



## Objetivos

Nos exercícios a seguir, vamos realizar dois scripts em Bash.

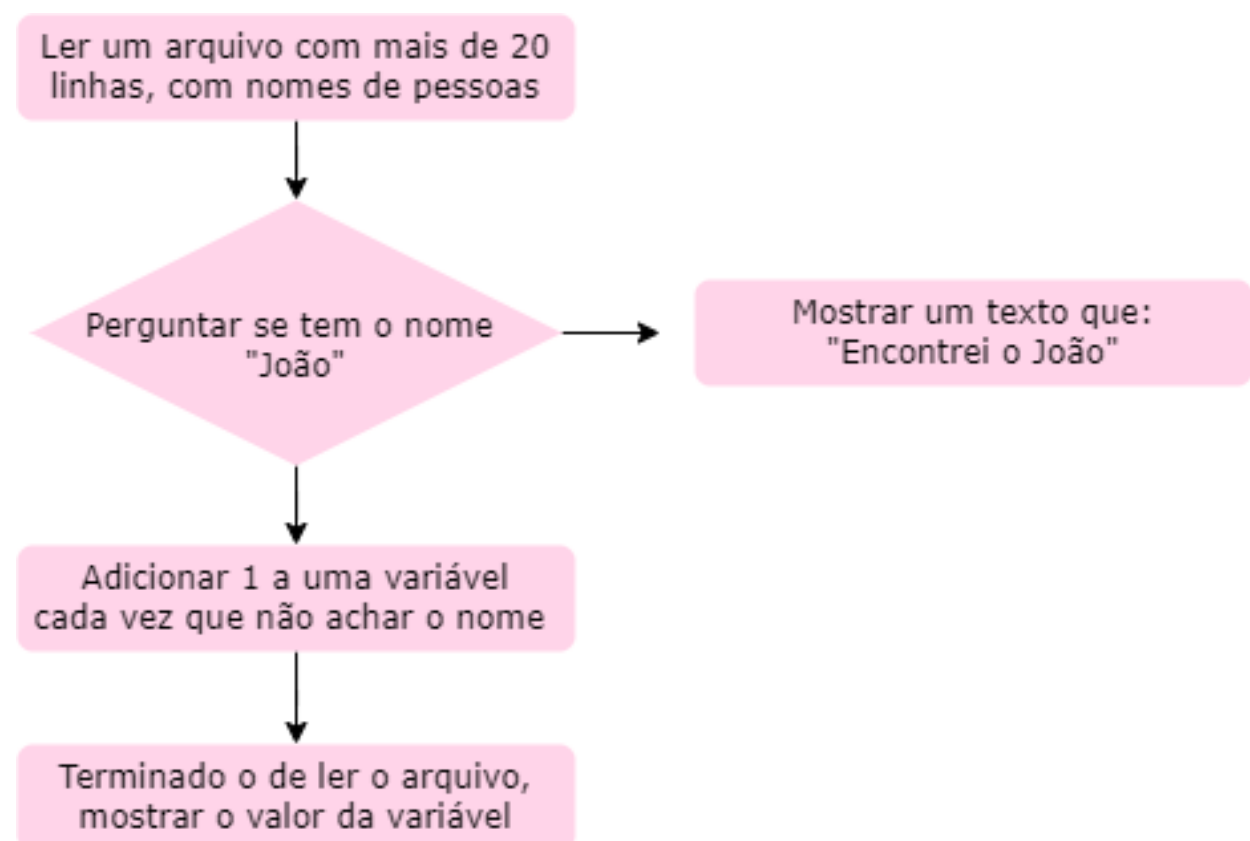
## O que devemos ter?

Uma lista de serviços web gratuitos que vamos usar para o exercício 2.

## Instruções

### Exercício 1

Individualmente, criamos um script em Bash que deve fazer o seguinte:



Os passos que devemos seguir para preparar nosso exercício são:

- Criar em primeiro lugar nosso arquivo, podemos usar nano e copiar para lá, ou com a instrução `cat` escrevemos os nomes 1 a 1, vamos chamá-lo de **lista\_nomes**.
- Antes de iniciar nosso script, podemos testar o que aprendemos na aula sobre os comandos `for`, `do` e `done`, criando um script menor que contenha o seguinte:

```
#!/bin/bash
for word in $(cat lista_nomes)
do
    echo "O nome é $word"
done
```

Como resultado, devemos percorrer o arquivo e mostrá-lo linha por linha.

Sabendo disso, agora o que vamos fazer é adicionar a lógica de controle, para isso devemos usar uma instrução `if-else`, onde vamos comparar o valor da variável **\$word** com o texto "João", essa parte deve ser codificado.

```
if [$word = "João"]; then
    echo "Encontrei a $word"
else
    let a = $a + 1
fi
```

Tendo essas duas partes, prosseguimos para completar o exercício, não esquecendo de declarar a variável **a** no script final.

## Exercício 2

Individualmente, vamos realizar, dependendo do nosso arquivo de nomes, um script que nos mostre:

- O gênero desse nome.
- O código do país onde é mais popular.

Para obter esses dois dados, usaremos essas duas APIs de referência.

## API GENDERIZE

<https://api.genderize.io/?name=emilio>

Ao passar o nome como parâmetro, ele retorna um JSON que dá um resultado como o seguinte:

```
{"name": "emilio", "gender" : "male", "probability": 0,99,  
"count": 25883}
```

De nosso interesse é o valor do atributo gender.

## API NACIONALIZE

<https://api.nationalize.io/?name=emilio>

Ao passar o nome como parâmetro, ele retorna um JSON que dá um resultado como o seguinte:

```
{"name": "emilio", "country": [{"country_id": "ES", "probability":  
0.13849973696 268725 } , {"country_id": "MZ", "probability":  
0.10571390688819501}, {"country_id": "CL", "probability":  
0.08975185148274877}]}
```

De nosso interesse é o valor do primeiro atributo country\_id do objeto country.

Podemos usar parte do nosso script anterior como base, pois a lista de nomes a ser percorridos é a mesma, o que muda neste exercício é obter os dados do webservice, considerando que o parâmetro para cada requisição será dinâmico.

Vejamos um exemplo de como fazemos uma chamada assumindo que o nome a descobrir é Emilio.

```
#!/bin/bash  
nome="Emilio"  
curl -s https://api.genderize.io/?name=$nome | jq '.gender' |  
{read -r gen; echo "Sexo de $nome é: $ gen"; }  
curl -s https://api.nationalize.io/?name=$nome |  
jq '.country[0].country_id' | {read -r cn; echo "País do $nome  
é: $ cn"; }
```

Vamos observar algumas novidades, por exemplo, usamos o modificador **-s** no curl, o que faz com que ele não mostre os dados do download e também, depois do jq vem o comando read, com o modificador **-r**, que permite o comando a capturar a saída do **jq** e levá-la para uma variável, na primeira requisição é a variável **gen** e na segunda requisição é a variável **cn**.

De posse deste script, podemos agora integrá-lo com parte do anterior, completar a realização do exercício.