Objetivo

Desenvolver um micro serviço para cadastro de beneficiários e consultas, utilizando o padrão REST.

Para facilitar o setup do ambiente de banco de dados você pode rodar a imagem do postgresql através dos comandos abaixo. Mas será necessário instalar o docker no seu computador para utilizar essa imagem.

Pré requisitos:

Para facilitar a construção do backend disponibilizando um script para a criação do banco de dados em uma imagem postgresql utilizando docker.
Para fazer uso dessa imagem você precisará instalar na sua máquina o Docker e o GIT.

Passo a passo para executar o banco de dados local:

```
git clone https://github.com/gmsgilberto/proposta-teste-fullstack.git
cd proposta-teste-fullstack
docker-compose down
docker-compose build
docker-compose up
```

O Banco de dados irá iniciar na porta http://localhost:5432, e as credenciais de acesso são:
User: user
Password: pass
Database: DB

Para consulta o schema criado e as tabelas que fazem parte do modelo você pode se basear nos arquivos SQL:

```
--Tabela de Beneficicario
CREATE TABLE TbBeneficiario(
    CPF CHAR(11) NOT NULL PRIMARY KEY,
    NM_BENEFICIARIO VARCHAR(55) NOT NULL,
    DT_NASCIMENTO DATE NOT NULL
);
CREATE INDEX TBBENEFI_NOME_INDEX ON TbBeneficiario (NM_BENEFICIARIO);

--Tabela de especialidades
CREATE TABLE TbEspecialidade(
    COD_ESPECIALIDADE SERIAL NOT NULL PRIMARY KEY,
    NM_ESPECIALIDADE VARCHAR(55) NOT NULL
```

```
);
CREATE INDEX TBESPECIALIDADE_NOMEINDEX ON TbEspecialidade (NM_ESPECIALIDADE);

--Tabela de consultas
CREATE TABLE TbConsulta(
    COD_CONSULTA SERIAL PRIMARY KEY,
    COD_ESPECIALIDADE INTEGER NOT NULL,
    CPF_BENEFICIARIO CHAR(11) NOT NULL,
    DATA_CONSULTA DATE NOT NULL,
    HORA_CONSULTA CHAR(5) NOT NULL,
    FOREIGN KEY (COD_ESPECIALIDADE) REFERENCES TbEspecialidade(COD_ESPECIALIDADE),
    FOREIGN KEY (CPF_BENEFICIARIO) REFERENCES TbBeneficiario(CPF)
);
CREATE INDEX TBCONSULTA_INDEX1 ON TbConsulta (COD_ESPECIALIDADE,CPF_BENEFICIARIO);
CREATE UNIQUE INDEX TBCONSULTA_UQ_AGENDAMENTO ON TbConsulta
(CPF_BENEFICIARIO,DATA_CONSULTA,HORA_CONSULTA);
```

Script de carga inicial

```
--Popula a tabela de especialidades
INSERT
  INTO TbEspecialidade(COD_ESPECIALIDADE, NM_ESPECIALIDADE)
VALUES (1,'Clínica Geral');
INSERT
  INTO TbEspecialidade(COD_ESPECIALIDADE, NM_ESPECIALIDADE)
VALUES (2,'Oftalmologista');
INSERT
  INTO TbEspecialidade(COD_ESPECIALIDADE, NM_ESPECIALIDADE)
VALUES (3,'Ginicologista');
INSERT
  INTO TbEspecialidade(COD_ESPECIALIDADE, NM_ESPECIALIDADE)
VALUES (4,'Urologista');

-- Popula a tabela de beneficiarios
INSERT
  INTO TbBeneficiario(CPF, NM_BENEFICIARIO, DT_NASCIMENTO)
VALUES ('12345678901','ANA DAS QUANTAS', '1980-05-29');

INSERT
  INTO TbBeneficiario(CPF, NM_BENEFICIARIO, DT_NASCIMENTO)
```

```
VALUES ('23456789011','JOSE DAS NEVES', '1990-12-15');



INSERT
  INTO TbConsulta(COD_ESPECIALIDADE, CPF_BENEFICIARIO, DATA_CONSULTA,HORA_CONSULTA)
VALUES ('1','23456789011', '2023-12-15', '15:00');

INSERT
  INTO TbConsulta(COD_ESPECIALIDADE, CPF_BENEFICIARIO, DATA_CONSULTA,HORA_CONSULTA)
VALUES ('2','23456789011', '2023-12-15', '16:00');

INSERT
  INTO TbConsulta(COD_ESPECIALIDADE, CPF_BENEFICIARIO, DATA_CONSULTA,HORA_CONSULTA)
VALUES ('3','23456789011', '2023-12-15', '17:00');

INSERT
  INTO TbConsulta(COD_ESPECIALIDADE, CPF_BENEFICIARIO, DATA_CONSULTA,HORA_CONSULTA)
VALUES ('4','23456789011', '2023-12-15', '18:00');

 SELECT consulta.COD_CONSULTA,
        especialidade.COD_ESPECIALIDADE,
        especialidade.NM_ESPECIALIDADE,
        consulta.DATA_CONSULTA,
        consulta.HORA_CONSULTA,
        consulta.CPF_BENEFICIARIO,
        beneficiario.NM_BENEFICIARIO,
        beneficiario.DT_NASCIMENTO
   FROM TbBeneficiario beneficiario
          INNER JOIN TbConsulta consulta
                  ON consulta.CPF_BENEFICIARIO = beneficiario.CPF
          INNER JOIN TbEspecialidade especialidade
                  ON especialidade.COD_ESPECIALIDADE = consulta.COD_ESPECIALIDADE;
```

Funcionalidades

1) **Criar o endpoint [POST] /reset**
   Objetivo: excluir os registros de beneficiários e consultas;
   Response: 200 OK

2) Criar o endpoint para cadastro de um beneficiário:
Request: [POST] /beneficiarios

```
{
    "cpf": "string",
    "nome": "string",
    "data-nascimento": "2023-06-27"
}
```

Response: 201 OK

3) Criar o endpoint para consulta de beneficiarios com base no ID
Request: [GET] /beneficiarios/{cpf}

```
Responses:
  HTTPCODE 200
    {
        "cpf": "33654725848",
        "nome": "Gilberto",
        "dataNascimento": "2023-06-27"
    }
  HTTPCODE 404 "Beneficiario nao cadastrado"
```

4) Criar um endpoint para listar as especialidades cadastradas
Request: [GET] /especialidades
Response: 200

```
[
    {
      "codigo-especialidade": 1,
      "nome-especialidade": "Clínica Geral"
    },
    {
      "codigo-especialidade": 4,
      "nome-especialidade": "Urologista"
    },
    ...
  ]
```

5) Criar um endpoint para cadastro de Consultas
Request: [POST] /consultas

```
Exemplo:
{
    "cpf-beneficiario": "33674878974",
    "nome-especialidade": "Urologista",
    "data": "2023-12-15",
    "hora": "17:30"
}
```

Response: 201 OK

Diferenciais

1) Utilizar princípios SOLID
2) Cobertura de testes unitários acima de 80%