

PONTIFÍCIA UNIVERSIDADE CATÓLICA DE MINAS GERAIS

Instituto de Informática — Unidade Praça da Liberdade

Curso: Engenharia de Software. **Prof.^a:** Laura Castro

Disciplina: Bancos de Dados

Alunos: Bruno Armanelli, Cristiano Souza, Henrique Freire, Lucas Lima.

Sistema: Defesa Civil Contagem

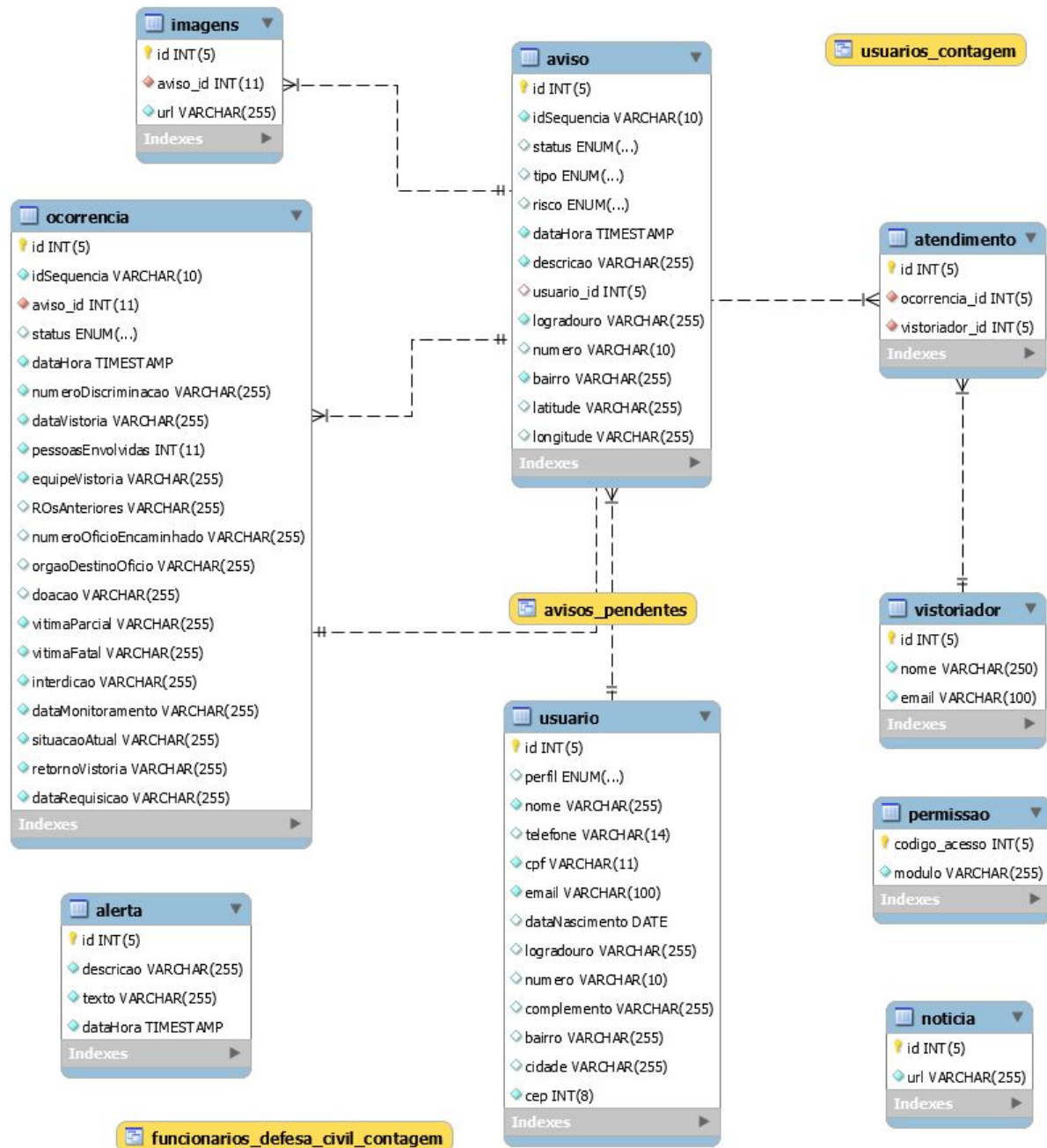
1. Descrição do sistema

O sistema escolhido para ser modelado tem como objetivo servir de interface de comunicação entre a população e a Defesa Civil do município de Contagem-MG. O sistema conta com uma aplicação móvel que a população utiliza para visualizar o mapa das áreas de risco, comunicar ao órgão situações de risco, receber alertas e acompanhar notícias sobre as ações do Órgão. A equipe da Defesa Civil utiliza uma aplicação web para gerenciar os avisos recebidos, registrar ocorrências, atribuir funcionários para realizarem as vistorias e controlar as demandas em andamento a nível gerencial.

