



**Certified Tech
Developer**

The Ultimate Degree

Infraestructura I

Objetivos

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

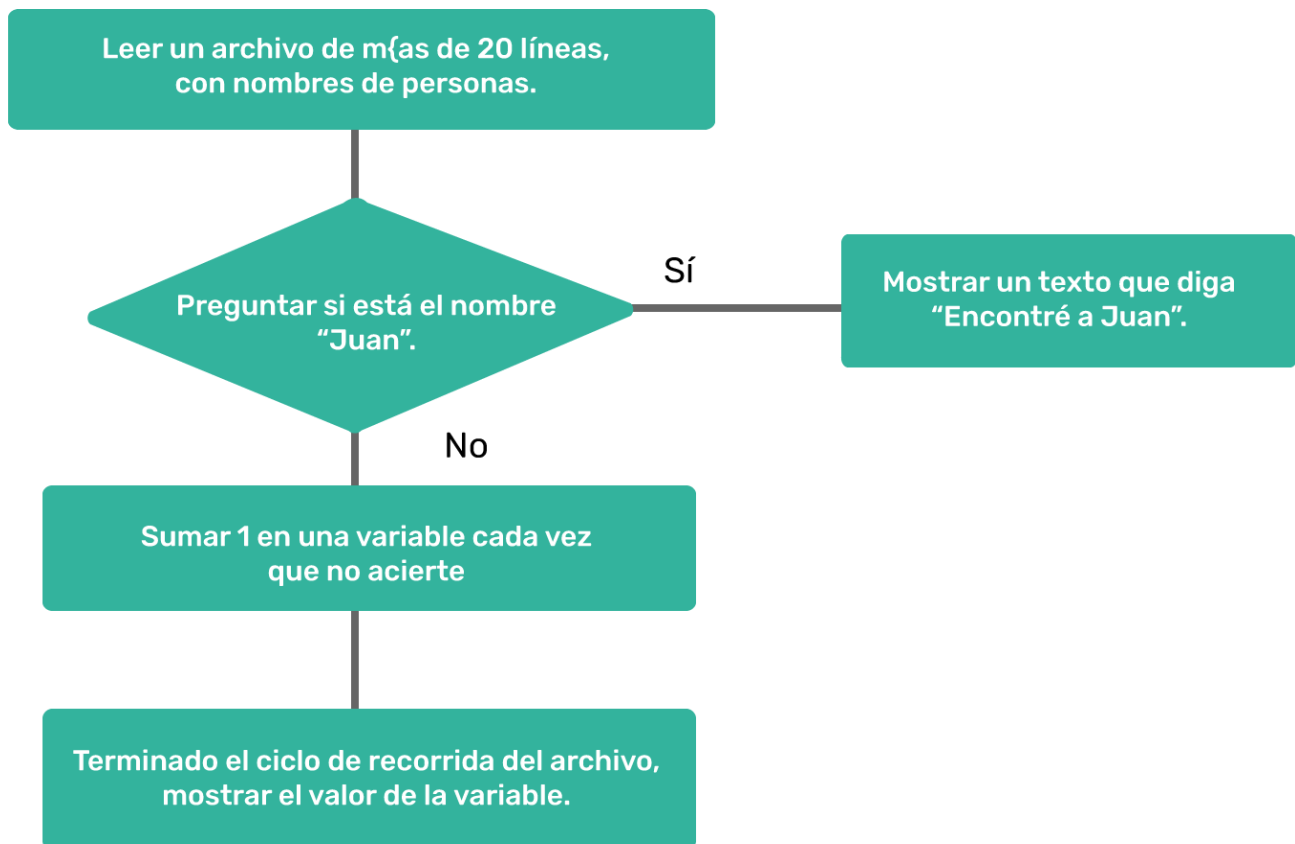
¿Qué recibimos?

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

Instrucciones

Ejercicio 1

De forma individual, créa un script en Bash que realice lo siguiente:



Los pasos que a seguir para poder preparar el ejercicio son:

- Crear, en primer lugar, el archivo. Podés usar nano y allí copiar, o con la instrucción cat vas escribiendo 1 a 1 los nombres. El archivo se llamará: **lista_nombres**.
- Antes de comenzar el script, podés 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, debería recorrer el archivo y mostrarlo línea por línea. Sabiendo esto, ahora vas a realizar el agregado de la lógica de control. Para ello tenés que utilizar una sentencia if-else, en donde se comparará 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, procedé a completar la ejercitación. No te olvides de declarar la variable a en el script final.

Ejercicio 2

De forma individual, armá —en función de nuestro archivo de nombres— 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, 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, 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**. Podés usar como base parte del script que armaste anteriormente, 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 web service, teniendo en cuenta que el parámetro para cada solicitud será dinámico.

Veamos un ejemplo de cómo hacer 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"; }
```

Podés ver 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 primera solicitud es la variable `gen` y en la segunda solicitud es la variable `cn`.

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