

9.4 LISTA DE CODIGOS DAS FUNÇÕES, ASSERÇÕES E TRIGGERS

- TRIGGER NA TABELA PESSOA

Objetivo: Evitar que seja criada várias contas com o mesmo e-mail

```
CREATE OR REPLACE FUNCTION insert_existing_pessoa() RETURNS trigger AS $$
BEGIN
    PERFORM 1 FROM pessoa WHERE email = NEW.email;
    IF NOT FOUND THEN
        RETURN NEW;
    ELSE
        RAISE EXCEPTION 'Já existe uma pessoa com esse e-mail registrada!';
    END IF;
END $$ LANGUAGE plpgsql;
```

```
CREATE TRIGGER a_insert_existing_pessoa BEFORE INSERT ON pessoa
FOR EACH ROW EXECUTE PROCEDURE insert_existing_pessoa();
```

```
INSERT INTO Pessoa (nome_pessoa, email, login, senha, rua, numero, bairro,
L cidade, estado, cep, colaboracao, autenticidade, ativo, id_imagem_pessoa, id_pessoa)
VALUES ('Maria Silva', 'maria.silva@mail.com', 'maria.silva', '123456', 'Rua A', '54', 'Jacaraípe',
L 'Serra', 'ES', '29.000-000', 3, 3, 'S', 1, 4)

ERROR: Já existe um registro com esse e-mail!
***** Error *****

ERROR: Já existe um registro com esse e-mail!
SQL state: P0001
```

- TRIGGER NA TABELA LUGAR

```
CREATE OR REPLACE FUNCTION insert_existing_lugar() RETURNS trigger AS $$
BEGIN
    PERFORM 1 FROM lugar WHERE nome_lugar = NEW.nome_lugar;
    IF NOT FOUND THEN
        RETURN NEW;
    ELSE
        RAISE EXCEPTION 'Já existe um lugar com esse nome registrado!';
    END IF;
END $$ LANGUAGE plpgsql;
```

```
CREATE TRIGGER a_insert_existing_lugar BEFORE INSERT ON lugar
FOR EACH ROW EXECUTE PROCEDURE insert_existing_lugar();
```

```
INSERT INTO Lugar (nome_lugar, descricao, lat, long, id_pessoa, id_categoria, id_lugar)
VALUES ('Shopping Vitória', 'Shopping Center', -20.3127, -40.2878, 1, 1, 1);
```

```
ERROR: Já existe um lugar com esse nome registrado!
***** Error *****

ERROR: Já existe um lugar com esse nome registrado!
SQL state: P0001
```

- TRIGGER CATEGORIA

```
CREATE OR REPLACE FUNCTION insert_existing_categoria() RETURNS trigger AS $$
BEGIN
    PERFORM 1 FROM categoria WHERE nome_categoria =
NEW.nome_categoria;
    IF NOT FOUND THEN
        RETURN NEW;
    ELSE
        RAISE EXCEPTION 'Já existe uma categoria com esse nome
registrado!';
    END IF;
END $$ LANGUAGE plpgsql;
```

```
CREATE TRIGGER a_insert_existing_categoria BEFORE INSERT ON categoria
FOR EACH ROW EXECUTE PROCEDURE insert_existing_categoria();
```

```
INSERT INTO Categoria (nome_categoria, descricao, id_pessoa)
VALUES ('Shopping', 'Centro comercial com lojas, prestação de serviços e lazer.', 5)
```

```
ERROR: Já existe uma categoria com esse nome registrado!
***** Error *****

ERROR: Já existe uma categoria com esse nome registrado!
SQL state: P0001
```