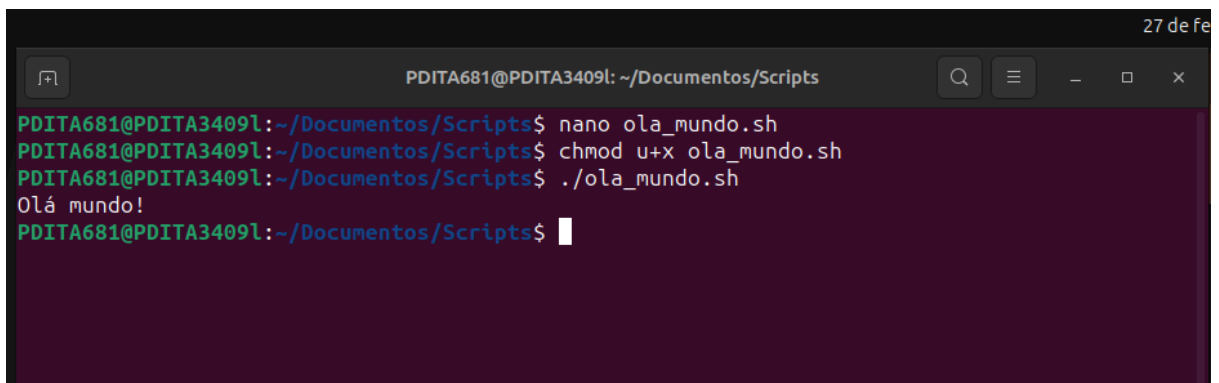


Exercício de Shell Script:

1. Crie um script que imprima uma mensagem de “Olá mundo!”.

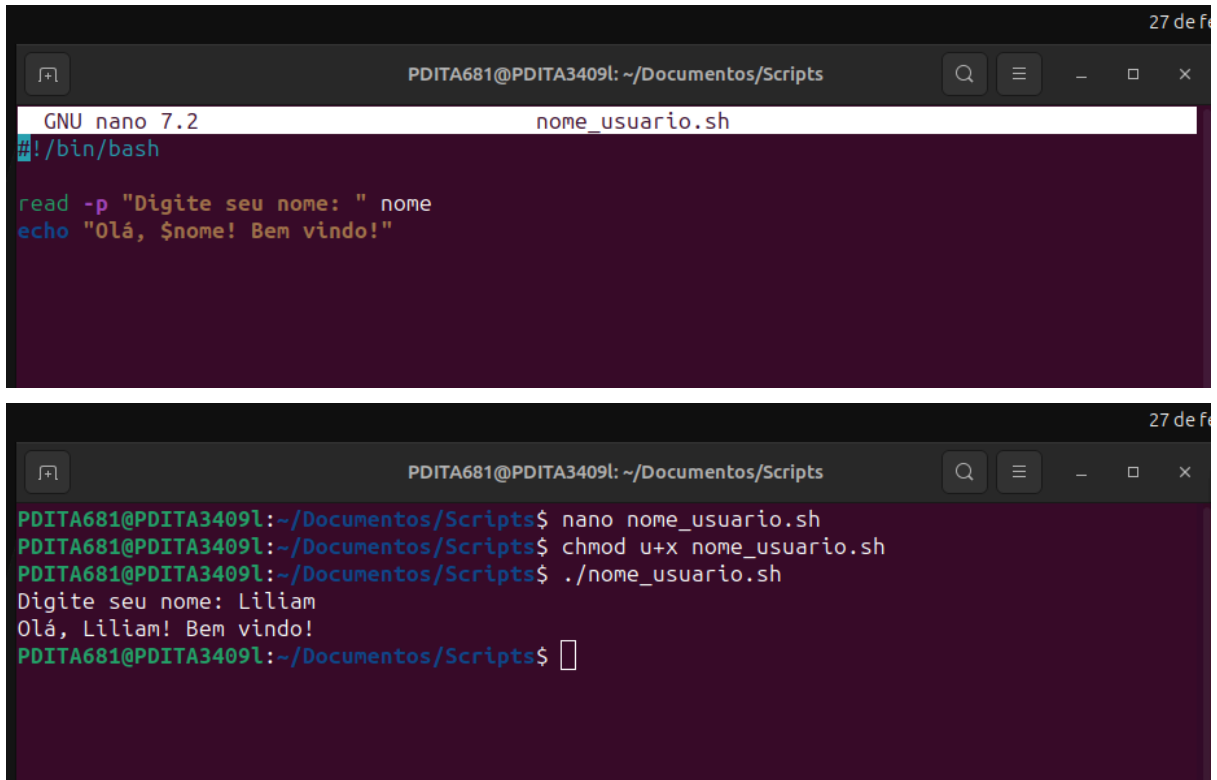


```
GNU nano 7.2 ola_mundo.sh
#!/bin/bash
echo "Olá mundo!"
```



```
PDITA681@PDITA3409l: ~/Documentos/Scripts
PDITA681@PDITA3409l:~/Documentos/Scripts$ nano ola_mundo.sh
PDITA681@PDITA3409l:~/Documentos/Scripts$ chmod u+x ola_mundo.sh
PDITA681@PDITA3409l:~/Documentos/Scripts$ ./ola_mundo.sh
Olá mundo!
PDITA681@PDITA3409l:~/Documentos/Scripts$
```

