Certified Tech Developer
The Ultimate Degree

Objetivos

En los siguientes ejercicios vamos a realizar dos scripts en PowerShell. Estos mismos problemas ya los resolvieron en Bash, el objetivo es entender cómo podemos usar una herramienta diferente para resolver el mismo problema.

¿Qué recibimos?

Un listado de web services o APIs libres que vamos a utilizar para el ejercicio 2.

- Genderize
- Nationalize

API GENDERIZE (https://genderize.io)

Esta API nos permite obtener un resultado en formato JSON con información relacionada al género de un determinado nombre, por ejemplo, para el nombre “Emilio”, el resultado que se obtiene es:

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

Las propiedades del objeto Json que obtenemos son:

- **name**: el nombre por el que consultamos es incluido en la respuesta.
- **gender**: el género del nombre por el que consultamos.
- **probability**: probabilidad en que la predicción de género sea correcta.
- **count**: cantidad de veces que ese nombre ha sido consultado.

Para obtener el resultado previamente indicado, la llamada a ejecutar es una llamada tipo GET incluyendo el parámetro 'name' a la URL: https://api.genderize.io/. Por ejemplo: https://api.genderize.io/?name=emilio

> **Tip:** Para el ejercicio 2, la propiedad que nos interesa es la propiedad 'gender'.

> **¡Importante!** La API tiene un límite de 1000 llamadas diarias. De hacer uso de la misma más de 1000 (mil) veces en un mismo día, deberemos esperar 24hs para volver a utilizarla.

API NATIONALIZE (https://nationalize.io/)

Esta API nos permite predecir la nacionalidad de una persona en función de su nombre. El resultado es entregado en formato JSON, por ejemplo, para el nombre "Emilio", el resultado que se obtiene es:


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

Las propiedades del objeto Json que obtenemos son:

- **name**: el nombre por el que consultamos es incluido en la respuesta.
- **country**: contiene un array de objetos por cada uno de los países en donde potencialmente se usa ese nombre, ordenado de mayor probabilidad a menor. Cada uno incluye las siguientes propiedades:
    - **country_id**: el código del país basado en la normativa ISO 3166.
    - **probability**: la probabilidad de que el nombre corresponda a una persona de ese país.

Para obtener el resultado previamente indicado, la llamada a ejecutar es una llamada tipo GET incluyendo el parámetro 'name' a la URL: https://api.nationalize.io/. Por ejemplo: https://api.nationalize.io/?name=emilio

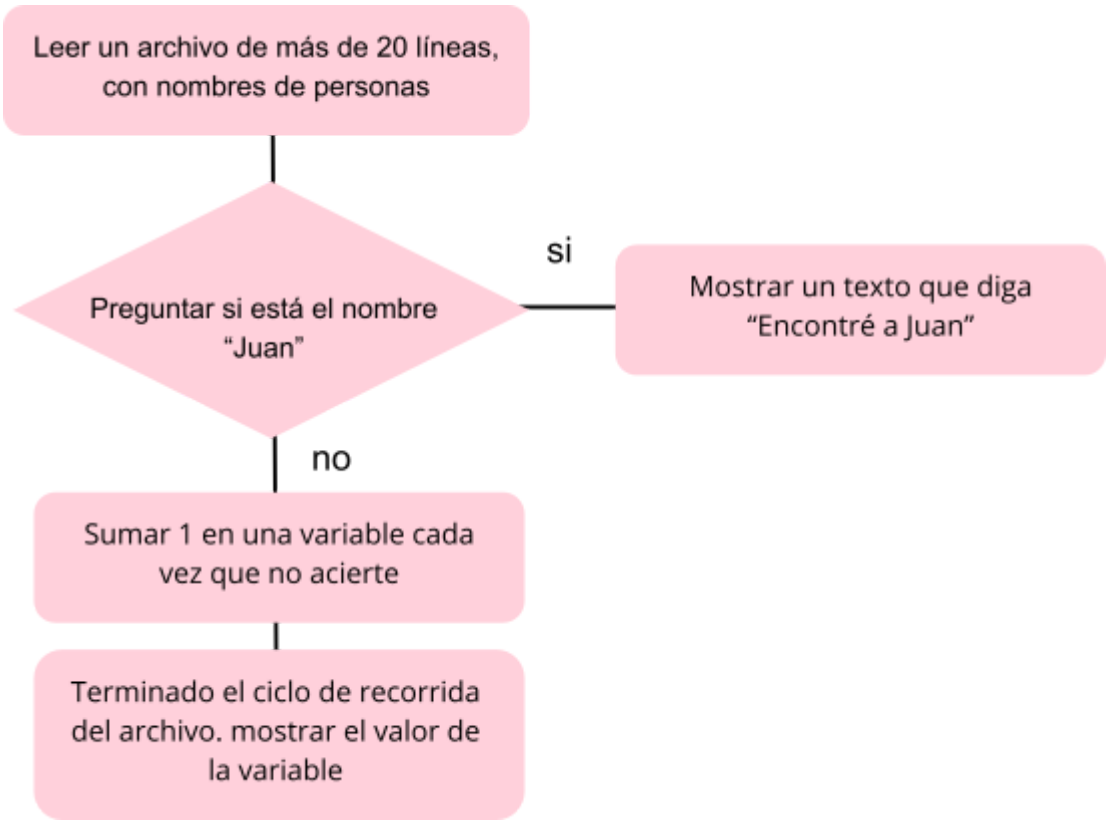
> **Tip:** Para el ejercicio 2, la propiedad que nos interesa es la propiedad 'country_id' del objeto que representa al país con mayor probabilidad.

