

## Desafio

A partir dos dados mais recentes do CENSO, o IBGE lançou uma aplicação Web que fez muito sucesso em 2016: o site Nomes no Brasil (<https://www.ibge.gov.br/censo2010/apps/nomes>), que traz um ranking iterativo dos nomes mais populares que pode ser organizado em décadas e/ou estados do país. Além disso você pode buscar pelo seu nome e entender quão frequente é esse nome em um estado ou uma cidade do país. Algumas informações podem ser separadas ainda pela década ou pelo sexo atribuído ao nome.

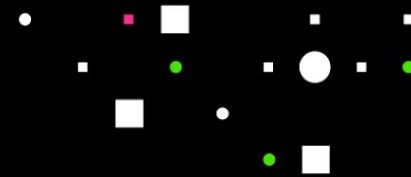
Todas essas informações, tanto as cidades/regiões/estados quanto a frequência da presença de nomes são oferecidas via APIs públicas do IBGE. E nosso desafio vai usar essas APIs para criar uma nova versão dessa app, uma versão *old school*. Então prepare seus conhecimentos em Ruby puro e vamos lá!

O desafio é composto de 3 sub-desafios, interligados e sequenciais. Fique a vontade para distribuir os esforços em cada um dos sub-desafios, ou seja, você pode ir fazendo os 3 aos poucos ou só mudar de etapa quando considerar que a anterior está completa. Essa decisão fica a seu critério.

### Desafio 1

A primeira etapa do desafio consta na criação de uma aplicação Ruby para uso no terminal. Ao ser executada essa aplicação deve oferecer ao seu usuário as seguintes opções de consulta:

1. Ranking dos nomes mais comuns em uma determinada Unidade Federativa (UF)
  - a. Listar todas UFs e obter a sigla da UF que o usuário deseja consultar
  - b. Exibir o resultado em 3 tabelas:
    - i. Uma para o ranking geral
    - ii. Outras duas tabelas separando os resultados para cada sexo
2. Ranking dos nomes mais comuns em uma determinada cidade
  - a. Obter o nome de uma cidade no terminal
  - b. Exibir o resultado em 3 tabelas:
    - i. Uma para o ranking geral
    - ii. Outras duas tabelas separando os resultados para cada sexo
3. Frequência do uso de um nome ao longo dos anos
  - a. Solicitar um ou mais nomes (separados sempre por vírgula)



- b. Exibir uma tabela única, onde cada linha representa uma década e cada coluna representa um dos nomes informados na etapa anterior

Além da qualidade e organização do seu código, vamos avaliar também a facilidade de uso das interfaces e a qualidade da exibição dos resultados de cada consulta. Então dedique algum tempo para formatar minimamente as tabelas solicitadas.

A sua aplicação pode ser executada de duas formas:

1. Como uma aplicação que fica em execução de forma constante e conta com um menu para o usuário escolher a consulta a ser realizada ou sair da aplicação.
2. Como um CLI, similar a comandos como ``git`` e ``rails`` que você executa em seu terminal. Nesse formato você deve ter uma opção ``--help`` que traz todas as opções possíveis para execução (todas as consultas e como o usuário deve passar os parâmetros necessários em cada consulta)

## Desafio 2

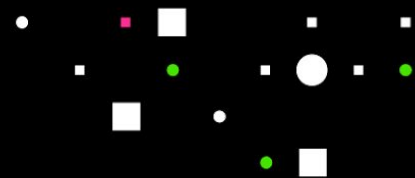
Nesta etapa, queremos otimizar nossa aplicação. As UFs e as cidades do país raramente sofrem alterações, então não faz muito sentido consultar a API de localidades várias vezes para exibir o resultado, especialmente nas consultas 1 e 2 da etapa anterior.

Nessa etapa você deve:

1. Criar uma estrutura de conexão a um banco de dados SQLite em sua aplicação
2. Ao iniciar sua aplicação pela primeira vez, você deve criar e popular tabelas para armazenar todas as UFs e todos os municípios, consultando a API de localidades. Lembre-se de armazenar somente os dados essenciais para o uso dentro da sua aplicação. A organização das tabelas fica a seu critério.
3. Ao realizar as operações que demandam o uso de códigos de UFs e/ou cidades, utilizar os dados vindos do banco de dados ao invés da API de localidades.

## Desafio 3

A última etapa traz dados da população brasileira residente em 2019 através de um arquivo CSV. Queremos incrementar os resultados das consultas da etapa 1, adicionando em todas as tabelas exibidas para o usuário, uma nova coluna contendo o percentual que aquele resultado representa dentro do universo escolhido.



Exemplo: Ao consultar o ranking de nomes de mulheres em Angra dos Reis, encontramos no topo o nome “Maria” com 7.820 pessoas. No CSV com dados da população vimos que Angra dos Reis possui 203.785 habitantes. Então deveríamos acrescentar uma nova coluna ao resultado indicando que as 7.820 pessoas chamadas de Maria representam aproximadamente 3,8% da população da cidade.

Considerações:

1. O arquivo CSV deve ser armazenado junto com sua aplicação em um diretório a seu critério
2. A leitura do arquivo pode ser feita em tempo real, de acordo com as consultas realizadas ou, pode ser incluído como parte do processo de inicialização do banco de dados. No CSV as UFs e municípios possuem o mesmo código utilizado pela API de localidades do IBGE.

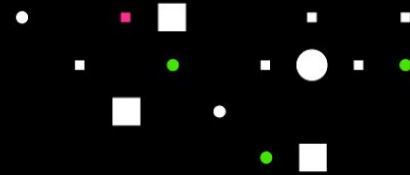
## Observações Finais

Esse projeto possui desafios em temas variados, partindo de lógica e organização de código até banco de dados, leitura de arquivos e consultas a APIs. Por isso veja algumas dicas que podem te ajudar:

1. Não existe uma única ordem ou forma correta para resolver todos os problemas propostos como foi dito no começo deste texto. O que recomendamos é:
  - Leia toda a proposta mais de uma vez, fazendo anotações ao seu jeito;
  - Faça consultas nas APIs do IBGE usando seu navegador, ou aplicações como curl ou Postman, para conhecer melhor como elas funcionam;
  - Abra o arquivo CSV da população para conhecer sua estrutura e seus dados.

Ao final de tudo isso, faça um planejamento e comece a escrever seu código.

2. Não esqueça de criar um repositório Git em alguma plataforma online e compartilhar conosco assim que possível, enviando um email para [treinamentos@campuscode.com.br](mailto:treinamentos@campuscode.com.br) com o título: Desafio – Programa de Onboarding. Queremos acompanhar o seu trabalho e tentaremos ajudar com dicas unindo informações de todos participantes.
3. Documente no arquivo README.md, na raiz do seu projeto, como funciona sua aplicação e decisões tomadas durante a implementação. Não deixe de citar `gems` e outros recursos utilizados para resolver os problemas que encontrou ao longo do projeto.
4. Vamos criar um canal privado no Slack da Campus Code e reunir, além de todos participantes, alguns devs da Rebase para tirar suas dúvidas. Então voltem a



acessar e ficar online por lá. Vamos usar o mesmo usuário que você utilizou durante o treinamento, caso isso seja um problema e você precise criar uma nova conta nos avise também.

## Referências

API de Localidades: <https://servicodados.ibge.gov.br/api/docs/localidades?versao=1>

API de Nomes: <https://servicodados.ibge.gov.br/api/docs/censos/nomes?versao=2>

CSV com dados da população: <https://we.tl/t-zJ836FXXC6> ou <https://gofile.io/?c=BfE3dR>