

# PROGRAMAÇÃO EM LINUX SHELL

---

UNIVERSIDADE FEDERAL DO PARÁ

INSTITUTO DE CIÊNCIAS EXATAS E NATURAIS

FACULDADE DE COMPUTAÇÃO

BACHARELADO EM CIÊNCIA DA COMPUTAÇÃO



# VARIÁVEIS

---

- Não precisa de declaração
- Criada no momento da atribuição do valor
  - `num = 10`
  - `pi = 3.14159256`
  - `fileName = "teste.c"`
- Modo de acesso através de `$`
  - `echo $fileName`
  - `echo "Número = $num"`

```
#!/bin/sh
na = 10
nb = 20
nc = $na + $nb
echo "Soma de $na + $nb = $nc"
```

# EXPRESSÃO CONDICIONAL

---

```
if condition
then
    comandos
fi
```

- Ou

```
if condition
then
    comandos
else
    comandos
fi
```

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

```
if [ -e $linux ]
then
    echo 'A variável $linux
existe.'
else
    echo 'A variável $linux não
existe.'
fi
```

# OPERADOS DE CONDIÇÃO

---

Operandos que podem ser usados para testar uma condição nas estruturas condicionais.

-eq	Igual
-ne	Diferente
-gt	Maior
-lt	Menor
-o	Ou
-d	Se for um diretório
-e	Se existir
-z	Se estiver vazio
-f	Se conter texto
-o	Se o usuário for o dono
-r	Se o arquivo pode ser lido
-w	Se o arquivo pode ser alterado
-x	Se o arquivo pode ser executado

# ARGUMENTOS

---

- **\$0** – Retorna o nome do script que foi executado
- **\$N** – Onde N é um número, corresponde ao argumento passado (1 = primeiro argumento, 2 = segundo argumento, 3 = terceiro argumento, etc)
- **\$\*** – Retorna todos os argumentos de uma vez.
- **\$#** – Retorna a quantidade de argumentos passado para o script. (*argc*)

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

```
if [ $# -lt 1 ]; then  
    echo "Faltou utilizar pelo menos um argumento!"  
    exit 1  
fi
```

```
echo "Numero de argumentos: $#"
```

# EXPRESSÃO CONDICIONAL IF-ELSE-IF

---

if condição

comando

elif condição

comando

elif condição

comando

else

comando

fi

```
if [ -z $1 ]; then
    echo "Você tem de entrar com um parâmetro válido"
    exit
elif [ $1 = "parametro1" ]; then
    comando1
    comando2
elif [ $1 = "parametro2" ]; then
    comando3
    comando4
else
    echo "Você tem de entrar com um parâmetro válido"
fi
```



# COMANDO DE REPETIÇÃO FOR

---

```
for [ condição ];  
do  
#Seu codigo  
done
```

```
for i in *  
do  
    cp $i $i.backup  
    mv $i.backup /usr/home/backup  
done
```

Como se  
comporta o \*

```
#!/bin/bash  
echo "Contando ate 5"  
for i in 1 2 3 4 5;  
do  
echo $i;  
done
```

```
echo "Contando ate 10"  
for i in $(seq 10);  
do  
echo $i;  
done
```

```
echo "Contando ate 5 em estilo C"  
for ((i=0; i<=5; i++))  
do  
echo "Executando o $i"  
done
```

`#!/bin/bash`

# COMANDO READ

---

- Permite efetuar a leitura do teclado
  - Ler diretamente um valor
  - Ler uma quantidade de caracteres
  - Ler com tempo determinado
  - Ler sem exibir na tela o valor digitado
  - Ler até um caractere ser digitado
  - Ler um array
  - Ler diversas variáveis

```
echo "Informe seu nome: "; read Nome
echo -e "Seu nome é: $Nome"
```

```
read -n2 Idade; echo
```

```
read -t2 -p "Responda em 2 segundos.Qual a capital do Brasil?"
Resp || echo "Tempo esgotado."
```

```
read -s Senha
```

```
read -sp "Senha: "; echo
```

```
read -p "Faça uma pergunta: " -d'?' PERGUNTA; echo
echo $PERGUNTA
```

```
read -a MinhaArray
echo ${MinhaArray[0]}
echo ${MinhaArray[1]}
```

```
read var1 var2 var3
echo "$var1 $var2 $var3"
```