2022-01-01'),
(4, 'Renault Trucks', 12, '2018-01-01'),
(5, 'Hino', 18, '2020-01-01'),
(6, 'Iveco', 16, '2021-01-01'),
(7, 'Isuzu', 10, '2019-01-01'),
(8, 'Peterbilt', 25, '2022-01-01'),
(9, 'International', 20, '2017-01-01'),
(10, 'Mack', 15, '2020-01-01');
```

```

-- Perguntas SQL, 10 perguntas no total sobre o BD
SET SCHEMA 'urblixo';

--01 - Quais os caminhões que possuem capacidade superior a 15 ?
SELECT * FROM caminhao WHERE capacidade > 15;

--02 - Qual o número total de cargas de lixo em cada rota de coleta ?
SELECT r.id_rota, COUNT(l.id_carga) AS total_cargas
FROM rota_de_coleta r
LEFT JOIN lixo l ON r.id_ponto = l.id_ponto
GROUP BY r.id_rota;

--03 - Qual a listar os usuários ordenados pelo nome em ordem alfabética ?
SELECT * FROM usuario ORDER BY nome;

--04 - Qual a contagem de motoristas associados a cada caminhão ?
SELECT c.id_caminhao, COUNT(m.id_motorista) AS total_motoristas
FROM caminhao c
LEFT JOIN motorista m ON c.id_caminhao = m.id_caminhao
GROUP BY c.id_caminhao;

--05 - Quais as cargas de lixo classificadas como "Plástico" e exibir o
ponto de coleta que receberam a carga ?
SELECT l.id_carga, l.classificacao, pc.bairro, pc.rua, pc.numero
FROM lixo l
INNER JOIN ponto_de_coleta pc ON l.id_ponto = pc.id_ponto
WHERE l.classificacao = 'Plástico';

--06 - Quais os motoristas que dirigem caminhões fabricados após 2020 ?
SELECT m.id_motorista, m.cpf, m.nome
FROM motorista m
INNER JOIN caminhao c ON m.id_caminhao = c.id_caminhao
WHERE c.ano_de_fabricacao > '2020-01-01';

--07 - Qual a média de capacidade dos caminhões ?
SELECT AVG(capacidade) AS media_capacidade
FROM caminhao;

/*08 - Qual a contagem de cargas de lixo em cada ponto de coleta e
exibir apenas os pontos que possuem mais de/igual a 1 carga ?
(obs : povoando mais o BD haverá mais resultados)
*/
SELECT p.id_ponto, p.bairro, p.rua, p.numero, COUNT(l.id_carga) AS
total_cargas
FROM ponto_de_coleta p
LEFT JOIN lixo l ON p.id_ponto = l.id_ponto
GROUP BY p.id_ponto, p.bairro, p.rua, p.numero
HAVING COUNT(l.id_carga) >= 1;

--09 - Quais os pontos de coleta localizados no bairro "Centro" ?
SELECT * FROM ponto_de_coleta WHERE bairro = 'Centro';

--10 - Qual a contagem total de cargas de lixo em cada ponto de coleta ?

```

```
SELECT pc.id_ponto, pc.rua, COUNT(l.id_carga) AS total_cargas
FROM ponto_de_coleta pc
LEFT JOIN lixo l ON pc.id_ponto = l.id_ponto
GROUP BY pc.id_ponto, pc.rua;
```