Para acessar a aplicação, tanto os cidadãos quanto a equipe devem se registrar na aplicação por meio de uma conta do Google, o que elimina a necessidade de se gerenciar as senhas dos usuários. Os usuários cadastrados tem como perfil padrão o nível de acesso “cidadão”; Um funcionário pré-cadastrado na aplicação tem permissão de alterar os privilégios de acesso dos outros usuários, garantindo que somente pessoal autorizado tenha acesso às informações gerenciais do sistema. Os perfis de usuário possuem permissões de acesso diferentes a cada módulo e são controlados via aplicação, em que usuários com perfil “Gestor” definem quais módulos os outros perfis podem acessar.

Os Atendimentos são registrados automaticamente a cada vez que um Aviso é processado e possuem os identificadores de Aviso e Ocorrência. As Ocorrências criadas sempre são vinculadas a um aviso pré-existente para obter seus dados e fotos informados pelo cidadão. O histórico de Alertas emitidos é mantido com o registro do funcionário que o emitiu, data, hora, e texto. As notícias são elaboradas pela Secretaria de Comunicação, que envia para a equipe da Defesa Civil a URL da postagem.

2. Modelo DER



3. Script SQL de criação do Banco de Dados e Tabelas:

```
DROP DATABASE IF EXISTS DEFESACIVILCONTAGEM;
```

```
CREATE DATABASE IF NOT EXISTS DEFESACIVILCONTAGEM;
```

```
USE DEFESACIVILCONTAGEM;

DROP TABLE IF EXISTS USUARIO;

CREATE TABLE IF NOT EXISTS USUARIO (

    id int(5) NOT NULL AUTO_INCREMENT,

    perfil ENUM('Cidadao', 'Atendente', 'Vistoriador', 'Gestor')
    DEFAULT 'Cidadao',

    nome varchar(255) NOT NULL,

    telefone varchar(14),

    cpf varchar(11) NOT NULL,

    email varchar(100) NOT NULL,

    dataNascimento date,

    logradouro varchar(255),

    numero varchar(10),

    complemento varchar(255),

    bairro varchar(255),

    cidade varchar(255) DEFAULT 'Contagem',

    cep int(8) NOT NULL,

    PRIMARY KEY (id),

    CONSTRAINT UQ_cpf_email UNIQUE(cpf, email)

);

DROP TABLE IF EXISTS AVISO;

CREATE TABLE IF NOT EXISTS AVISO (

    id int(5) NOT NULL AUTO_INCREMENT,

    idSequencia varchar(10) NOT NULL,
```

```
status ENUM('Pendente', 'Processado', 'Cancelado') DEFAULT
'Pendente',

tipo ENUM(

    'Risco Construtivo',

    'Interdição',

    'Escorregamento',

    'Remoção',

    'Abatimento',

    'Solo',

    'Solopamento',

    'Drenagem',

    'Erosão',

    'Inundação',

    'Alagamento',

    'Desabamento',

    'Terreno',

    'impróprio',

    'Outros/Sem dados'

) DEFAULT 'Outros/Sem dados',

risco ENUM(

    'Inexistente ou Baixo',

    'Médio',

    'Alto',

    'Muito Alto'
```

```

) DEFAULT 'Inexistente ou Baixo',

dataHora TIMESTAMP DEFAULT CURRENT_TIMESTAMP,

descricao varchar(255) NOT NULL,

usuario_id int(5),

logradouro varchar(255) NOT NULL,

numero varchar(10),

bairro varchar(255) NOT NULL,

latitude varchar(255),

longitude varchar(255),

PRIMARY KEY (id),

CONSTRAINT UQ_id_seq UNIQUE(idSequencia),

CONSTRAINT fk_usuario_id FOREIGN KEY (usuario_id) REFERENCES
USUARIO (id) ON DELETE SET NULL

);

DROP TABLE IF EXISTS IMAGENS;

CREATE TABLE IF NOT EXISTS IMAGENS (

    id int(5) NOT NULL AUTO_INCREMENT,

    aviso_id INT NOT NULL,

    url VARCHAR(255) NOT NULL,

    PRIMARY KEY(id),

    CONSTRAINT UQ_url UNIQUE(url),

    CONSTRAINT fk_imagens_aviso_id FOREIGN KEY (aviso_id) REFERENCES
    AVISO(id) ON DELETE CASCADE

);

