

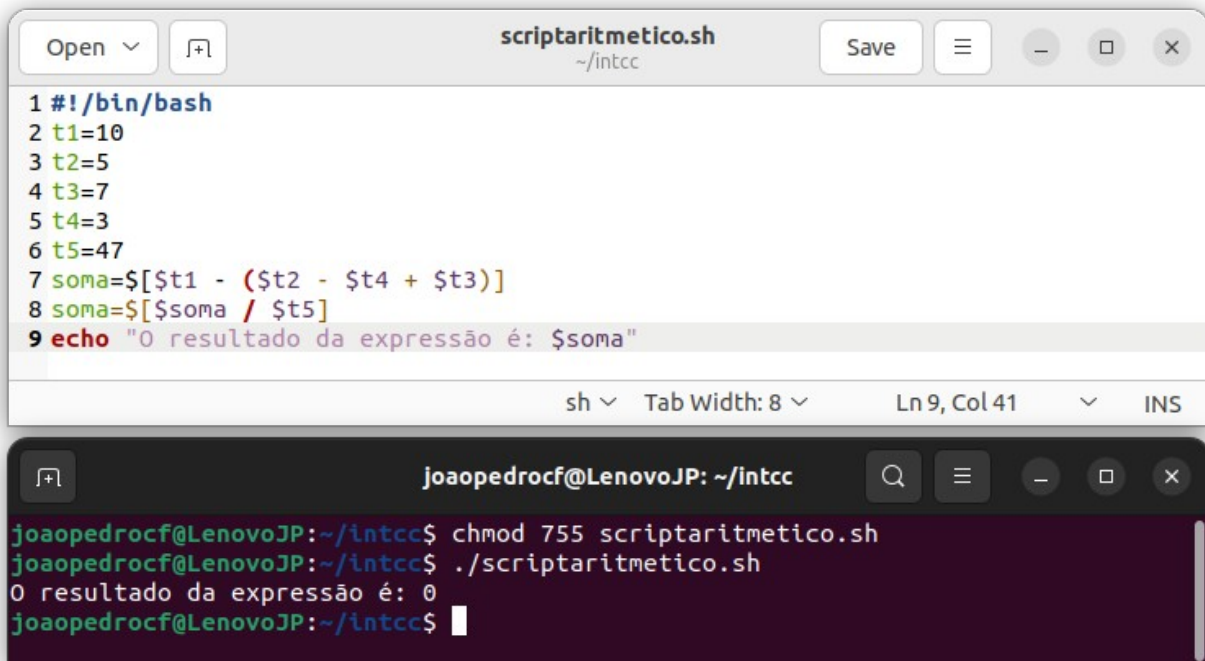
## Introdução à Ciência da Computação - Lista 6

### Shell script - parte 3

**Nome:** João Pedro Carvalho Ferreira

**RA:** 2024.1.08.030

1) Crie um script chamado scriptaritmetico, com uma operação aritmética arbitrária usando pelo menos 4 variáveis, realizando uma operação de divisão cujo resultado não seja um número inteiro. Execute o script e mostre o resultado. Qual o recurso a ser utilizado caso você queira que o valor não inteiro apareça no resultado? Qual variável eu uso para isso?



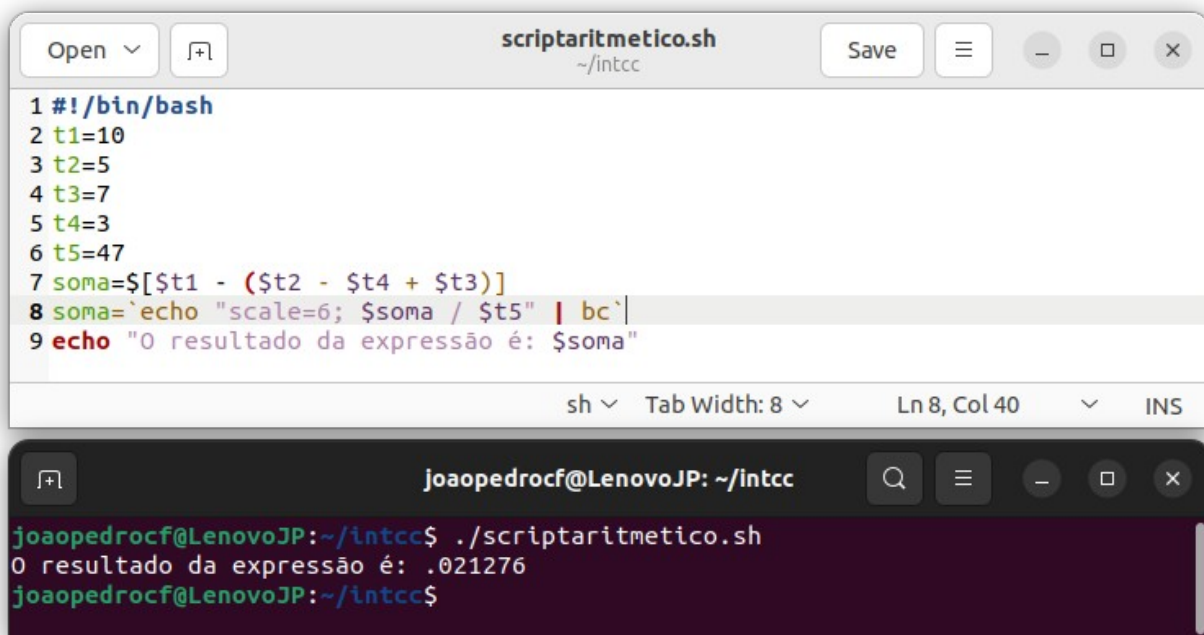
The image shows a code editor window titled 'scriptaritmetico.sh' with the following content:

```
1 #!/bin/bash
2 t1=10
3 t2=5
4 t3=7
5 t4=3
6 t5=47
7 soma=$((t1 - (t2 - t4 + t3)))
8 soma=$((soma / t5))
9 echo "O resultado da expressão é: $soma"
```

