



Introdução à Ciência da Computação – Lista 7

Shell script – parte 4

Nome: Pedro Ferreira Prado RA: 2025.1.08.028

- 1) Crie um script chamado escrevenome, faça com que a saída desse script seja seu nome completo. Não utilize o comando chmod. Depois crie um script chamado testecompara, utilize o operador AND e verifique se o usuário logado tem permissão r e x sobre o script escrevenome. Mostre o resultado da saída.

```
GNU nano 7.2 escrevenome.sh
#!/bin/bash

#aa
echo "Pedro Ferreira Prado"
```

```
GNU nano 7.2 testecompara.sh *
#!/bin/bash

if [ -r escrevenome.sh ] && [ -x escrevenome.sh ]; then
    echo "O usuário tem permissão de leitura e execução no script escrevenome."
else
    echo "O usuário NÃO tem permissão de leitura e/ou execução no script escrevenome."
fi
```

```
pedro@DESKTOP-UAUJT7E:~$ nano escrevenome.sh
pedro@DESKTOP-UAUJT7E:~$ nano testecompara.sh
pedro@DESKTOP-UAUJT7E:~$ chmod +x testecompara.sh
pedro@DESKTOP-UAUJT7E:~$ bash testecompara.sh
O usuário NÃO tem permissão de leitura e/ou execução no script escrevenome.
pedro@DESKTOP-UAUJT7E:~$
```

- 2) Crie um script chamado frutascase. Com base no valor da variável fruta mostre uma breve descrição da fruta. Faça com 5 frutas. Exemplo: fruta=uva, echo "A uva é o fruto da videira ou parreira, uma planta da família Vitaceae. É originária da Ásia e uma das frutas mais antigas utilizadas na alimentação humana. Existem mais de 60 mil variedades da fruta. A cor, o sabor e o tamanho variam de acordo com cada espécie. A uva também é classificada quanto ao destino de produção, de mesa ou para vinicultura. Pode ser consumida in natura ou usada na preparação de doce, vinho, passas, mussels, geléias, tortas, gelatinas, sucos."

```
GNU nano 7.2 frutascase.sh *
#!/bin/bash

fruta=$1

case $fruta in
    uva)
        echo "A uva é o fruto da videira, usada na alimentação humana há milênios." ;;
    banana)
        echo "A banana é rica em potássio e uma das frutas mais consumidas no mundo." ;;
    maçã)
        echo "A maçã é uma fruta versátil, rica em fibras e antioxidantes." ;;
    morango)
        echo "O morango é uma fruta vermelha, suculenta e muito usada em sobremesas." ;;
    laranja)
        echo "A laranja é cítrica, rica em vitamina C e ótima para sucos naturais." ;;
    *)
        echo "Fruta não reconhecida." ;;
esac
```

```
pedro@DESKTOP-UAUJT7E:~$ nano frutascase.sh
pedro@DESKTOP-UAUJT7E:~$ bash frutascase.sh
Fruta não reconhecida.
pedro@DESKTOP-UAUJT7E:~$ bash frutascase.sh banana
A banana é rica em potássio e uma das frutas mais consumidas no mundo.
```

- 3) Cite, explique e faça um script simples para cada estrutura de repetição do shell bash. Use sua criatividade para os scripts.

```
GNU nano 7.2 for_loop.sh *
#!/bin/bash

echo "Pedro Ferreira Prado - RA: 2025.1.08.028"

for nome in Ana João Maria Lucas Pedro
do
    echo "Chamando: $nome"
done
```

```
GNU nano 7.2 while_loop.sh *
#!/bin/bash

echo "Contagem para começar os estudos - Pedro Ferreira Prado"
contador=5

while [ $contador -gt 0 ]
do
    echo "$contador..."
    sleep 1
    contador=$((contador-1))
done

echo "Bora estudar!"
```

```
GNU nano 7.2                                until_loop.sh *
#!/bin/bash

foco=0
echo "Monitorando o foco de Pedro..."

until [ $foco -ge 100 ]
do
    echo "Foco atual: $foco%"
    foco=$((foco+25))
    sleep 1
done

echo "Foco 100%! Estudando como nunca."
```

4) Explique o que é IFS e faça um script diferente do que foi visto em aula. Use sua criatividade.

```
GNU nano 7.2                                ifs_script.sh *
#!/bin/bash

nome_completo="Pedro;Ferreira;Prado"
IFS=';'

echo "Dividindo o nome completo de Pedro Ferreira Prado:"
for parte in $nome_completo
do
    echo "Parte: $parte"
done
```

```
pedro@DESKTOP-UAUJT7E:~$ chmod +x ifs_script.sh
pedro@DESKTOP-UAUJT7E:~$ bash ifs_script.sh
Dividindo o nome completo de Pedro Ferreira Prado:
Parte: Pedro
Parte: Ferreira
Parte: Prado
```

