



CEU

*Fundación San Pablo  
Andalucía*

| 25<sup>o</sup> aniversario

# SISTEMAS INFORMÁTICOS (DISEÑO APLICACIONES MULTIMEDIA Y DISEÑO DE APLICACIONES WEB) TEMA 9: INTRODUCCIÓN A LINUX

Profesor: Rafael Madrigal Toscano

# TEMA 9: INTRODUCCIÓN A LINUX

1. HISTORIA GENERAL
2. INSTALACIÓN Y GESTIÓN DEL SISTEMA (UBUNTU)
3. ARQUITECTURA DE SISTEMA
4. SHELLS Y EDITORES
5. ESTRUCTURA DE DIRECTORIOS
6. FLUJO DE DATOS
7. PERMISOS
8. ENTORNO DE TRABAJO BASH
9. PROCESOS

# TEMA 9: INTRODUCCIÓN A LINUX

**10. DISPOSITIVOS DE ALMACENAMIENTO. MONTAJE Y DESMONTAJE**

**11. MANEJO DE PAQUETES**

**12. ESTRUCTURA BÁSICA DE CONTROL Y BUCLES BASH**

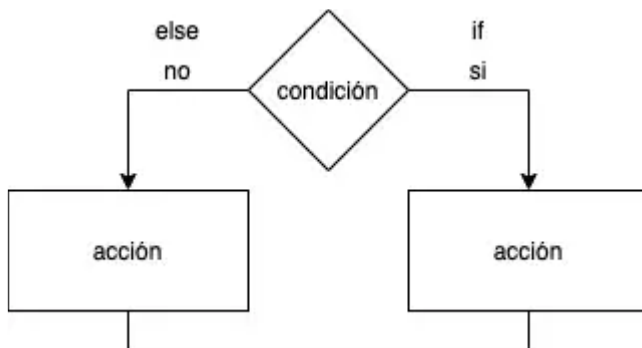
# TEMA 9: INTRODUCCIÓN A LINUX

## 12. ESTRUCTURAS BÁSICAS DE CONTROL Y BUCLES EN BASH

En Bash existen diferentes estructuras para controlar el flujo en un script. Hoy nos centraremos en dos muy concretas: if/else y for

### 1. if/else sintaxis:

```
if [ expresión 1 ]; then
    Realizar si expresión 1 es verdadera
else
    Realizar si expresión 1 es falsa
fi
```



```
1  #!/bin/bash
2  # Comparación de cadenas
3
4  CAD1="Pablo"
5  CAD2="Cesar"
6  CAD3="Alvaro"
7
8  if [ $CADENA1 = $CADENA2 ]; then
9      echo "\$CAD1 es igual a \$CAD2"
10
11  elif [ $CADENA1 != $CADENA2 ]; then
12      echo "\$CAD1 no es igual a \$CAD2"
13
14  fi
15
16  if [ -z $CADENA3 ]; then
17      echo "\$CAD3 está vacía"
18  else
19      echo "\$CAD3 no está vacía"
20  fi
```

# TEMA 9: INTRODUCCIÓN A LINUX

## 12. ESTRUCTURAS BÁSICAS DE CONTROL Y BUCLES EN BASH

### 1. if/else

Es posible anidar expresiones usando tanto and (y, &&) como or (o, ||). También se puede usar el operador not (!) para indicar una negación.

#### Operadores de comparación de cadenas alfanuméricas

|  |  |
|--|--|
| Cadena1 = Cadena2                          | Cadena1 = Cadena2                          |
| Verdadero si Cadena1 es IGUAL a Cadena2    | Verdadero si Cadena1 es IGUAL a Cadena2    |
| Cadena1 != Cadena2                         | Cadena1 != Cadena2                         |
| Verdadero si Cadena1 NO es IGUAL a Cadena2 | Verdadero si Cadena1 NO es IGUAL a Cadena2 |
| Cadena1 < Cadena2                          | Cadena1 < Cadena2                          |
| Verdadero si Cadena1 es MENOR a Cadena2    | Verdadero si Cadena1 es MENOR a Cadena2    |

#### Operadores de comparación de valores numéricos

|                     |  |
|---------------------|--|
| Numero1 -eq Numero2 | Verdadero si Numero1 es IGUAL a Numero2. (equal)                       |
| Numero1 -ne Numero2 | Verdadero si Numero1 NO es IGUAL a Variable2. (not equal)              |
| Numero1 -lt Numero2 | Verdadero si Numero1 es MENOR a Variable2. (less that)                 |
| Numero1 -gt Numero2 | Verdadero si Numero1 es MAYOR que Variable2. (greather that)           |
| Numero1 -le Numero2 | Verdadero si Numero1 es MENOR O IGUAL que Numero2 (les or equal).      |
| Numero1 -ge Numero2 | Verdadero si Numero1 es MAYOR O IGUAL que Numero2. (greather or equal) |

# TEMA 9: INTRODUCCIÓN A LINUX

## 12. ESTRUCTURAS BÁSICAS DE CONTROL Y BUCLES EN BASH

El comando `read` sirve para leer información del teclado y guardarla en variables. Su sintaxis es: `read var1 var2 var3 ...`

| Opción | Descripción  |
|--------|--|
| -a     | Para leer las palabras como elementos de un array  |
| -d     | Indica un delimitador de fin de línea. La línea se lee hasta encontrar este delimitador.                                     |
| -n max | Lee como máximo max caracteres   |
| -p     | Permite indicar un texto de prompt. nos permite aportar un texto de prompt al comando, que se imprime antes de pedir el dato |

`read -p "texto de la pregunta" variable`

```
1  #!/bin/bash
2  # Se nos pide un número y decidimos si es par o impar.
3  clear
4  read -p "Introduzca un número : " NUMERO
5  let RESTO=NUMERO%2
6  if [ $RESTO -eq 0 ]; then
7  echo "El número $NUMERO es par"
8  else
9  echo "El número $NUMERO es impar" fi
```

# TEMA 9: INTRODUCCIÓN A LINUX

## 12. ESTRUCTURAS BÁSICAS DE CONTROL Y BUCLES EN BASH

### 2. Bucle for

Sintaxis básica de la estructura for:

for VARIABLE in conjunto; do

    Estas líneas se repiten una vez por cada elemento del conjunto,

    Y variable va tomando los valores del conjunto uno por uno.

done

```
rmadrigal@valkyria:~$ nano ejemplo_for.sh
#!/bin/bash
for NOMBRE in $( cat nombres.txt ); do
    echo "La variable vale $NOMBRE en este paso."
done
```

```
1  #!/bin/bash
2  for ((NUM=2; NUM <=40; NUM=NUM+2 )); do
3      echo $NUM
4  done
5
```

# TEMA 9: INTRODUCCIÓN A LINUX

## 12. ESTRUCTURAS BÁSICAS DE CONTROL Y BUCLES EN BASH

### Ejercicio

Realice un trabajo que sobre la introducción a las estructuras de control, bucles y funciones en Bash.

Estas construcciones nos ayudan a controlar la ejecución de un script.

De la misma forma escriba un script en bash que contenga un programa que pida al usuario un número de 5 cifras mayor de 1000 nos devuelve precisamente la mitad del valor numérico de la variable \$NUMERO con al menos 3 decimales y compruebe si el número es capicúa llamando a la función palíndromo.