2. Crie um script que peça ao usuário para inserir seu nome e, em seguida, imprima uma mensagem de saudação usando essa entrada.



The image consists of two screenshots of a terminal window. The top screenshot shows the nano text editor editing a file named 'nome_usuario.sh'. The content of the file is a shell script that prompts the user for their name and prints a greeting. The bottom screenshot shows the terminal commands to create the script, set permissions, and run it, with the output of the script execution.

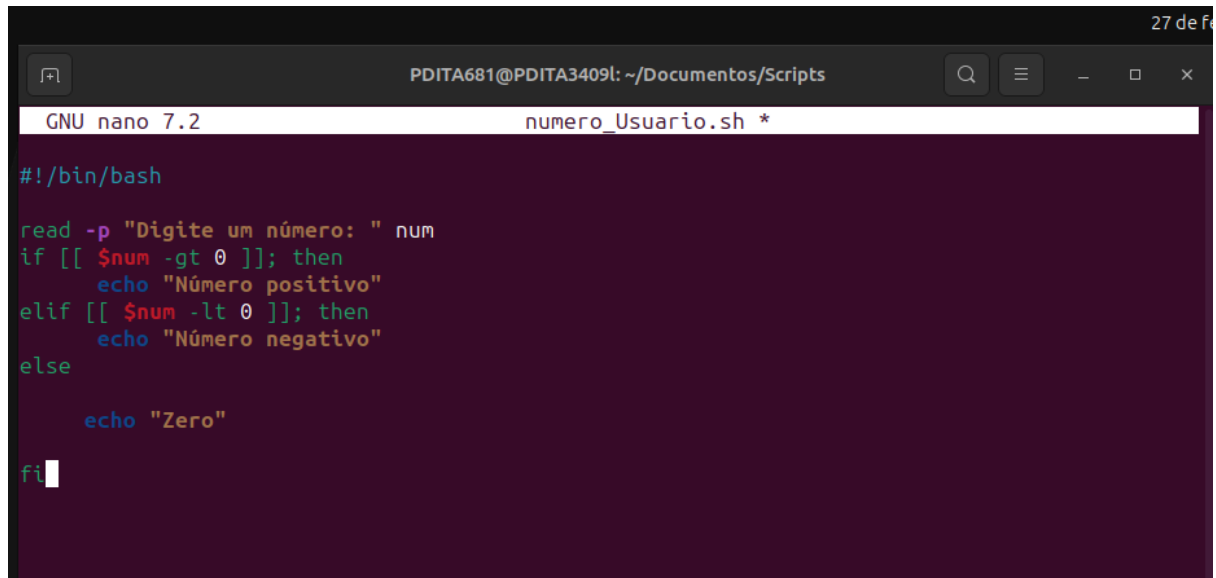
```
GNU nano 7.2 nome_usuario.sh
#!/bin/bash

read -p "Digite seu nome: " nome
echo "Olá, $nome! Bem vindo!"

PDITA681@PDITA3409l: ~/Documentos/Scripts
```

```
PDITA681@PDITA3409l:~/Documentos/Scripts$ nano nome_usuario.sh
PDITA681@PDITA3409l:~/Documentos/Scripts$ chmod u+x nome_usuario.sh
PDITA681@PDITA3409l:~/Documentos/Scripts$ ./nome_usuario.sh
Digite seu nome: Liliam
Olá, Liliam! Bem vindo!
PDITA681@PDITA3409l:~/Documentos/Scripts$
```