5) Crie um script for no estilo C que mostre na tela os números de 50 a 20.

```
GNU nano 7.2 for_c.sh *
#!/bin/bash

for ((i=50; i>=20; i--))
do
    echo $i
done
```

```
pedro@DESKTOP-UAUJT7E:~$ nano for_c.sh
pedro@DESKTOP-UAUJT7E:~$ chmod +x for_c.sh
pedro@DESKTOP-UAUJT7E:~$ bash for_c.sh
50
49
48
47
46
45
44
43
42
41
40
39
38
37
36
35
34
33
32
31
30
29
28
27
26
25
24
23
22
21
20
```

- 6) Desenvolva um script que receba um parâmetro e verifique se o valor está entre 0 e 10. Caso sim mostre o triplo do valor. Caso ele esteja entre 10 e 20 mostre o dobro. Caso não esteja nos anteriores apresente uma mensagem.

```
GNU nano 7.2                                parametro_valor.sh *
#!/bin/bash

valor=$1

if [ $valor -ge 0 ] && [ $valor -le 10 ]; then
    echo "Triplo: $((valor * 3))"
elif [ $valor -gt 10 ] && [ $valor -le 20 ]; then
    echo "Dobro: $((valor * 2))"
else
    echo "Valor fora do intervalo esperado."
fi
```

```
pedro@DESKTOP-UAUJT7E:~$ bash parametro_valor.sh 8
Triplo: 24
pedro@DESKTOP-UAUJT7E:~$
```

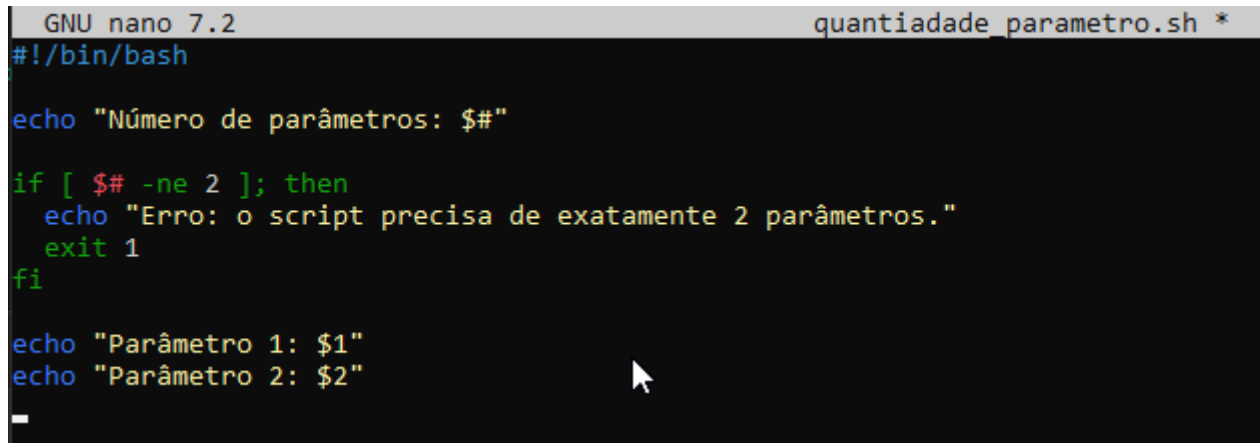
```
pedro@DESKTOP-UAUJT7E:~$ bash parametro_valor.sh 200
Valor fora do intervalo esperado.
```

7) Explique o que é \$# e faça um script diferente do que foi visto em aula. Faça com dois parâmetros. Use sua criatividade.

é uma variável especial do Bash que indica quantos argumentos (parâmetros) foram passados para um script na linha de comando.

```
GNU nano 7.2                                quantidade_parametro.sh *
#!/bin/bash

echo "Número de parâmetros: $#"
```



```
if [ $# -ne 2 ]; then
    echo "Erro: o script precisa de exatamente 2 parâmetros."
    exit 1
fi

echo "Parâmetro 1: $1"
echo "Parâmetro 2: $2"
```

```
pedro@DESKTOP-UAUJT7E:~$ nano quantidade_parametro.sh
pedro@DESKTOP-UAUJT7E:~$ bash quantidade_parametro.sh 10 20
Número de parâmetros: 2
Parâmetro 1: 10
Parâmetro 2: 20
pedro@DESKTOP-UAUJT7E:~$
```