Below the code editor is a terminal window showing the execution of the script:

```
joaopedrocf@LenovoJP: ~/intcc
joaopedrocf@LenovoJP:~/intcc$ chmod 755 scriptaritmetico.sh
joaopedrocf@LenovoJP:~/intcc$ ./scriptaritmetico.sh
O resultado da expressão é: 0
joaopedrocf@LenovoJP:~/intcc$
```

Caso queira que o valor decimal apareça no resultado, o recurso a ser utilizado é a variável especial `scale` informando o número de casas decimais desejadas.



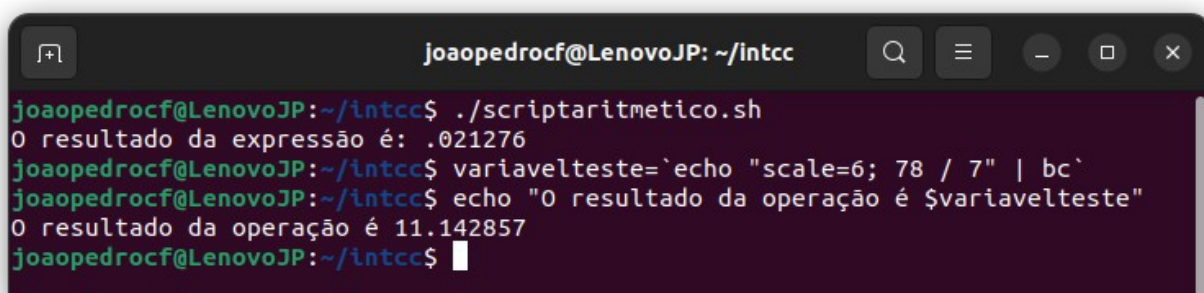
The image shows a code editor window titled "scriptaritmetico.sh" with the following content:

```
1 #!/bin/bash
2 t1=10
3 t2=5
4 t3=7
5 t4=3
6 t5=47
7 soma=$((t1 - (t2 - t4 + t3)))
8 soma=`echo "scale=6; $soma / $t5" | bc`
9 echo "O resultado da expressão é: $soma"
```

Below the editor is a terminal window showing the execution of the script:

```
joaopedrocf@LenovoJP: ~/intcc
joaopedrocf@LenovoJP:~/intcc$ ./scriptaritmetico.sh
O resultado da expressão é: .021276
joaopedrocf@LenovoJP:~/intcc$
```

2) Ponha em execução a calculadora bc. Mostre o uso da variável scale, exibindo um resultado de operação aritmética com 6 casas decimais.



The image shows a terminal window with the following commands and output:

```
joaopedrocf@LenovoJP:~/intcc$ ./scriptaritmetico.sh
O resultado da expressão é: .021276
joaopedrocf@LenovoJP:~/intcc$ variavelteste=`echo "scale=6; 78 / 7" | bc`
joaopedrocf@LenovoJP:~/intcc$ echo "O resultado da operação é $variavelteste"
O resultado da operação é 11.142857
joaopedrocf@LenovoJP:~/intcc$
```

3) Crie um script simples chamado testebc, em que você utilize a calculadora bc dentro dele, envolvendo o uso de algumas variáveis e a operação de divisão, com o direcionamento via pipe. Execute o script, mostrando o resultado.