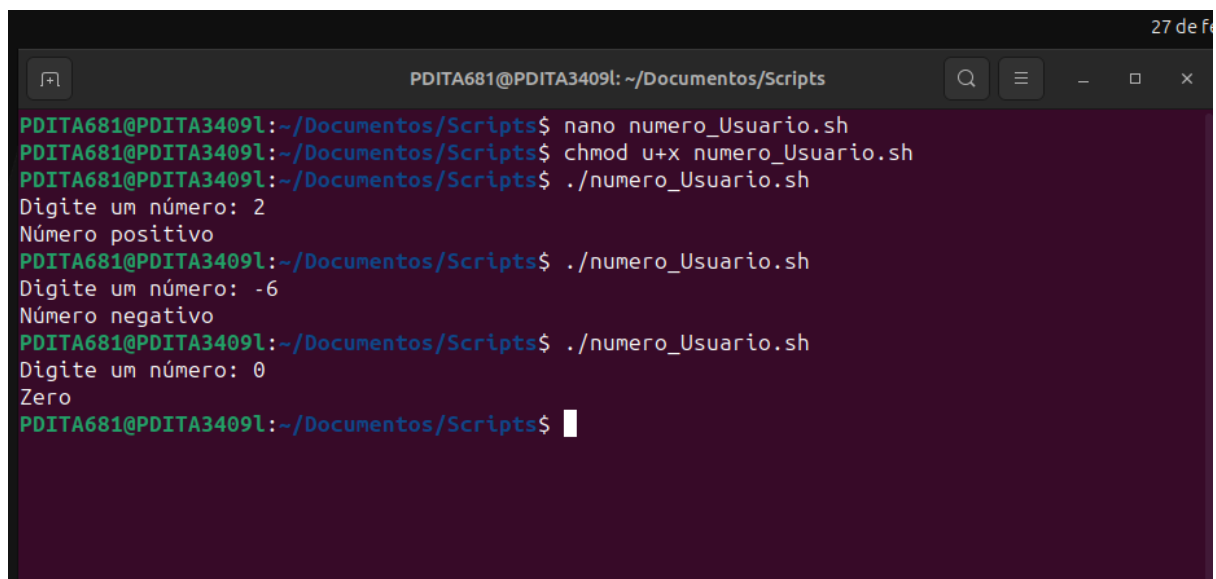
3. Escreva um script que solicite um número do usuário. Se o número for positivo, exiba "Número positivo". Se for negativo, exiba "Número negativo". Se for zero, exiba "Zero".



```
GNU nano 7.2 numero_usuario.sh *

#!/bin/bash

read -p "Digite um número: " num
if [[ $num -gt 0 ]]; then
    echo "Número positivo"
elif [[ $num -lt 0 ]]; then
    echo "Número negativo"
else
    echo "Zero"
fi
```



```
PDITA681@PDITA3409l: ~/Documentos/Scripts
PDITA681@PDITA3409l:~/Documentos/Scripts$ nano numero_usuario.sh
PDITA681@PDITA3409l:~/Documentos/Scripts$ chmod u+x numero_usuario.sh
PDITA681@PDITA3409l:~/Documentos/Scripts$ ./numero_usuario.sh
Digite um número: 2
Número positivo
PDITA681@PDITA3409l:~/Documentos/Scripts$ ./numero_usuario.sh
Digite um número: -6
Número negativo
PDITA681@PDITA3409l:~/Documentos/Scripts$ ./numero_usuario.sh
Digite um número: 0
Zero
PDITA681@PDITA3409l:~/Documentos/Scripts$
```