> **¡Importante!** La API tiene un límite de 1000 llamadas diarias. De hacer uso de la misma más de 1000 (mil) veces en un mismo día, deberemos esperar 24hs para volver a utilizarla.

Instrucciones

Ejercicio 1

De forma individual realizamos un script en Bash el cual debe implementar el siguiente flujo:



Para lograrlo, debemos ejecutar los siguientes pasos::

1) Crear en primer lugar nuestro archivo, podemos usar Visual Studio Code e ingresamos una lista de nombres. Finalmente, guardamos el archivo con el nombre 'lista_nombres.txt'.

Tip: Antes comenzar a escribir el script, podemos probar el comando 'foreach', realizando un script más pequeño que contenga las siguientes líneas:

```
foreach($nombre in $(Get-Content -Path .\lista_nombres.txt)) {
      Write-Output "El nombre es $nombre"
}
```

Como resultado nos debería recorrer el archivo y mostrar los nombres línea por línea anteponiendo el mensaje "El nombre es" a cada ítem dentro del archivo.

2) Sabiendo esto, ahora lo que vamos a realizar es el agregado de la lógica de control, para ello debemos utilizar una sentencia 'if-else', en donde vamos a comparar el valor de la variable '$nombre' con el texto "Juan", esa parte debería codificarse de la siguiente manera:

```
if ($nombre -eq "Juan") {
  Write-Output "Encontré a $nombre"
} else {
   $otrosNombres++
}
```

Teniendo estas dos partes, procedemos a completar la ejercitación, no olvidar declarar la variable 'otrosNombres' en el script final.

Ejercicio 2

De forma individual, vamos a escribir un script que nos permita verificar algunos datos para cada uno de los nombres que incluimos en el archivo 'lista_nombres.txt'. Los datos que vamos a verificar son:

- El género de ese nombre
- El codigo de pais en donde es más popular

1) Podemos usar como base parte de nuestro script anterior, ya que la lista a recorrer de los nombres es la misma. Sin embargo, tendremos que modificar el script para consumir las APIs correspondientes, teniendo en cuenta que el parámetro para cada solicitud deberá ser dinámico.

2) Vamos a ver un ejemplo de cómo hacemos una llamada suponiendo que el nombre a averiguar es Emilio.

```
$nombre = "Emilio"
```

```
$gender = Invoke-RestMethod -Method Get -Uri
https://api.genderize.io/?name=$nombre | Select-Object -ExpandProperty
Gender
$country = Invoke-RestMethod -Method Get -Uri
https://api.nationalize.io/?name=$nombre | Select-Object -ExpandProperty
Country | Select-Object -First 1 -Property country_id

Write-Output "El género de $nombre es: $gender"
Write-Output "El país de $nombre es: $($country.country_id)"
```

3) Teniendo este ejemplo, ya podemos integrarlo como parte del script anterior para completar la ejercitación.