The image shows a code editor window titled "testebc.sh" with the following content:

```
1 #!/bin/bash
2 v=14
3 h=8
4 l=9
5 r=`echo "scale=5; ($l - $h) / $v" | bc`
6 echo "O resultado é $r"
```

```
joaopedrocf@LenovoJP: ~/intcc
joaopedrocf@LenovoJP:~/intcc$ chmod 755 testebc.sh
joaopedrocf@LenovoJP:~/intcc$ ./testebc.sh
0 resultado é .07142
joaopedrocf@LenovoJP:~/intcc$
```

4) Crie um script chamado testebccomplexo, em que você utilize operações aritméticas diversas com a calculadora bc (pelo menos duas), armazenando os resultados em variáveis, como mostrado na aula. Neste caso, utilize a técnica de redirecionamento de entrada inline. Execute o script, mostrando o resultado.

```
testebccomplexo.sh
1 #!/bin/bash
2 va1=66
3 va2=34
4 va3=50
5 va4=25
6 var=`bc << EXP
7 scale=4
8 a=($va1 + $va2)
9 b=($va4 * $va3)
10 b / a
11 EXP
12 `
13 echo "0 resultado é $var"

joaopedrocf@LenovoJP: ~/intcc
joaopedrocf@LenovoJP:~/intcc$ chmod 755 testebccomplexo.sh
joaopedrocf@LenovoJP:~/intcc$ ./testebccomplexo.sh
0 resultado é 12.5000
joaopedrocf@LenovoJP:~/intcc$
```

5) O que consiste o status de saída de um programa? Mostre um exemplo de execução de dois comandos (um com sucesso e outro desconhecido) e verifique esse status. Mostre em tela.

R: O status de saída de um programa consiste em um valor usado por cada comando que roda no shell para indicar que o processamento terminou.

```
joaopedrocf@LenovoJP: ~/intcc
joaopedrocf@LenovoJP:~/intcc$ ls -a
.          frutas.txt          sequencia.txt          testevariaveis.sh
..         numeros1.txt       testebccomplexo.sh
arquivos   numeros2.txt       testebc.sh
comandos.txt scriptaritmetico.sh testevariaveisambiente.sh
joaopedrocf@LenovoJP:~/intcc$ echo $?
0
joaopedrocf@LenovoJP:~/intcc$ cd ...
bash: cd: ...: No such file or directory
joaopedrocf@LenovoJP:~/intcc$ echo $?
1
joaopedrocf@LenovoJP:~/intcc$
```

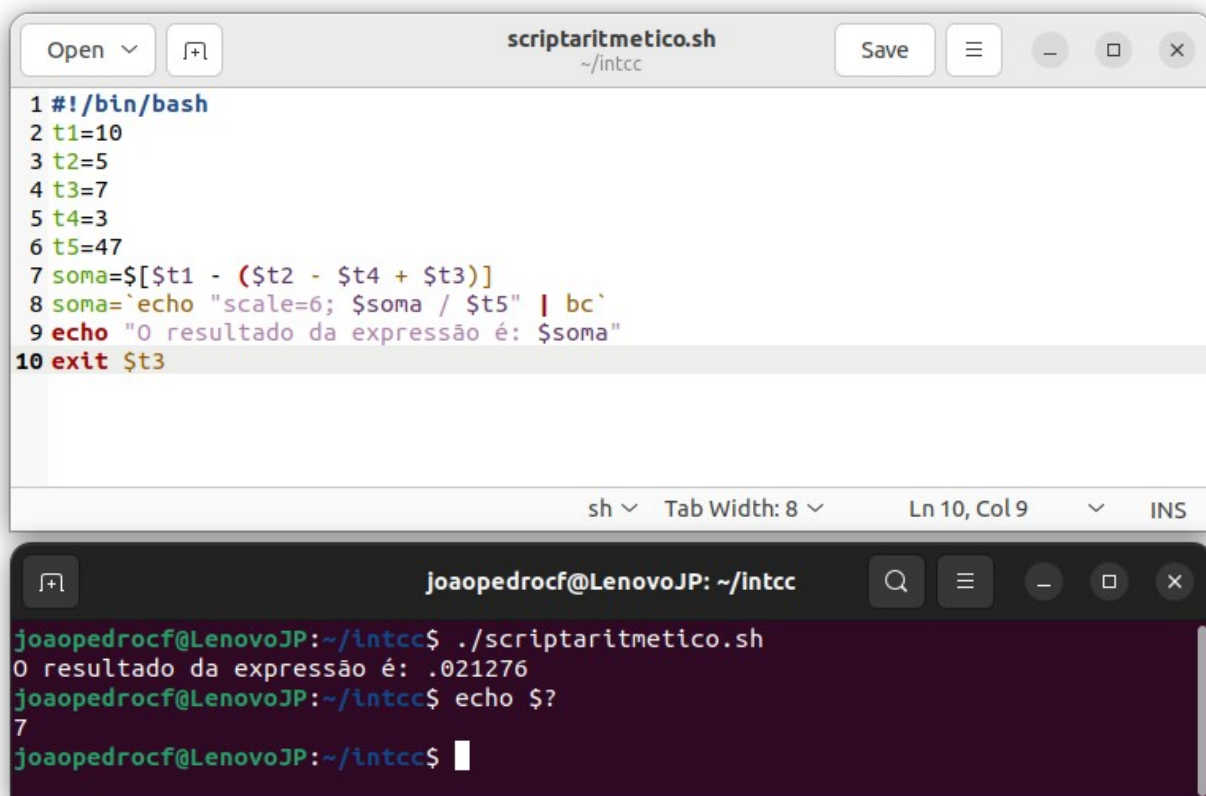
6) Qual a função do comando exit? Mostre um exemplo do uso do comando exit dentro de um script, mudando o valor padrão do status de saída. Mostre tanto o uso do exit exibindo um número qualquer até 255, quanto o valor de uma variável que você utilize no script. Execute o script e mostre o valor do status de saída em cada caso.