4. Escreva um script que solicite um número do usuário e crie essa quantidade de arquivos logs com a extensão “.txt”.

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

read -p "Quantos arquivos deseja criar?" quantidade

for ((i=1; i<=quantidade; i++)); do
    touch "log_$i.txt"
done
echo "$quantidade arquivos criados!"
```

```
PDITA681@PDITA3409l: ~/Documentos/Scripts
PDITA681@PDITA3409l:~/Documentos/Scripts$ nano arq_log.sh
PDITA681@PDITA3409l:~/Documentos/Scripts$ chmod u+x arq_log.sh
PDITA681@PDITA3409l:~/Documentos/Scripts$ ./arq_log.sh
Quantos arquivos deseja criar?5
5 arquivos criados!
PDITA681@PDITA3409l:~/Documentos/Scripts$ ls
arq_log.sh      find.sh      log_1.txt    log_5.txt    outros.txt
cafe.sh         find.sh.save log_2.txt    nome_usuario.sh
contador_arquivo.sh grupo.txt    log_3.txt    numero_usuario.sh
dono.txt        i_am_root.sh log_4.txt    ola_mundo.sh
PDITA681@PDITA3409l:~/Documentos/Scripts$
```

5. Escreva um script para renomear todos os arquivos do exercício anterior de “.txt” para “.log”. DICA: Use o comando sed para realizar a substituição das strings

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

for file in *.txt; do

mv "$file" "$(echo "$file" | sed 's/.txt/.log/')"

done

echo "Arquivos renomeados para .log!"
```

```
PDITA681@PDITA3409l: ~/Documentos/Scripts
PDITA681@PDITA3409l:~/Documentos/Scripts$ nano renon_Arquivo.sh
PDITA681@PDITA3409l:~/Documentos/Scripts$ chmod u+x renon_Arquivo.sh
PDITA681@PDITA3409l:~/Documentos/Scripts$ ./renon_Arquivo.sh
Arquivos renomeados para .log!
PDITA681@PDITA3409l:~/Documentos/Scripts$ ls
arq_log.sh      find.sh      log_1.log    log_5.log    outros.log
cafe.sh         find.sh.save log_2.log    nome_usuario.sh renon_Arquivo.sh
contador_arquivo.sh grupo.log    log_3.log    numero_usuario.sh
dono.log        i_am_root.sh log_4.log    ola_mundo.sh
PDITA681@PDITA3409l:~/Documentos/Scripts$
```

