

# Recomendador de amigos

---

Você deve implementar um sistema que permita que uma pessoa obtenha sugestões de novos amigos se baseando nas amizades já existentes. Você deve criar uma aplicação Node.js que escute a porta 3000. Armazene dados em memória durante a execução do programa (não utilize nenhum banco de dados externo, utilize variáveis globais, não serão aceitos testes que dependam da instalação de softwares externos para armazenar os dados durante a execução) e implemente as seguintes rotas:

## Create Person - [POST] <http://localhost:3000/person>

Esta rota deve receber um CPF e um nome, e realizar o cadastro do usuário. Deve retornar erro com status code 400 caso o usuário já esteja cadastrado ou o CPF informado tenha tamanho diferente de 11 dígitos numéricos (Não implemente qualquer algoritmo validador de CPF).

entrada:

```
{
  "cpf": "12345678909",
  "name": "Joaozinho"
}
```

saída:

- Deve retornar código HTTP 200 em caso de sucesso.
- Deve retornar código HTTP 400 caso o usuário cadastrado já exista ou caso o CPF informado não consista de 11 dígitos numéricos

## Get Person - [GET] <http://localhost:3000/person/:CPF>

Esta rota deve receber um CPF e, se o usuário existir, retornar seus dados (nome e CPF), caso contrário, deve retornar erro com status code 404.

saída:

```
{  
  "cpf" : "12345678909",  
  "name" : "Joaozinho"  
}
```

### **Clean - [DELETE] <http://localhost:3000/clean>**

Esta rota deve limpar todos os dados (pessoas e relacionamentos) em memória.

### **Create Relationship - [POST] <http://localhost:3000/relationship>**

Esta rota deve receber dois CPFs e, caso os dois usuários existam, criar um relacionamento entre eles, caso contrário, deve retornar erro com status code 404.

entrada:

```
{  
  "cpf1" : "11111111111",  
  "cpf2" : "22222222222"  
}
```

saída:

- Deve retornar código HTTP 200 em caso de sucesso.
- Deve retornar código HTTP 404 caso um dos usuários não exista

### **Get Recommendations - [GET] <http://localhost:3000/recommendations/:CPF>**

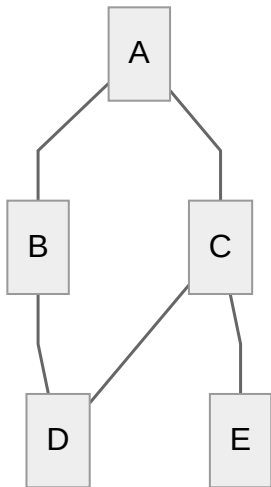
Deve receber um CPF e retornar erro com status code 400 se o CPF informado não consistir em 11 dígitos numéricos, erro com status code 404 se o usuário correspondente não existir. Caso o CPF corresponda a um usuário cadastrado, o retorno deve ser um Array contendo a lista de CPFs de todos os amigos dos amigos do usuário informado que não são seus amigos, ordenada de maneira decrescente pela relevância, ou seja, deve-se verificar quantos amigos tem relacionamento com a pessoa, e as pessoas com mais relacionamentos com amigos devem ser

informados primeiro. Apenas amigos dos amigos devem ser listados, e devemos ignorar casos nos quais a pontuação é zero.

Retorno:

```
[  
    "1111111111", "2222222222"  
]
```

Exemplo:



Neste caso, assumindo que a entrada seja A, o retorno deve ser exatamente [D, E], nesta ordem, pois 2 amigos de A tem amizade com D, e apenas um amigo de A tem amizade com E.

Seu projeto deve conter um README explicando como executá-lo.