R: O comando exit tem a função de alterar o status de saída de um programa.

```
scriptaritmetico.sh
~/intcc
1 #!/bin/bash
2 t1=10
3 t2=5
4 t3=7
5 t4=3
6 t5=47
7 soma=$((t1 - (t2 - t4 + t3))
8 soma=`echo "scale=6; $soma / $t5" | bc`
9 echo "O resultado da expressão é: $soma"
10 exit 10

sh Tab Width: 8 Ln 10, Col 8 INS

joaopedrocf@LenovoJP: ~/intcc
joaopedrocf@LenovoJP:~/intcc$ chmod 755 scriptaritmetico.sh
joaopedrocf@LenovoJP:~/intcc$ ./scriptaritmetico.sh
O resultado da expressão é: .021276
joaopedrocf@LenovoJP:~/intcc$ echo $?
10
joaopedrocf@LenovoJP:~/intcc$
```



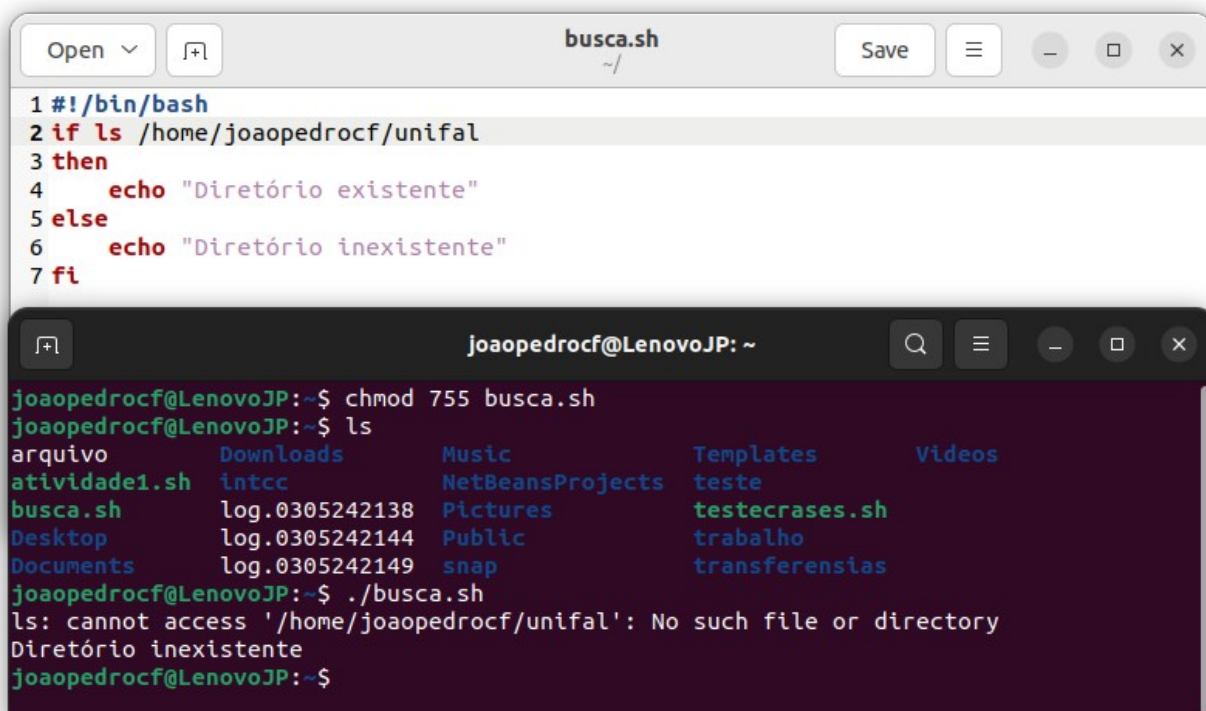
The image shows a code editor window titled 'scriptaritmetico.sh' with the file path '~/.intcc'. The script contains the following code:

```
1 #!/bin/bash
2 t1=10
3 t2=5
4 t3=7
5 t4=3
6 t5=47
7 soma=$((t1 - (t2 - t4 + t3)))
8 soma=`echo "scale=6; $soma / $t5" | bc`
9 echo "O resultado da expressão é: $soma"
10 exit $t3
```

Below the editor is a terminal window with the prompt 'joaopedrocf@LenovoJP: ~/.intcc'. The terminal shows the execution of the script:

```
joaopedrocf@LenovoJP:~/.intcc$ ./scriptaritmetico.sh
0 resultado da expressão é: .021276
joaopedrocf@LenovoJP:~/.intcc$ echo $?
7
joaopedrocf@LenovoJP:~/.intcc$
```

7) Crie um script simples envolvendo comandos condicionais if then else, para verificar a existência de um diretório específico no seu home. Primeiro procure um diretório inexistente, depois um diretório existente e exiba as mensagens específicas de acordo com o resultado. Execute o script e mostre em tela.