6. Crie um script que solicite ao usuário um número e imprima a tabela de multiplicação desse número até 10.

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

read -p "Digite um número para ver sua tabuada até 10! " num

for ((i=1; i<=10; i++)); do
    echo "$num x $i = $((num * i))"
done
```

```
PDITA681@PDITA3409l: ~/Documentos/Scripts
PDITA681@PDITA3409l:~/Documentos/Scripts$ nano mult_tabela.sh
PDITA681@PDITA3409l:~/Documentos/Scripts$ chmod u+x mult_tabela.sh
PDITA681@PDITA3409l:~/Documentos/Scripts$ ./mult_tabela.sh
Digite um número para ver sua tabuada até 10! 5
5 x 1 = 5
5 x 2 = 10
5 x 3 = 15
5 x 4 = 20
5 x 5 = 25
5 x 6 = 30
5 x 7 = 35
5 x 8 = 40
5 x 9 = 45
5 x 10 = 50
PDITA681@PDITA3409l:~/Documentos/Scripts$ ./mult_tabela.sh
Digite um número para ver sua tabuada até 10! 2
2 x 1 = 2
2 x 2 = 4
2 x 3 = 6
2 x 4 = 8
2 x 5 = 10
2 x 6 = 12
2 x 7 = 14
2 x 8 = 16
2 x 9 = 18
2 x 10 = 20
PDITA681@PDITA3409l:~/Documentos/Scripts$
```

7. Escreva um script que solicite ao usuário o nome de um arquivo ou diretório. Após receber a entrada, o script deve determinar se é um arquivo, um diretório ou outro tipo. Além disso, o script deve listar as permissões associadas ao arquivo ou diretório fornecido.

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

read -p "Digite o nome do arquivo ou diretório: " nome

if [[ -f $nome ]]; then
    echo "$nome é um arquivo."
elif [[ -d $nome ]]; then
    echo "$nome é um diretório."
fi

ls -l "$nome"
```

```
PDITA681@PDITA3409l: ~/Documentos/Scripts
PDITA681@PDITA3409l:~/Documentos/Scripts$ nano determ_arquivo.sh
PDITA681@PDITA3409l:~/Documentos/Scripts$ chmod u+x determ_arquivo.sh
Comando 'chmod' não encontrado, você quis dizer:
  comando 'chmod' do deb coreutils (9.1-1ubuntu2)
Experimente: sudo apt install <deb name>
PDITA681@PDITA3409l:~/Documentos/Scripts$ chmod u+x determ_arquivo.sh
PDITA681@PDITA3409l:~/Documentos/Scripts$ ./deter
bash: ./deter: Arquivo ou diretório inexistente
PDITA681@PDITA3409l:~/Documentos/Scripts$ ./determ_arquivo.sh
Digite o nome do arquivo ou diretório: f
ls: não foi possível acessar 'f': Arquivo ou diretório inexistente
PDITA681@PDITA3409l:~/Documentos/Scripts$ ls
arq_log.sh      dono.log        i_am_root.sh    log_4.log        numero_usuario.sh
cafe.sh         find.sh         log_1.log        log_5.log        ola_mundo.sh
contador_arquivo.sh find.sh.save    log_2.log        mult_tabela.sh   outros.log
determ_arquivo.sh grupo.log       log_3.log        nome_usuario.sh  renon_Arquivo.sh
PDITA681@PDITA3409l:~/Documentos/Scripts$ ./determ_arquivo.sh
Digite o nome do arquivo ou diretório: dono.log
dono.log é um arquivo.
-rwxrw-r-- 1 PDITA681 PDITA681 0 fev 23 19:21 dono.log
```

PDITA681@PDITA3409L: ~/Documentos/Scripts

```
PDITA681@PDITA3409L:~/Documentos$ cd Scripts/
PDITA681@PDITA3409L:~/Documentos/Scripts$ nano determ_arquivo.sh
PDITA681@PDITA3409L:~/Documentos/Scripts$ ./determ_arquivo.sh
Digite o nome do arquivo ou diretório: etc
ls: não foi possível acessar 'etc': Arquivo ou diretório inexistente
PDITA681@PDITA3409L:~/Documentos/Scripts$ ./determ_arquivo.sh
Digite o nome do arquivo ou diretório: /etc
/etc é um diretório.
total 1132
-rw-r--r-- 1 root root 3444 jul 5 2023 adduser.conf
drwxr-xr-x 3 root root 4096 out 16 2023 alsa
drwxr-xr-x 2 root root 4096 fev 20 16:53 alternatives
-rw-r--r-- 1 root root 335 jul 26 2023 anacrontab
-rw-r--r-- 1 root root 433 mar 23 2022 apg.conf
drwxr-xr-x 5 root root 4096 out 16 2023 apm
drwxr-xr-x 3 root root 4096 out 16 2023 apparmor
drwxr-xr-x 9 root root 4096 mai 13 2024 apparmor.d
drwxr-xr-x 3 root root 4096 out 16 2023 appport
-rw-r--r-- 1 root root 833 abr 26 2023 appstream.conf
drwxr-xr-x 8 root root 4096 mar 6 2024 apt
```