```

```
DROP TABLE IF EXISTS OCORRENCIA;

CREATE TABLE IF NOT EXISTS OCORRENCIA (

    id int(5) NOT NULL AUTO_INCREMENT,

    idSequencia VARCHAR(10) NOT NULL,

    aviso_id INT NOT NULL,

    status ENUM(      'Pendente',

        'Em aberto',

        'Concluído',

        'Em Espera',

        'Cancelado'

    ) DEFAULT 'Pendente',

    dataHora TIMESTAMP DEFAULT CURRENT_TIMESTAMP ON UPDATE CURRENT_TIMESTAMP,

    numeroDiscriminacao VARCHAR(255) NOT NULL,

    dataVistoria VARCHAR(255) NOT NULL,

    pessoasEnvolvidas INT NOT NULL,

    equipeVistoria VARCHAR(255) NOT NULL,

    ROsAnteriores VARCHAR(255),

    numeroOficioEncaminhado VARCHAR(255),

    orgaoDestinoOficio VARCHAR(255),

    doacao VARCHAR(255),

    vitimaParcial VARCHAR(255) NOT NULL,

    vitimaFatal VARCHAR(255) NOT NULL,

    interdicao VARCHAR(255) NOT NULL,
```

```

dataMonitoramento VARCHAR(255) NOT NULL,

situacaoAtual VARCHAR(255) NOT NULL,

retornoVistoria VARCHAR(255) NOT NULL,

dataRequisicao VARCHAR(255) NOT NULL,

PRIMARY KEY (id),

CONSTRAINT UQ_id_seq UNIQUE (idSequencia),

CONSTRAINT fk_ocorrencia_aviso_id FOREIGN KEY (aviso_id)
REFERENCES AVISO(id)

);

DROP TABLE IF EXISTS VISTORIADOR;

CREATE TABLE IF NOT EXISTS VISTORIADOR (

    id int(5) NOT NULL AUTO_INCREMENT,

    nome VARCHAR(250) NOT NULL,

    email VARCHAR(100) NOT NULL,

    PRIMARY KEY (id),

    CONSTRAINT UQ_vistoriador UNIQUE (nome, email)

);

DROP TABLE IF EXISTS ATENDIMENTO;

CREATE TABLE IF NOT EXISTS ATENDIMENTO (

    id int(5) NOT NULL AUTO_INCREMENT,

    ocorrencia_id int(5) NOT NULL,

    vistoriador_id int(5) NOT NULL,

    PRIMARY KEY (id),

    FOREIGN KEY (ocorrencia_id) REFERENCES OCORRENCIA(id),

```

```
FOREIGN KEY (vistoriador_id) REFERENCES VISTORIADOR(id)

);

DROP TABLE IF EXISTS PERMISSAO;

CREATE TABLE IF NOT EXISTS PERMISSAO (

    codigo_acesso int(5) NOT NULL AUTO_INCREMENT,

    modulo VARCHAR(255) NOT NULL,

    PRIMARY KEY (codigo_acesso)

);

DROP TABLE IF EXISTS NOTICIA;

CREATE TABLE IF NOT EXISTS NOTICIA (

    id int(5) NOT NULL AUTO_INCREMENT,

    url VARCHAR(255) NOT NULL,

    PRIMARY KEY (id),

    CONSTRAINT UQ_url UNIQUE(url)

);

DROP TABLE IF EXISTS ALERTA;

CREATE TABLE IF NOT EXISTS ALERTA (

    id int(5) NOT NULL AUTO_INCREMENT,

    descricao varchar(255) NOT NULL,

    texto VARCHAR(255) NOT NULL,

    dataHora TIMESTAMP DEFAULT CURRENT_TIMESTAMP,

    PRIMARY KEY (id)

);
```