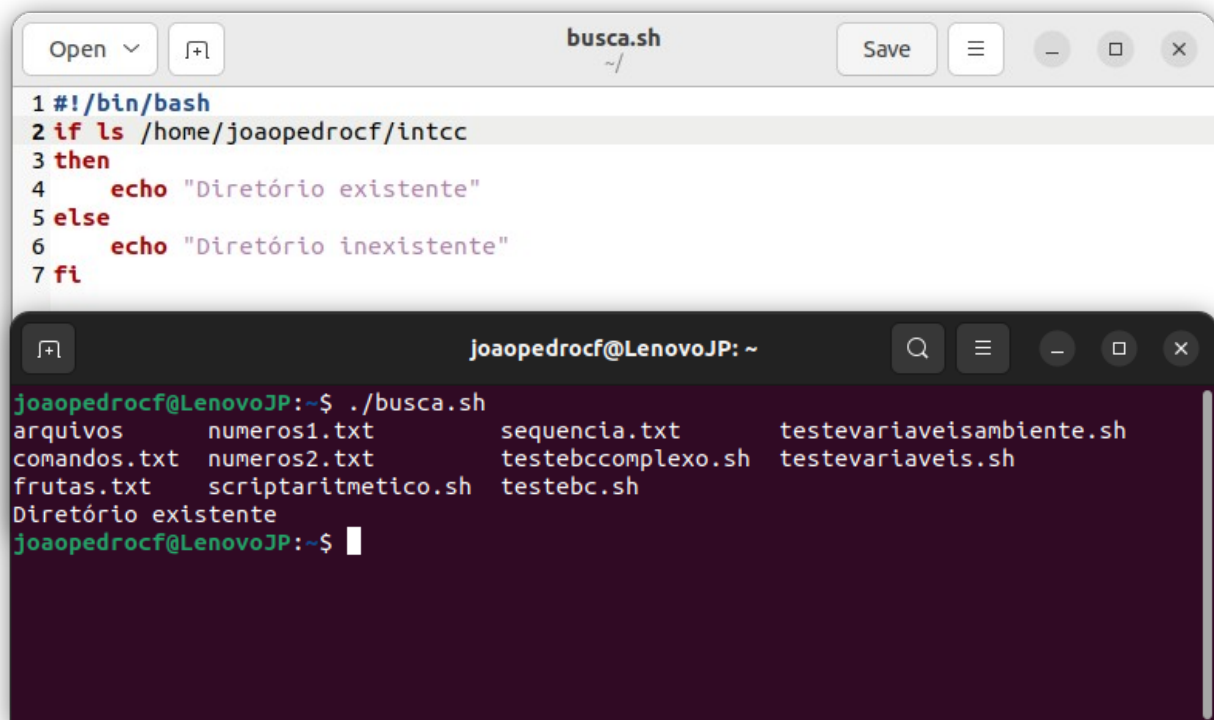
The image shows a code editor window titled 'busca.sh' with the file path '~/'. The script contains the following code:

```
1 #!/bin/bash
2 if ls /home/joaopedrocf/unifal
3 then
4     echo "Diretório existente"
5 else
6     echo "Diretório inexistente"
7 fi
```

Below the editor is a terminal window with the prompt 'joaopedrocf@LenovoJP: ~'. The terminal shows the execution of the script:

```
joaopedrocf@LenovoJP:~$ chmod 755 busca.sh
joaopedrocf@LenovoJP:~$ ls
arquivo      Downloads    Music        Templates    Videos
atividade1.sh intcc        NetBeansProjects teste
busca.sh     log.0305242138 Pictures      testecrases.sh
Desktop      log.0305242144 Public        trabalho
Documents    log.0305242149 snap          transferencias
joaopedrocf@LenovoJP:~$ ./busca.sh
ls: cannot access '/home/joaopedrocf/unifal': No such file or directory
Diretório inexistente
joaopedrocf@LenovoJP:~$
```





The image shows a code editor window titled 'busca.sh' with a file icon and 'Open' button. The code inside is a shell script that checks if a directory exists. Below the editor is a terminal window titled 'joaopedrocf@LenovoJP: ~'. The terminal shows the command './busca.sh' being executed, followed by a list of files in the current directory: arquivos, numeros1.txt, sequencia.txt, testevariaveisambiente.sh, comandos.txt, numeros2.txt, testebccomplexo.sh, testevariaveis.sh, frutas.txt, scriptaritmetico.sh, and testebc.sh. The script then outputs 'Diretório existente'.

```
1#!/bin/bash
2if ls /home/joaopedrocf/intcc
3then
4    echo "Diretório existente"
5else
6    echo "Diretório inexistente"
7fi
```

```
joaopedrocf@LenovoJP:~$ ./busca.sh
arquivos      numeros1.txt  sequencia.txt  testevariaveisambiente.sh
comandos.txt  numeros2.txt  testebccomplexo.sh  testevariaveis.sh
frutas.txt    scriptaritmetico.sh  testebc.sh
Diretório existente
joaopedrocf@LenovoJP:~$
```

8) Crie um script envolvendo várias condicionais usando a estrutura if then elif else, fazendo duas operações aritméticas arbitrárias, verificando o valor das variáveis que armazenam essa operação, checando se o valor da primeira é maior, menor ou igual ao valor da segunda. Execute o script e mostre o resultado em tela.

