



# Certified Tech Developer

The Ultimate Degree

## Infraestrutura I

### Objetivos

No exercício a seguir, vamos fazer um script um pouco mais complexo. O idioma é escolhido por você (Bash ou PowerShell).

### O que devemos ter?

Uma lista de nomes que vamos usar com os serviços web (APIs) gratuitos para realizar o Exercício 2 (script trabalhado na mesa da aula anterior).

Genderize

Nationalize

Um arquivo hospedado no GitHub com uma lista de 500 nomes:

<https://raw.githubusercontent.com/nidiodolfini/digitalhouse/main/nomes-proprios.txt>

### API Genderize

Esta API (<https://genderize.io>) nos permite obter um resultado em formato JSON com informações relacionadas ao gênero de um determinado nome. Por exemplo, para o nome "Emilio", o resultado obtido é:

```
{
  "name": "emilio",
  "gender": "male",
  "probability": 0.99,
  "count": 25883
}
```

As propriedades do objeto JSON que obtemos são:

**name:** o nome pelo qual consultamos é incluído na resposta.

**gender:** o gênero do nome pelo qual consultamos.

**probability:** probabilidade de que a previsão de gênero esteja correta.



**count:** número de vezes que esse nome foi consultado.

Para obter o resultado indicado anteriormente, a chamada a executar é uma chamada do tipo GET incluindo o parâmetro 'name' para a URL: <https://api.genderize.io/>. Por exemplo: <https://api.genderize.io/?name=emilio>

**Dica:** Para o exercício, a propriedade que nos interessa é a propriedade 'gender'.

**Importante:** a API tem um limite de 1.000 chamadas por dia. Se você usar mais de 1000 (mil) vezes no mesmo dia, terá que esperar 24 horas para usá-lo novamente.

### API de nacionalização

Esta API (<https://nationalize.io/>) nos permite prever a nacionalidade de uma pessoa com base em seu nome. O resultado é entregue no formato JSON. Por exemplo, para o nome "Emilio", o resultado obtido é:

```
{
  "name": "emilio",
  "country":
  [
    {
      "country_id": "ES",
      "probability": 0.13849973696268725
    },
    {
      "country_id": "MZ",
      "probability": 0.10571390688819501
    },
    {
      "country_id": "CL",
      "probability": 0.08975185148274877
    }
  ]
}
```

As propriedades do objeto JSON que obtemos são:

**name:** o nome pelo qual consultamos é incluído na resposta.

**country:** contém uma matriz de objetos para cada um dos países onde esse nome é potencialmente usado, classificados do mais provável para o menos provável.

Cada um inclui as seguintes propriedades:

**country\_id** – O código do país baseado no padrão ISO 3166.

**probabilidade** – A probabilidade de que o nome corresponda a uma pessoa desse país.

Para obter o resultado indicado anteriormente, temos que executar uma chamada do tipo GET incluindo o parâmetro 'name' para a URL: <https://api.nationalize.io/>. Por exemplo: <https://api.nationalize.io/?name=emilio>

**Dica:** Para o exercício, a propriedade que nos interessa é a propriedade 'country\_id' do objeto que provavelmente representa o país.

**Importante:** a API tem um limite de 1.000 chamadas por dia. Se você usar mais de 1000 (mil) vezes no mesmo dia, terá que esperar 24 horas para usá-lo novamente.

## Instruções

Individualmente, execute os seguintes passos:

A partir do arquivo que contém os 500 nomes, selecione aleatoriamente 5 nomes diferentes que começam com "A", 5 nomes diferentes que começam com "L" e 5 nomes diferentes que nem comecem com "A." ", nem com "L".

Exemplo de lista válida:

Alonso  
Alfonso  
Alfredo



Agustín  
Ananías  
Lidia  
Lino  
Lorena  
Lorenzo  
Leonor  
Montserrat  
Patricio  
Porfirio  
Irene  
Vicente

Então, para cada um dos nomes obtidos do arquivo, submeta-os às duas APIs (Genderize e Nationalize) e imprima uma mensagem na tela para cada Nome indicando sexo e nacionalidade.

**Dica:** no **PowerShell** considere essas linhas, certifique-se de entender o que elas fazem para você poder incluí-las e/ou adaptá-las ao seu script. Substitua o texto em vermelho pelo endereço web do arquivo.

```
$req = Invoke-WebRequest -Method Get -Uri <urlToList>  
$namesWithA = $req.Content.split("`n") | Where-Object { $_ -like  
"A*" }  
  
1..5 | ForEach-Object{  
    $random = Get-Random -Minimum 0 -maximum $ ($namesWithA.count-1)  
    $namesWithA[$random]  
}
```



**Dica:** em **Bash** considere essas linhas, certifique-se de entender o que elas fazem para que você possa incluí-las e/ou adaptá-las ao seu script. Substitua o texto em vermelho pelo endereço web do arquivo.

```
req=`curl <urlToList> | shuf`  
  
countA=0  
  
for name em $req  
do  
    if [[ $name == A* ]] && [ $countA -le 4 ] ;  
    then  
        echo $name  
        let "countA++"  
    fi  
done
```