4. Scripts de criação das Views:

```
-- View que retorna os funcionários cadastrados na aplicação

-- A partir dessa view pode-se saber quais usuários podem acessar
a aplicação web.

DROP VIEW IF EXISTS Funcionarios_Defesa_Civil_Contagem;

CREATE VIEW Funcionarios_Defesa_Civil_Contagem AS

SELECT id, nome, perfil

FROM USUARIO

WHERE perfil != 'Cidadao';


-- View que retorna os avisos pendentes.

-- Essa view facilita a obtenção dos avisos da população que
precisam de atendimento

DROP VIEW IF EXISTS Avisos_Pendentes;

CREATE VIEW Avisos_Pendentes AS

SELECT idSequencia, status, tipo, risco, bairro, logradouro

FROM aviso

WHERE status = 'Pendente';


-- View com os usuários de Contagem e com apenas campos
selecionados
```

```

DROP VIEW IF EXISTS usuarios_contagem;

CREATE VIEW

usuarios_contagem

AS

SELECT nome, cpf, telefone, email FROM usuario WHERE Lower(cidade)
= 'contagem'

```

5. Scripts de criação dos índices:

```

-- Índice que retorna os bairros que possuem avisos cadastrados
-- Esse índice permite saber quais bairros possuem ocorrências

DROP INDEX IF EXISTS aviso_bairro_IDX ON AVISO;

CREATE INDEX aviso_bairro_IDX ON AVISO (bairro);

-- Esse índice facilita as pesquisas na Tabela Usuários

ALTER TABLE `usuario`

ADD INDEX `campos_mais_usados_IDX` (`cpf`, `cidade`);

```

6. Consultas SQL ao dicionário de dados:

```

--a.

SELECT TABLE_NAME, TABLE_SCHEMA FROM INFORMATION_SCHEMA.TABLES

WHERE TABLE_SCHEMA = 'DEFESACIVILCONTAGEM';

--b.

```

```
SELECT DISTINCT COLUMN_NAME, DATA_TYPE, IS_NULLABLE,  
COLUMN_DEFAULT, TABLE_NAME FROM INFORMATION_SCHEMA.COLUMNS WHERE  
TABLE_SCHEMA = 'DEFESACIVILCONTAGEM';
```

```
--c.
```

```
SELECT DISTINCT TABLE_NAME, COLUMN_NAME, CONSTRAINT_NAME,  
REFERENCED_COLUMN_NAME, REFERENCED_TABLE_NAME
```

```
FROM INFORMATION_SCHEMA.KEY_COLUMN_USAGE
```

```
WHERE TABLE_SCHEMA = 'DEFESACIVILCONTAGEM';
```

```
--d.
```

```
SELECT DISTINCT TABLE_NAME, INDEX_NAME
```

```
FROM INFORMATION_SCHEMA.STATISTICS
```

```
WHERE TABLE_SCHEMA = 'DEFESACIVILCONTAGEM';
```

Resultados:

TABLE_NAME	TABLE_SCHEMA
IMAGENS	defesacivilcontagem
PERMISSAO	defesacivilcontagem
usuarios_contagem	defesacivilcontagem
NOTICIA	defesacivilcontagem
USUARIO	defesacivilcontagem
ATENDIMENTO	defesacivilcontagem
AVISO	defesacivilcontagem
VISTORIADOR	defesacivilcontagem
ALERTA	defesacivilcontagem
OCORRENCIA	defesacivilcontagem

COLUMN_NAME	DATA_TYPE	IS_NULLABLE	COLUMN_DEFAULT	TABLE_NAME
id	int	NO	NULL	IMAGENS
aviso_id	int	NO	NULL	IMAGENS
url	varchar	NO	NULL	IMAGENS
codigo_acesso	int	NO	NULL	PERMISSAO
modulo	varchar	NO	NULL	PERMISSAO
nome	varchar	NO	NULL	usuarios_contagem
cpf	varchar	NO	NULL	usuarios_contagem
telefone	varchar	YES	NULL	usuarios_contagem
email	varchar	NO	NULL	usuarios_contagem
id	int	NO	NULL	NOTICIA
url	varchar	NO	NULL	NOTICIA
id	int	NO	NULL	USUARIO
perfil	enum	YES	'Cidadao'	USUARIO
nome	varchar	NO	NULL	USUARIO
telefone	varchar	YES	NULL	USUARIO
cpf	varchar	NO	NULL	USUARIO
email	varchar	NO	NULL	USUARIO
dataNascimento	date	YES	NULL	USUARIO
logradouro	varchar	YES	NULL	USUARIO
numero	varchar	YES	NULL	USUARIO
complemento	varchar	YES	NULL	USUARIO
bairro	varchar	YES	NULL	USUARIO
cidade	varchar	YES	'Contagem'	USUARIO
cep	int	NO	NULL	USUARIO
id	int	NO	NULL	ATENDIMENTO
ocorrencia_id	int	NO	NULL	ATENDIMENTO
vistoriador_id	int	NO	NULL	ATENDIMENTO
id	int	NO	NULL	AVISO
idSequencia	varchar	NO	NULL	AVISO
status	enum	YES	'Pendente'	AVISO
tipo	enum	YES	'Outros/Sem dados'	AVISO
risco	enum	YES	'Inexistente ou Baixo'	AVISO
dataHora	timestamp	NO	current_timestamp()	AVISO
descricao	varchar	NO	NULL	AVISO
usuario_id	int	YES	NULL	AVISO
logradouro	varchar	NO	NULL	AVISO
numero	varchar	YES	NULL	AVISO
bairro	varchar	NO	NULL	AVISO
latitude	varchar	YES	NULL	AVISO
longitude	varchar	YES	NULL	AVISO