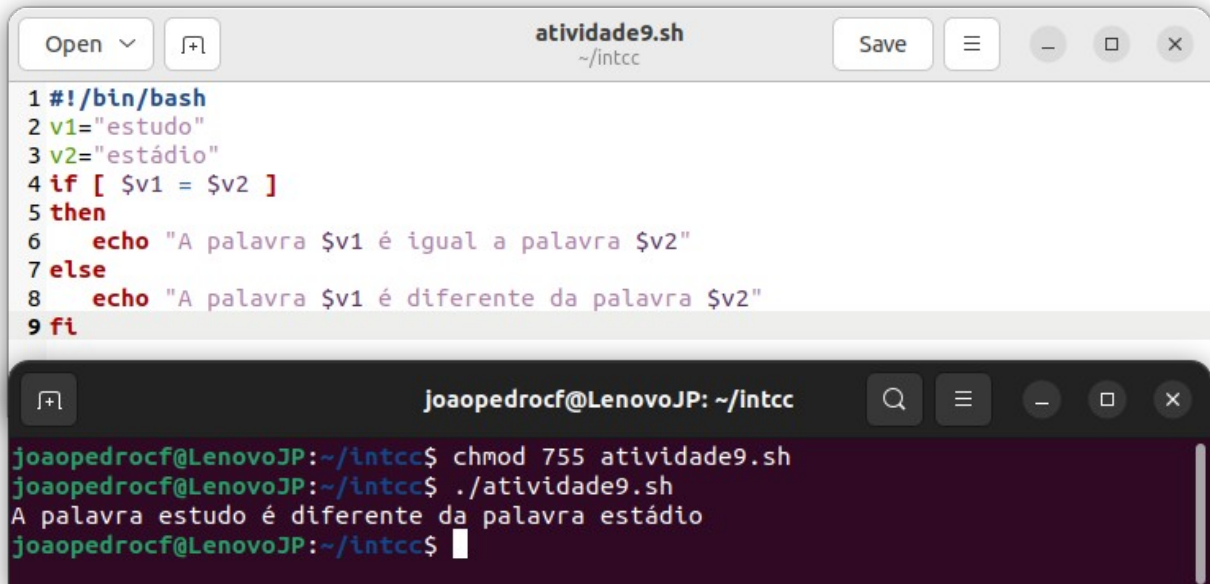


The image shows a code editor window titled 'atividade8.sh' with a file icon and 'Open' button. The code inside is a shell script that performs arithmetic operations and compares the results. Below the editor is a terminal window titled 'joaopedrocf@LenovoJP: ~/intcc'. The terminal shows the command 'chmod 755 atividade8.sh' being executed, followed by './atividade8.sh'. The script outputs '20 é menor que 64'.

```
1#!/bin/bash
2j1=4
3j2=16
4r1=$((j1 + j2))
5r2=$((j2 * j1))
6if [ $r1 -gt $r2 ]
7then
8    echo "$r1 é maior que $r2"
9elif [ $r1 -lt $r2 ]
10then
11    echo "$r1 é menor que $r2"
12else
13    echo "$r1 é igual a $r2"
14fi
```

```
joaopedrocf@LenovoJP:~/intcc$ chmod 755 atividade8.sh
joaopedrocf@LenovoJP:~/intcc$ ./atividade8.sh
20 é menor que 64
joaopedrocf@LenovoJP:~/intcc$
```

9) Crie um script envolvendo condicionais usando a estrutura if then else, criando duas variáveis string arbitrárias e verificando seus valores, checando se o conteúdo das variáveis é igual. Execute o script e mostre o resultado em tela.



