Teoría bucles BASH (IF, ELIF, WHILE y FOR)

IF: evento que se ejecuta cuando se cumple la condición expuesta en el bucle

- La condición debe ir dentro de los corchetes [ ]
- Una vez terminada la condición debemos añadir el "cierre de condición" con el símbolo ";"
- Después de la condición indicaremos el evento que se ejecutará a través de "then"
- Una vez finalizado el bucle, lo tendremos que dar finalizado con "fi"
- Formato:

if [condición]; then
EJECUCIÓN DEL PROGRAMA
fi

Ejemplo) Si la variable X es igual a 1, mostramos el mensaje "Es correcto"

```
if [ $x -eq 1 ]; then
echo "Es correcto"
fi
```

ELIF: evento que va unido al primer bucle IF visto. Evento que se ejecuta como una alternativa a la primera condición expuesta.

- Las condiciones y formato son idénticas a las del bucle IF

Ejemplo) Si la variable X es igual a 1, mostramos el mensaje "Es correcto"; Si la variable X es igual a 2, mostramos el mensaje "NO ES CORRECTO"

```
if [ $x -eq 1 ]; then
echo "Es correcto"
elif [ $x -eq 2 ]; then
echo "NO ES CORRECTO"
fi
```

ELSE: evento que va unido al primer bucle IF visto. Evento que se ejecuta cuando no se cumpla la condición o condiciones de los IF o ELIF anteriores.

- El formato es más sencillo al del bucle IF, ya que no es necesario insertar ninguna condición

Ejemplo) Si la variable X es igual a 1, mostramos el mensaje "Es correcto"; Si la variable X es igual a 2, mostramos el mensaje "No es correcto", si no se cumple ninguna de las anteriores condiciones, mostramos el mensaje "NINGUNA".

```
if [ $x -eq 1 ]; then
echo "Es correcto"
elif [ $x -eq 2 ]; then
echo "No es correcto"
else
echo "NINGUNA"
fi
```

WHILE: evento que se ejecuta siempre y cuando la condición expuesta en el bucle WHILE se cumpla. Una vez recorrido el bucle (si la condición se sigue cumpliendo), se volverá a ejecutar una y otra vez.

- La condición debe ir dentro de los corchetes [ ]
- Una vez terminada la condición debemos añadir el "cierre de condición" con el símbolo ";"
- Después de la condición indicaremos el evento que se ejecutará a través de "do"
- Una vez finalizado el bucle, lo tendremos que dar finalizado con "done"

Ejemplo) Muestra todos los números del 1 al 10, es decir, hasta que el número sea 10

```
#Creamos una variable "numero"
numero = 1

#Mostramos el valor del número en el bucle, hasta que sea igual a 10
while [ $numero != 10 ]; do
       echo "El número es" $numero

#En cada "vuelta" del bucle, sumamos uno a la variable "numero"
numero=$(($numero + 1))
done
```

**Este programa no será del todo correcto, porque el número 10 no lo mostrará.*

FOR: la descripción del bucle FOR es similar a la del WHILE, con las diferencias que en el FOR no es necesario crear una variable anteriormente, ni hay que incrementar en contador de dicha variable (ya que estos pasos se insertan en la condición del FOR)

- Insertamos una variable y le aplicamos un valor (primer valor, c=0)
- Insertamos una condición, cuando esta se cumpla, el programa saldrá del bucle (segundo valor, c<=5)
- Insertamos lo que ocurrirá en cada "vuelta" que demos al bucle (tercer valor, c++)
- Después de los datos anteriores indicaremos el evento que se ejecutará a través de "do"
- Una vez finalizado el bucle, lo tendremos que dar finalizado con "done"
- Formato:

```
#!/bin/bash
for (( c=1; c<=5; c++ ))
do
echo $c
done
```

Ejemplo) Muestra todos los números del 1 al 10

```
#!/bin/bash
for (( c=1; c<=10; c++ ))
do
echo $c
done
```