id	int	NO	NULL	VISTORIADOR
nome	varchar	NO	NULL	VISTORIADOR
email	varchar	NO	NULL	VISTORIADOR
id	int	NO	NULL	ALERTA
descricao	varchar	NO	NULL	ALERTA
texto	varchar	NO	NULL	ALERTA
dataHora	timestamp	NO	current_timestamp()	ALERTA
id	int	NO	NULL	OCORRENCIA
idSequencia	varchar	NO	NULL	OCORRENCIA
aviso_id	int	NO	NULL	OCORRENCIA
status	enum	YES	'Pendente'	OCORRENCIA
dataHora	timestamp	NO	current_timestamp()	OCORRENCIA
numeroDiscriminacao	varchar	NO	NULL	OCORRENCIA
dataVistoria	varchar	NO	NULL	OCORRENCIA
peessoasEnvolvidas	int	NO	NULL	OCORRENCIA
equipeVistoria	varchar	NO	NULL	OCORRENCIA
ROsAnteriores	varchar	YES	NULL	OCORRENCIA
numeroOficioEncaminhado	varchar	YES	NULL	OCORRENCIA
orgaoDestinoOficio	varchar	YES	NULL	OCORRENCIA
doacao	varchar	YES	NULL	OCORRENCIA
vitimaParcial	varchar	NO	NULL	OCORRENCIA
vitimaFatal	varchar	NO	NULL	OCORRENCIA
interdicao	varchar	NO	NULL	OCORRENCIA
dataMonitoramento	varchar	NO	NULL	OCORRENCIA
situacaoAtual	varchar	NO	NULL	OCORRENCIA
retornoVistoria	varchar	NO	NULL	OCORRENCIA
dataRequisicao	varchar	NO	NULL	OCORRENCIA

TABLE_NAME	COLUMN_NAME	CONSTRAINT_NAME	REFERENCED_COLUMN_NAME	REFERENCED_TABLE_NAME
IMAGENS	id	PRIMARY	NULL	NULL
IMAGENS	url	UQ_url	NULL	NULL
IMAGENS	aviso_id	fk_imagens_aviso_id	id	AVISO
PERMISSAO	codigo_acesso	PRIMARY	NULL	NULL
NOTICIA	id	PRIMARY	NULL	NULL
NOTICIA	url	UQ_url	NULL	NULL
USUARIO	id	PRIMARY	NULL	NULL
USUARIO	cpf	UQ_cpf_email	NULL	NULL
USUARIO	email	UQ_cpf_email	NULL	NULL
ATENDIMENTO	id	PRIMARY	NULL	NULL
ATENDIMENTO	ocorrencia_id	atendimento_ibfk_1	id	OCORRENCIA
ATENDIMENTO	vistoriador_id	atendimento_ibfk_2	id	VISTORIADOR
AVISO	id	PRIMARY	NULL	NULL
AVISO	idSequencia	UQ_id_seq	NULL	NULL
AVISO	usuario_id	fk_usuario_id	id	USUARIO
VISTORIADOR	id	PRIMARY	NULL	NULL
VISTORIADOR	nome	UQ_vistoriador	NULL	NULL
VISTORIADOR	email	UQ_vistoriador	NULL	NULL
ALERTA	id	PRIMARY	NULL	NULL
OCORRENCIA	id	PRIMARY	NULL	NULL
OCORRENCIA	idSequencia	UQ_id_seq	NULL	NULL
OCORRENCIA	aviso_id	fk_ocorrencia_aviso_id	id	AVISO

