

Requisito Funcional - Atribuir Demanda

Objetivo: Criar um endpoint em uma API REST para atribuir uma demanda a uma pessoa.

Regras de Negócio:

- **Demanda não fechada:** A demanda não pode estar fechada para ser atribuída.
- **Mesma Unidade de Negócio:** A pessoa e a demanda devem pertencer à mesma unidade de negócio.
- **Situação da Pessoa:** A situação da pessoa não pode bloquear a entrega de demandas.
- **Agendamento da Demanda:** A demanda só pode ser atribuída à pessoa para quem foi agendada.

Endpoint:

Unset

```
POST /demandas/{idDemanda}/atribuir-pessoa/{idPessoa}
```

Parâmetros:

- **idDemanda:** Identificador da demanda a ser atribuída.
- **idPessoa:** Identificador da pessoa que receberá a demanda.

Retorno:

- **Sucesso:** Código HTTP 200 OK.
- **Erro:** Código HTTP 400 Bad Request com mensagem de erro detalhada.

Exemplos de Erro:

- **Demanda não encontrada:** "A demanda não pode ser atribuída porque não foi encontrada."
- **Demanda fechada:** "A demanda não pode ser atribuída porque está fechada."
- **Unidades de negócio diferentes:** "A pessoa e a demanda não estão na mesma unidade de negócio."

- **Pessoa não encontrada:** "A demanda não pode ser atribuída porque a pessoa não foi encontrada."
- **Situação da pessoa bloqueia entrega:** "A pessoa não pode receber demandas porque sua situação bloqueia a entrega."
- **Demanda agendada para outra pessoa:** "A demanda só pode ser atribuída à pessoa para quem foi agendada."

Docker Compose:

```
Unset
version: "3.9"

services:
  db:
    image: "mysql:5.7"
    restart: "always"
    environment:
      MYSQL_ROOT_PASSWORD: "password"
    ports:
      - "3306:3306"

volumes:
  db_data:
```

Script de Criação de Tabelas:

```
Unset
CREATE TABLE IF NOT EXISTS unidade_de_negocio (
  id INT AUTO_INCREMENT,
  nome VARCHAR(255) NOT NULL,
  PRIMARY KEY (id)
);

CREATE TABLE IF NOT EXISTS situacao_pessoa (
  id INT AUTO_INCREMENT,
```

```

    nome VARCHAR(255) NOT NULL,
    bloqueia_entrega_de_demandas BOOLEAN NOT NULL,
    PRIMARY KEY (id)
);

CREATE TABLE IF NOT EXISTS pessoa (
    id INT AUTO_INCREMENT,
    id_situacao_pessoa INT NOT NULL,
    id_unidade_de_negocio INT NOT NULL,
    nome VARCHAR(255) NOT NULL,
    PRIMARY KEY (id),
    FOREIGN KEY (id_situacao_pessoa) REFERENCES situacao_pessoa
(id),
    FOREIGN KEY (id_unidade_de_negocio) REFERENCES
unidade_de_negocio (id)
);

CREATE TABLE IF NOT EXISTS demanda (
    id INT AUTO_INCREMENT,
    id_pessoa INT NULL,
    agendado_para_pessoa INT NULL,
    id_unidade_de_negocio INT NOT NULL,
    fechada BOOLEAN NOT NULL,
    PRIMARY KEY (id),
    FOREIGN KEY (id_pessoa) REFERENCES pessoa (id),
    FOREIGN KEY (id_unidade_de_negocio) REFERENCES
unidade_de_negocio (id),
    FOREIGN KEY (agendado_para_pessoa) REFERENCES pessoa (id)
);

```

Script de Inserção de Dados:

SQL

Unset

```
INSERT INTO situacao_pessoa (nome, bloqueia_entrega_de_demandas)
VALUES
    ('Disponivel', FALSE),
    ('Indisponivel', TRUE);
```

```
INSERT INTO unidade_de_negocio (nome) VALUES
    ('Unidade 1'),
    ('Unidade 2');
```

```
INSERT INTO pessoa (id_situacao_pessoa, id_unidade_de_negocio,
nome) VALUES
    (1, 1, 'João'),
    (1, 2, 'Maria'),
    (2, 1, 'Pedro');
```

```
INSERT INTO demanda (id_unidade_de_negocio, agendado_para_pessoa,
fechada) VALUES
    (1, 1, FALSE),
    (2, 2, FALSE);
```

Observações:

- O script de inserção de dados cria apenas dois exemplos de cada entidade. Você pode adicionar mais dados conforme necessário.
- Crie uma aplicação Spring Boot usando o scaffold <https://start.spring.io/>
- O código do endpoint não é fornecido, pois o objetivo do desafio é avaliar a capacidade do candidato de implementar a regra de negócio.