

## 105.2 Personalización y creación de scripts simples

If: Los comando if, then y else nos permiten comparar valores numéricos como se muestra en el siguiente ejemplo:

```
#!/bin/bash

echo "ingrese dos numeros";
read a b;

echo "a=$a";
echo "b=$b";

if [ $a -gt $b ]
then
echo "El numero $a es mayor que $b"
else
    if [ $a -eq $b ]
    then
echo "El numero $a es igual que $b"
    else
echo "El numero $b es mayor que $a"
    fi
fi
echo
echo "El script ha funcionado correctamente"
exit
```

**Case:** La condición del case nos va a ahorrar de todo el compilado de comandos if...then...else y la sintaxis es bastante simple. Ejemplo:

```
# TEST DE CASE
echo ""
case $VAR1 in
  0)
    echo "El valor digitado fui 0"
    ;;
  1|2|3|4|5)
    echo "El valor digitado fui entre 1 y 5"
    sleep 3
    ;;
  *)
    echo "El valor digitado fue mayor que 5"
esac
echo "Fin del Script"
```

**Rutinas de loop del Shell for,while y until:** Antes de comenzar vamos a estudiar dos comandos bien simples disponibles en el bash.

**Seq + número:** este va a realizar una sequencia de números.

**expr (expression):** este hace una cuenta aritmética. Por ejemplo:

```
# TEST DE FOR
echo ""
echo "=====LOOP DO FOR======"
for i in 1 2 3 4 5
do
  j=`expr $i + $VAR1`
  echo "$i + $VAR1 = $j"
done
```

Vamos a ver un segundo método utilizando el comando seq:

**While:** Este trabaja de manera parecida al comando for.

```
# \ se usa para proteger el valor y #  
# no interpretar esto como una variable#
```

```
#TEST DE WHILE  
echo ""  
echo "=====LOOP DO WHILE===== "  
while [ $VAR1 -le $1 ]  
do  
    echo "El Valor actual de \ $VAR es: $VAR1"  
    VAR1=`expr $VAR1 + 1`  
    sleep 1  
done
```

```
root@cesarm:~# bash primer_script.sh 20  
===== Mi primer script =====  
  
Nuestro script se llama primer_script.sh  
Este script recibio 1 parametros que son: 20  
  
Por favor, digite un numero de 0 a 10:12  
El valor digitado fue 12  
  
Este es el valor de VAR3 99  
  
El archivo /etc/passwd posee 41 lineas. La hora actual es 12  
  
El valor digitado fue mayor que 5  
Fin del Script
```

En cuanto el VAR1 fue menor o igual que 20 el va a realizar las actividades del loop.

**Until:** Es muy parecida al comando while.

```
#TEST DE UNTIL  
echo ""  
echo "=====LOOP DO UNTIL===== "  
until [ $VAR1 = $0 ]  
do  
    echo "El Valor actual de \ $VAR es: $VAR1"  
    VAR1=`expr $VAR1 + 1`  
    sleep 1  
done
```

```
===== Mi primer script =====
```

```
Nuestro script se llama primer_script.sh
```

```
Este script recibio 1 parametros que son: 20
```

```
Por favor, digite un numero de 0 a 10:15
```