TABLE_NAME	INDEX_NAME	
IMAGENS	PRIMARY	
IMAGENS	UQ_url	
IMAGENS	fk_imagens_aviso_id	
PERMISSAO	PRIMARY	
NOTICIA	PRIMARY	
NOTICIA	UQ_url	
USUARIO	PRIMARY	
USUARIO	UQ_cpf_email	
ATENDIMENTO	PRIMARY	
ATENDIMENTO	ocorrencia_id	
ATENDIMENTO	vistoriador_id	
AVISO	PRIMARY	
AVISO	UQ_id_seq	
AVISO	fk_usuario_id	
VISTORIADOR	PRIMARY	
VISTORIADOR	UQ_vistoriador	
ALERTA	PRIMARY	
OCORRENCIA	PRIMARY	
OCORRENCIA	UQ_id_seq	
OCORRENCIA	fk_ocorrencia_aviso_id	

7. Manipulação de dados:

a. Inserções de linhas:

```
-- Inserção Usuario Teste 1
```

```
INSERT INTO usuario (nome, telefone, cpf, email, dataNascimento,
logradouro, numero, complemento, bairro, cep) VALUES
```

```
    ('Caique Ribeiro', '31999999999', '12345678912',
'caique@gmail.com', '1993-12-08', 'Rua Scania', '50', 'Casa',
'Industrial', 32997150);
```

```
-- Inserção Usuario Teste 2
```

```
INSERT INTO usuario (perfil, nome, telefone, cpf, email,
dataNascimento, logradouro, numero, complemento, bairro, cep)
VALUES
```

```
    ('Gestor', 'Samuel Silva', '31999234929', '12245328910',
'sam.silva@gmail.com', '1984-12-08', 'Rua Paraíba', '17', 'Apto
12', 'Inconfidentes', 32515072);
```

```
-- Inserção Aviso Teste 1
```

```
INSERT INTO aviso (idSequencia, tipo, descricao, usuario_id,
logradouro, bairro)
```

```
VALUES ('0001/2019', 'Inundação', 'Inundação devido a chuva', 1,
'Rua Scania', 'Industrial');
```

```
-- Inserção Aviso Teste 2
```

```
INSERT INTO aviso (idSequencia, tipo, descricao, usuario_id,
logradouro, bairro)
```

```
VALUES ('0002/2019', 'Risco construtivo', 'Estrutura de prédio com
rachaduras', 2, 'Rua das Flores', 'Centro');
```

b. Remoções de linhas:


```

-- Remoção Usuario Teste 1

DELETE FROM USUARIO WHERE nome='Caique Ribeiro';

-- Remoção Usuario Teste 2

DELETE FROM USUARIO WHERE nome='Samuel Silva';

-- Remoção Aviso Teste 1

DELETE FROM AVISO WHERE idSequencia='0001/2019';

-- Remoção Aviso Teste 2

DELETE FROM AVISO WHERE idSequencia='0002/2019';

```

c. Alterações de linhas:

```

-- Alteração Usuario Teste 1

UPDATE USUARIO SET telefone = '31986847125' WHERE id = 1;

-- Alteração Aviso Teste 1

UPDATE AVISO SET descricao = 'Situação de risco eminente aos moradores' WHERE id = 1;

```

8. Consultas:

```

--Verificação de avisos de nível Grave

SELECT numeroDiscriminacao, dataVistoria FROM OCORRENCIA

```

```

WHERE aviso_id = (SELECT id FROM AVISO WHERE risco = ('Alto'));

--Obtém as vistorias da data atual

SELECT id, nome FROM VISTORIADOR WHERE id = (SELECT vistoriador_id FROM
ATENDIMENTO WHERE ocorrencia_id = (SELECT id FROM OCORRENCIA WHERE
(
DATE(dataVistoria) = CURDATE()
)));

--Obtém as próximas vistorias agendadas

SELECT numeroDiscriminacao, dataVistoria FROM OCORRENCIA

GROUP BY status HAVING DATE(dataVistoria) > CURDATE();

--Consulta usuários residentes em casas

SELECT nome, dataNascimento, complemento FROM USUARIO

GROUP BY bairro HAVING complemento = 'Casa';

SELECT n.nome, a.id, o.dataMonitoramento

From VISTORIADOR n INNER JOIN ATENDIMENTO a ON n.id = a.vistoriador_id

INNER JOIN OCORRENCIA o ON a.ocorrencia_id = o.id;

SELECT i.url, a.descricao, u.logradouro, u.numero, u.bairro, u.cidade
FROM IMAGENS i INNER JOIN AVISO a ON i.aviso_id = a.id

INNER JOIN USUARIO u ON a.usuario_id = u.id

WHERE a.risco = ('Muito Alto' OR 'Alto');

```