Por: Alice Dantas

Loops, Testes e Controle de Fluxo no Shell Script

As estruturas de controle permitem que seu script tome decisões (condições) e execute blocos diferentes de comandos dependendo de determinadas situações.

- Laços de Repetição (Loops)

Os loops permitem executar blocos de comandos **várias vezes**.

For — Loop sobre listas, sequências e arquivos

Sintaxe:

```
for variavel in lista; do
    comandos
done
```

Exemplos práticos:

Percorrer uma lista fixa:

```
for nome in Mel Lua Ana; do
    echo "Olá, $nome"
done
```

Percorrer uma seq. numérica:

```
for i in {1..5}; do
    echo "Número: $i"
done
```

Percorrer arq. de um diretório:

```
for arquivo in *.txt; do
    echo "Arquivo: $arquivo"
done
```

While — Repete enquanto a condição for verdadeira

Sintaxe:

```
while [ condição ]; do
    comandos
done
```

Exemplo:

```
contador=1
while [ $contador -le 5 ]; do
    echo "Contador: $contador"
    contador=$((contador + 1))
done
```

Until — Repete até a condição ser verdadeira (oposto do while)

Sintaxe:

```
until [ condição ]; do
    comandos
done
```

Exemplo:

```
contador=1
until [ $contador -gt 5 ]; do
    echo "Contador: $contador"
    contador=$((contador + 1))
done
```

Por: Alice Dantas

- Testes Lógicos e Condições

O Bash permite testar condições dentro de if, while, until, etc., usando:

Testes com colchetes simples [ ] ou duplos [[ ]]:

- **[ ]** é o comando **test** do Bash, mais tradicional.
- **[[ ]]** é uma evolução do Bash.

Testes com [ ] :

Tipo	Sintaxe	Significado
Numérico	[$a -gt 10]	$a > 10
String	["$a" = "abc"]	$a é igual a "abc"
String vazia	[-z "$a"]	String vazia
Arquivo existe	[-e arquivo]	Arquivo ou diretório existe
Arquivo comum	[-f arquivo]	Arquivo comum (não diretório)
Diretório existe	[-d pasta]	É um diretório

Testes com [[ ]]:

Aceita **operadores lógicos** internamente:

&& → **E (AND)**
Usado quando ambas as condições precisam ser verdadeiras.
|| → **OU (OR)**
Usado quando pelo menos uma das condições precisa ser verdadeira.
! → **NÃO (NOT)**
Inverte o resultado da condição. Se seria verdadeiro, vira falso, e vice-versa.

Exemplo:

```
if [[ -f "arquivo.txt" && -r "arquivo.txt" ]]; then
    echo "O arquivo existe E tem permissão de leitura."
fi
```

Por: Alice Dantas

- Controle de Fluxo com break e continue

break → sai do loop imediatamente.
continue → pula para a próxima iteração do loop.

Exemplo com break:

```
for i in {1..10}; do
    if [[ $i -eq 5 ]]; then
        echo "Parando no 5"
        break
    fi
    echo "Número: $i"
done
```

Exemplo com continue:

```
for i in {1..5}; do
    if [[ $i -eq 3 ]]; then
        echo "Pulando o 3"
        continue
    fi
    echo "Número: $i"
done
```