



Infraestructura I

Objetivos

En los siguientes ejercicios vamos a realizar dos scripts en Bash.

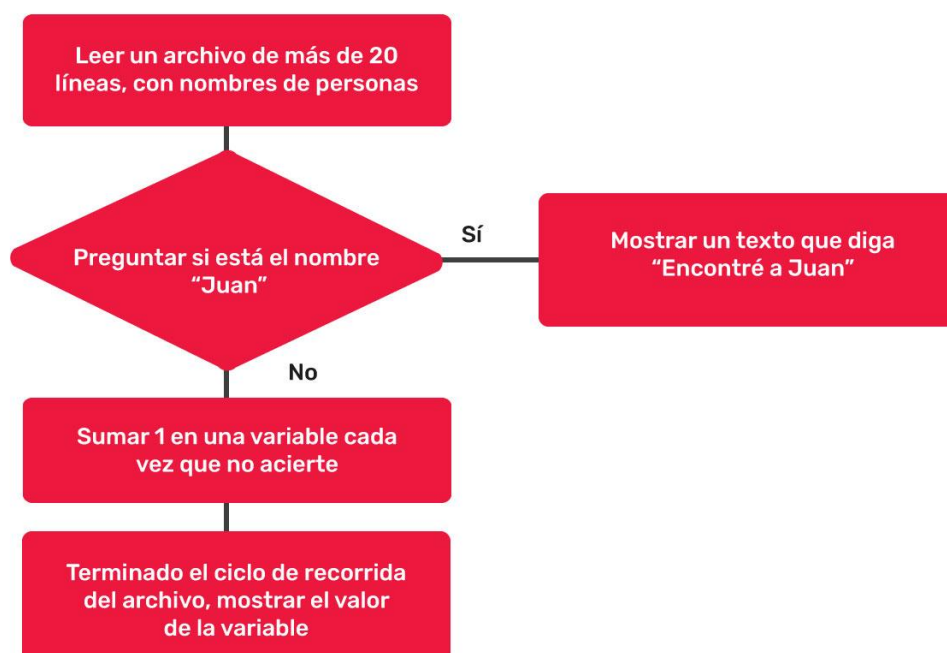
¿Qué recibimos?

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

Instrucciones

Ejercicio 1

De forma individual, realizamos un script en Bash que debe realizar lo siguiente:





Los pasos que debemos seguir para poder preparar nuestro ejercicio son:

- Crear en primer lugar nuestro archivo, podemos usar nano y allí copiamos, o con la instrucción cat vamos escribiendo 1 a 1 los nombres, lo llamaremos lista_nombres.
- Antes comenzar nuestro script, podemos probar lo aprendido en clase con respecto al comando for, do y done, realizando un script más pequeño que contenga lo siguiente:

```
#!/bin/bash
for word in $(cat lista_nombres)
do
    echo "El nombre es $word"
done
```

Como resultado nos debería recorrer el archivo y mostrarlo línea por línea.

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 \$word con el texto "Juan", esa parte debería codificarse.

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

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



Ejercicio 2

De forma individual, vamos a realizar es, en función de nuestro archivo de nombres, realizar un script que nos muestre:

- El género de ese nombre.
- El código de país en donde es más popular.

Para obtener estos dos datos, vamos a usar estas dos API de referencia.

API GENDERIZE

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

En donde pasando como parámetro el nombre, nos devuelve un JSON el cual da un resultado como el siguiente:

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

 De

nuestro interés es el valor del atributo gender.

API NATIONALIZE

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

En donde pasando como parámetro el nombre, nos devuelve un JSON el cual da un resultado como el siguiente:

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

De nuestro interés es el valor del primer atributo country_id del objeto country.

Podemos usar como base parte de nuestro script anterior, ya que la lista a



recorrer de los nombres es la misma, lo que cambia en esta ejercitación es la obtención del dato desde el webservice, teniendo en cuenta que el parámetro para cada solicitud será dinámico.

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

```
#!/bin/bash
nombre="Emilio"
curl -s https://api.genderize.io/?name=$nombre | jq '.gender' |
{ read -r gen; echo "Gender of $nombre is: $gen"; } curl -s
https://api.nationalize.io/?name=$nombre | jq
'.country[0].country_id' | { read -r cn; echo "Country of
$nombre is: $cn"; }
```

Observaremos algunas cosas nuevas, por ejemplo, usamos el modificador `-s` en `curl`, lo que hace que no muestre los datos de descarga y además, luego del `jq` está el comando **read**, con el modificador `-r`, lo que permite ese comando es capturar la salida de `jq` y llevarlo a una variable, en la primer solicitud es la variable **gen** y en la segunda solicitud es la variable **cn**.

Teniendo este script, ya podemos integrarlo con parte del anterior, en función de completar la ejercitación.