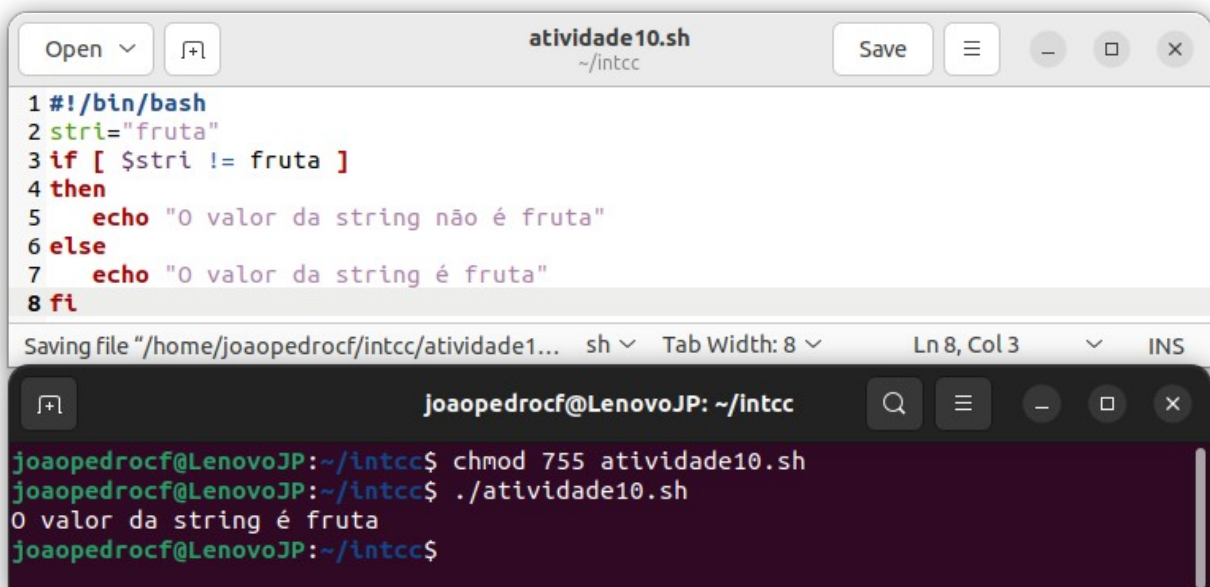
The screenshot shows a code editor window titled "atividade9.sh" with the following content:

```
1 #!/bin/bash
2 v1="estudo"
3 v2="estádio"
4 if [ $v1 = $v2 ]
5 then
6     echo "A palavra $v1 é igual a palavra $v2"
7 else
8     echo "A palavra $v1 é diferente da palavra $v2"
9 fi
```

Below the editor is a terminal window with the following commands and output:

```
joaopedrocf@LenovoJP: ~/intcc
joaopedrocf@LenovoJP:~/intcc$ chmod 755 atividade9.sh
joaopedrocf@LenovoJP:~/intcc$ ./atividade9.sh
A palavra estudo é diferente da palavra estádio
joaopedrocf@LenovoJP:~/intcc$
```

10) Crie um script envolvendo condicionais usando a estrutura if then else, criando uma string com um conteúdo, verificando se seu valor é "fruta". Execute o script e mostre o resultado em tela.



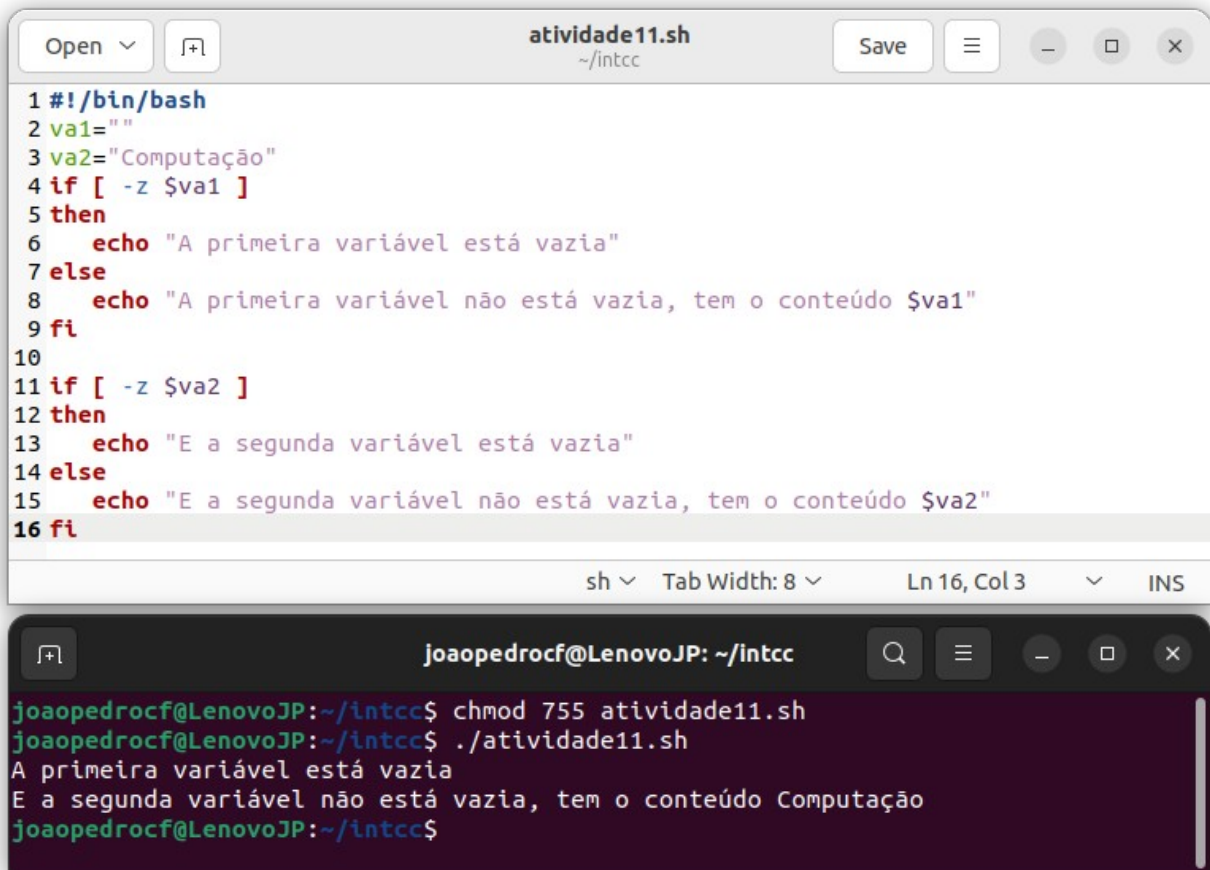
The screenshot shows a code editor window titled "atividade10.sh" with the following content:

```
1 #!/bin/bash
2 stri="fruta"
3 if [ $stri != fruta ]
4 then
5     echo "O valor da string não é fruta"
6 else
7     echo "O valor da string é fruta"
8 fi
```