8. Escreva um script que receba como argumento o nome de um arquivo ou diretório. O script deve determinar se é um arquivo, um diretório ou outro tipo. Além disso, o script deve listar as permissões associadas ao arquivo ou diretório fornecido.

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

# Verifica se um argumento foi passado
if [ $# -eq 0 ]; then
    echo "Uso: ./verificar_argumento.sh <arquivo_ou_diretorio>"
    exit 1
fi

# Obtém o argumento passado
ITEM="$1"

# Verifica se o item existe
if [ -e "$ITEM" ]; then
    if [ -f "$ITEM" ]; then
        echo "'$ITEM' é um arquivo."
    elif [ -d "$ITEM" ]; then
        echo "'$ITEM' é um diretório."
    else
        echo "'$ITEM' é outro tipo de item (link, dispositivo, etc.)."
    fi
    # Lista as permissões do arquivo ou diretório
    ls -l "$ITEM"
else
    echo "Erro: '$ITEM' não existe."
fi
```

```
PDITA681@PDITA3409L: ~/Documentos/Scripts
PDITA681@PDITA3409L:~/Documentos/Scripts$ nano verificar_argumento.sh
PDITA681@PDITA3409L:~/Documentos/Scripts$ chmod u+x verificar_argumento.sh
PDITA681@PDITA3409L:~/Documentos/Scripts$ ./verificar_argumento.sh
Uso: ./verificar_argumento.sh <arquivo_ou_diretorio>
PDITA681@PDITA3409L:~/Documentos/Scripts$ f
f: comando não encontrado
PDITA681@PDITA3409L:~/Documentos/Scripts$ ./verificar_argumento.sh verificar_argumento.sh
'verificar_argumento.sh' é um arquivo.
-rwxrw-r-- 1 PDITA681 PDITA681 587 fev 27 18:00 verificar_argumento.sh
PDITA681@PDITA3409L:~/Documentos/Scripts$
```

