

Los scripts en Linux se escriben generalmente en bash (Bourne Again Shell) y permiten automatizar tareas mediante la ejecución de comandos en secuencia. Aquí tienes un resumen de los comandos esenciales relacionados con los scripts de Linux:

Construcción de scripts:

1. Creación de un archivo de script:

- Utiliza un editor de texto como nano o vim para crear un archivo.
- Ejemplo:

```
bash
```

```
nano script.sh
```

2. Indicar el intérprete:

- Incluye en la primera línea del script:

```
bash
```

```
#!/bin/bash
```

- #### **3. Esto indica que el script debe ejecutarse con Bash.**

4. Hacer el script ejecutable:

- Usa el comando:

```
bash
```

```
chmod +x script.sh
```

Comandos básicos para scripts:

• Variables:

- Define variables para almacenar valores:

```
bash
```

```
nombre="Juan"
```

```
echo "Hola, $nombre"
```

• Condicionales:

- Usa if para decisiones:

```
bash
```

```
if [ $edad -ge 18 ]; then
    echo "Mayor de edad"
else
    echo "Menor de edad"
fi
```

- **Bucles:**

- Para repetir tareas:

```
bash
for i in {1..5}; do
    echo "Número $i"
done
bash
while [ $contador -lt 10 ]; do
    echo $contador
    contador=$((contador+1))
done
```

- **Funciones:**

- Reutiliza código con funciones:

```
bash
mi_funcion() {
    echo "Hola desde la función"
}
mi_funcion
```

- **Entrada de usuario:**

- Solicita datos al usuario:

```
bash
```

```
read -p "¿Cuál es tu nombre? " nombre
```

```
echo "Hola, $nombre"
```

- **Gestión de errores:**

- Controla el resultado de comandos con \$?:

```
bash
```

```
comando
```

```
if [ $? -eq 0 ]; then
```

```
    echo "Comando exitoso"
```

```
else
```

```
    echo "Error al ejecutar el comando"
```

```
fi
```

Ejecución del script:

- Para ejecutar un script, simplemente usa:

```
bash
```

```
./script.sh
```

Con estos fundamentos, puedes crear y personalizar scripts para automatizar muchas tareas en Linux.