

Introdução à Ciência da Computação

Shell Script – parte V

Professor Iago Augusto de Carvalho
iago.carvalho@unifal-mg.edu.br

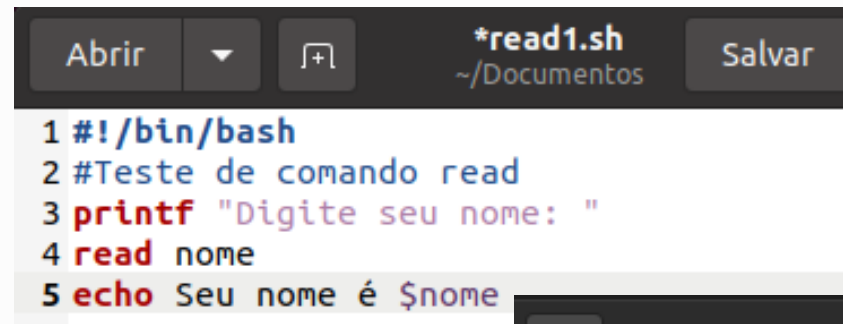
Comando read

O comando read aceita entrada de dados da entrada padrão ou de outro descritor de arquivos.

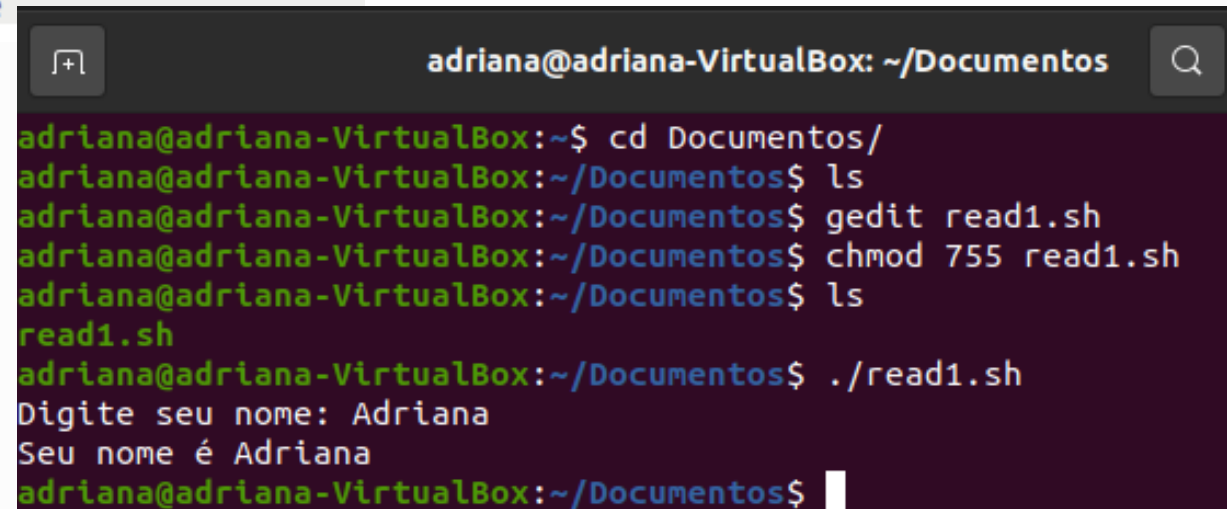
Após receber a entrada, o comando read coloca os dados em uma variável padrão.

Exemplo:

```
#!/bin/bash
#Teste de comando read
printf "Digite seu nome: "
read nome
echo Seu nome é $nome
```

A screenshot of a code editor window titled '*read1.sh' with a path of '~/Documentos'. The editor contains five lines of shell script code:

```
1 #!/bin/bash
2 #Teste de comando read
3 printf "Digite seu nome: "
4 read nome
5 echo Seu nome é $nome
```

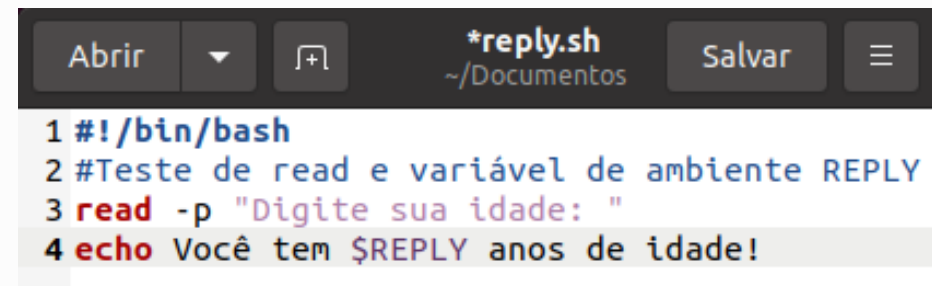
A screenshot of a terminal window with the title 'adriana@adriana-VirtualBox: ~/Documentos'. The terminal shows the following sequence of commands and output:

```
adriana@adriana-VirtualBox:~$ cd Documentos/
adriana@adriana-VirtualBox:~/Documentos$ ls
adriana@adriana-VirtualBox:~/Documentos$ gedit read1.sh
adriana@adriana-VirtualBox:~/Documentos$ chmod 755 read1.sh
adriana@adriana-VirtualBox:~/Documentos$ ls
read1.sh
adriana@adriana-VirtualBox:~/Documentos$ ./read1.sh
Digite seu nome: Adriana
Seu nome é Adriana
adriana@adriana-VirtualBox:~/Documentos$
```