```
PDITA681@PDITA3409L: ~/Documentos/Scripts
PDITA681@PDITA3409L:~/Documentos/Scripts$ nano verificar_argumento.sh
PDITA681@PDITA3409L:~/Documentos/Scripts$ chmod u+x verificar_argumento.sh
PDITA681@PDITA3409L:~/Documentos/Scripts$ ./verificar_argumento.sh
Uso: ./verificar_argumento.sh <arquivo_ou_diretorio>
PDITA681@PDITA3409L:~/Documentos/Scripts$ f
f: comando não encontrado
PDITA681@PDITA3409L:~/Documentos/Scripts$ ./verificar_argumento.sh verificar_argumento.sh
'verificar_argumento.sh' é um arquivo.
-rwxrwxr-- 1 PDITA681 PDITA681 587 fev 27 18:00 verificar_argumento.sh
PDITA681@PDITA3409L:~/Documentos/Scripts$ nano verificar_argumento.sh
PDITA681@PDITA3409L:~/Documentos/Scripts$ ./verificar_argumento.sh Documentos
Erro: 'Documentos' não existe.
PDITA681@PDITA3409L:~/Documentos/Scripts$ ls
arq_log.sh      find.sh.save    log_4.log       outros.log
cafe.sh         grupo.log       log_5.log       renon_Arquivo.sh
contador_arquivo.sh i_am_root.sh  mult_tabela.sh  verificar_argumento.sh
determ_arquivo.sh log_1.log       nome_usuario.sh Verificar_argumento.sh
dono.log        log_2.log       numero_usuario.sh
find.sh         log_3.log       ola_mundo.sh
PDITA681@PDITA3409L:~/Documentos/Scripts$ ./verificar_argumento.sh /Scripts
Erro: '/Scripts' não existe.
PDITA681@PDITA3409L:~/Documentos/Scripts$ ./verificar_argumento.sh /etc
'/etc' é um diretório.
total 1132
-rw-r--r-- 1 root root 3444 jul 5 2023 adduser.conf
drwxr-xr-x 3 root root 4096 out 16 2023 alsa
drwxr-xr-x 2 root root 4096 fev 20 16:53 alternatives
-rw-r--r-- 1 root root 335 jul 26 2023 anacrontab
-rw-r--r-- 1 root root 433 mar 23 2022 apg.conf
drwxr-xr-x 5 root root 4096 out 16 2023 apm
drwxr-xr-x 3 root root 4096 out 16 2023 apparmor
```

9. Crie um jogo simples de adivinhação. O script deve gerar um número aleatório e pedir ao usuário para adivinhar. Dê dicas se o número é maior ou menor. Continue até que o usuário adivinhe corretamente. DICA: Use a variável RANDOM para gerar números inteiros aleatórios.

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

# Verifica se um argumento foi passado
if [ $# -eq 0 ]; then
    echo "Uso: ./verificar_argumento.sh <arquivo_ou_diretorio>"
    exit 1
fi

# Obtém o argumento passado
ITEM="$1"

# Verifica se o item existe
if [ -e "$ITEM" ]; then
    if [ -f "$ITEM" ]; then
        echo "'$ITEM' é um arquivo."
    elif [ -d "$ITEM" ]; then
        echo "'$ITEM' é um diretório."
    else
        echo "'$ITEM' é outro tipo de item (link, dispositivo, etc.)."
    fi
    # Lista as permissões do arquivo ou diretório
    ls -l "$ITEM"
else
    echo "Erro: '$ITEM' não existe."
fi
```

```
PDITA681@PDITA3409L: ~/Documentos/Scripts
PDITA681@PDITA3409L:~/Documentos/Scripts$ nano jogo_adivinhacao.sh
PDITA681@PDITA3409L:~/Documentos/Scripts$ chmod u+x jogo_adivinhacao.sh
PDITA681@PDITA3409L:~/Documentos/Scripts$ ./jogo_adivinhacao.sh
Adivinhe um número entre 1 e 100: 5
Tente um número maior!
Adivinhe um número entre 1 e 100: 41
Tente um número menor!
Adivinhe um número entre 1 e 100: 10
Tente um número maior!
Adivinhe um número entre 1 e 100: 15
Tente um número maior!
Adivinhe um número entre 1 e 100: 25
Tente um número maior!
Adivinhe um número entre 1 e 100: 30
Tente um número menor!
Adivinhe um número entre 1 e 100: 29
Parabéns! Você acertou em 7 tentativas!
PDITA681@PDITA3409L:~/Documentos/Scripts$
```

10. Escreva um script que imprima a quantidade de processos em execução. DICA: Use o comando wc para contar linha

```
27 de fev
GNU nano 7.2 contar_processos.sh *
#!/bin/bash
echo "Quantidade de processos em execução: $(ps aux | wc -l)"
```

```
27 de fev
PDITA681@PDITA3409L: ~/Documentos/Scripts
PDITA681@PDITA3409L:~/Documentos/Scripts$ nano contar_processo.sh
PDITA681@PDITA3409L:~/Documentos/Scripts$ chmod u+x contar_processo.sh
PDITA681@PDITA3409L:~/Documentos/Scripts$ ./contar_processo.sh
Quantidade de processos em execução: 264
PDITA681@PDITA3409L:~/Documentos/Scripts$
```