Below the editor is a terminal window with the following commands and output:

```
joaopedrocf@LenovoJP: ~/intcc
joaopedrocf@LenovoJP:~/intcc$ chmod 755 atividade10.sh
joaopedrocf@LenovoJP:~/intcc$ ./atividade10.sh
O valor da string é fruta
joaopedrocf@LenovoJP:~/intcc$
```

11) Crie um script envolvendo condicionais usando a estrutura if then else, criando duas strings, uma vazia, outra com conteúdo e verificando estes resultados (se tem conteúdo em ambos os casos).



The image shows a code editor window titled 'atividade11.sh' with the following script:

```
1 #!/bin/bash
2 va1=""
3 va2="Computação"
4 if [ -z $va1 ]
5 then
6     echo "A primeira variável está vazia"
7 else
8     echo "A primeira variável não está vazia, tem o conteúdo $va1"
9 fi
10
11 if [ -z $va2 ]
12 then
13     echo "E a segunda variável está vazia"
14 else
15     echo "E a segunda variável não está vazia, tem o conteúdo $va2"
16 fi
```

Below the editor is a terminal window showing the execution of the script:

```
joaopedrocf@LenovoJP: ~/intcc
joaopedrocf@LenovoJP:~/intcc$ chmod 755 atividade11.sh
joaopedrocf@LenovoJP:~/intcc$ ./atividade11.sh
A primeira variável está vazia
E a segunda variável não está vazia, tem o conteúdo Computação
joaopedrocf@LenovoJP:~/intcc$
```

12) Cite 5 opções de comparações envolvendo arquivos. Escolha uma das opções e crie um script envolvendo essa opção.

- R: Opção -d: verifica se o arquivo existe e se é um diretório.  
Opção -x: verifica a existência do arquivo e se tem permissão de execução.  
Opção -O: verifica se o arquivo existe e se é propriedade do usuário atual.  
Opção -s: verifica a existência do arquivo e se não está vazio.  
Opção -f: verifica se o arquivo existe e se é um arquivo.



```
joaopedrocf@LenovoJP: ~/intcc
joaopedrocf@LenovoJP:~$ ls
arquivo          Downloads        NetBeansProjects testecrases.sh
atividade12.sh   intcc           Pictures         trabalho
atividade1.sh    log.0305242138 Public           transferencias
busca.sh         log.0305242144 snap            Videos
Desktop          log.0305242149 Templates
Documents        Music           teste
joaopedrocf@LenovoJP:~$ cd intcc
joaopedrocf@LenovoJP:~/intcc$ ls
arquivos          atividade9.sh   numeros2.txt    testebc.sh
atividade10.sh    comandos.txt   scriptaritmetico.sh testevariaveisambiente.sh
atividade11.sh    frutas.txt     sequencia.txt   testevariaveis.sh
atividade8.sh     numeros1.txt   testebccomplexo.sh
joaopedrocf@LenovoJP:~/intcc$
```

```
Open  atividade12.sh
1 #!/bin/bash
2 if [ -d $intcc ]
3 then
4     echo "O arquivo existe e também é um diretório. Seu conteúdo é:"
5     cd /home/joaopedrocf/intcc
6     ls
7 else
8     echo "O arquivo não existe ou não um diretório"
9 fi
sh  Tab Width: 8  Ln 9, Col 3  INS
```

```
joaopedrocf@LenovoJP: ~
joaopedrocf@LenovoJP:~$ chmod 755 atividade12.sh
joaopedrocf@LenovoJP:~$ ./atividade12.sh
O arquivo existe e também é um diretório. Seu conteúdo é:
arquivos          atividade9.sh   numeros2.txt    testebc.sh
atividade10.sh    comandos.txt   scriptaritmetico.sh testevariaveisambiente.sh
atividade11.sh    frutas.txt     sequencia.txt   testevariaveis.sh
atividade8.sh     numeros1.txt   testebccomplexo.sh
joaopedrocf@LenovoJP:~$
```