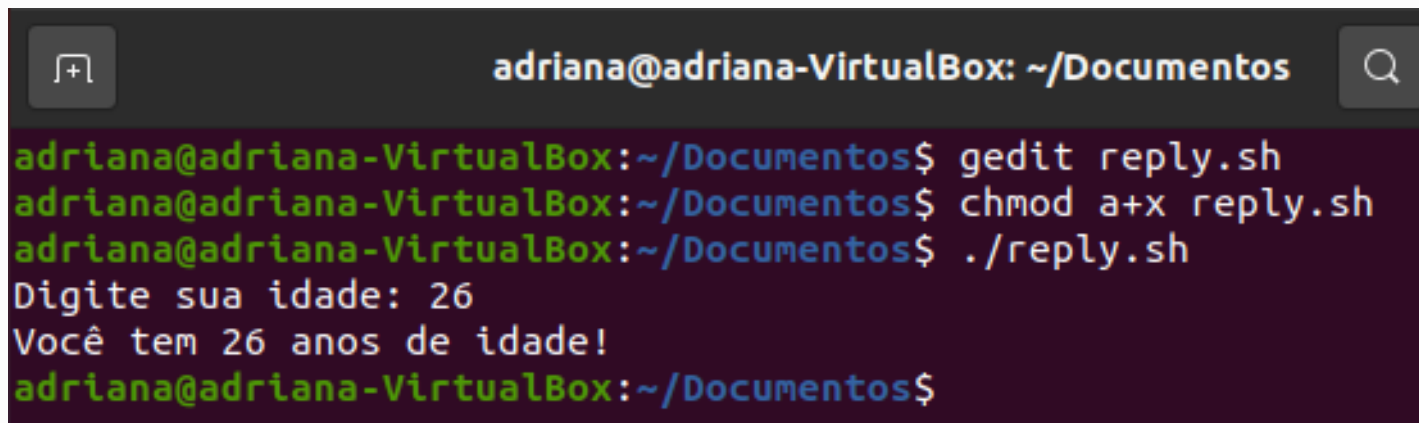
Comando read

Se não for especificada nenhuma variável para receber os dados no comando read, então o valor lido será colocado na variável de ambiente REPLY:

```
#!/bin/bash
#Teste de read e variável de ambiente REPLY
read -p "Digite sua idade: "
echo Você tem $REPLY anos de idade!
```



A screenshot of a code editor window titled '*reply.sh' with the path '~/Documentos'. The editor contains four lines of code: 1. `#!/bin/bash`, 2. `#Teste de read e variável de ambiente REPLY`, 3. `read -p "Digite sua idade: "`, and 4. `echo Você tem $REPLY anos de idade!`. The code is syntax-highlighted with blue for comments, red for keywords, and black for strings and variables.



A screenshot of a terminal window titled 'adriana@adriana-VirtualBox: ~/Documentos'. The terminal shows the execution of the script: `adriana@adriana-VirtualBox:~/Documentos$ gedit reply.sh`, `adriana@adriana-VirtualBox:~/Documentos$ chmod a+x reply.sh`, and `adriana@adriana-VirtualBox:~/Documentos$./reply.sh`. The output of the script is displayed: `Digite sua idade: 26` followed by `Você tem 26 anos de idade!`. The prompt `adriana@adriana-VirtualBox:~/Documentos$` is shown at the bottom.

Comando read com timer

É possível especificar um valor de timeout para o comando read. Para isso, usamos a opção `-t` para passar o número de segundos que o comando aguardará uma entrada do usuário. Quanto o timer expira, o read retorna status de saída não-zero.

```
#!/bin/bash
#Usando timer
echo Digite seu nome:
if read -t 4 nome
then
    echo Olá $nome, tudo bem?
else
    echo Você demorou mais de 4 segundos para responder!
fi
```

Comando read com timer

```
Abrir ▼ [ícone] *timer.sh ~/Documentos Salvar
1 #!/bin/bash
2 #usando timer
3 echo Digite seu nome:
4 if read -t 4 nome
5 then
6     echo "Olá $nome, tudo bem?"
7 else
8     echo "Você demorou mais de 4 segundos para responder!"
9 fi
```

```
[ícone] adriana@adriana-VirtualBox: ~/Documentos 🔍
adriana@adriana-VirtualBox:~/Documentos$ gedit timer.sh
adriana@adriana-VirtualBox:~/Documentos$ chmod 755 timer.sh
adriana@adriana-VirtualBox:~/Documentos$ ./timer.sh
Digite seu nome:
Você demorou mais de 4 segundos para responder!
adriana@adriana-VirtualBox:~/Documentos$ █
```

```
[ícone] adriana@adriana-VirtualBox: ~/Documentos 🔍
adriana@adriana-VirtualBox:~/Documentos$ ./timer.sh
Digite seu nome:
Adriana
Olá Adriana, tudo bem?
adriana@adriana-VirtualBox:~/Documentos$
```

Comando read

Limitando número de caracteres digitados

É possível também fazer com que o comando read conte o número de caracteres de entrada, e saia imediatamente quando um número predefinido for digitado, atribuindo os dados recolhidos a uma variável.

```
#!/bin/bash
#Lendo um caracter (-n1 significa um caracter lido)
echo Deseja prosseguir com a operação [s/n]?
read -n1 resposta
case $resposta in
    S | s ) echo
              echo Você escolheu continuar;;
    N | n ) echo
              echo Operação interrompida;;

    * )  echo
          echo Opção inválida. Digite apenas s ou n;;
esac
```

Comando read

Limitando número de caracteres digitados

```
Abrir  *read2.sh  Salvar
~/Documentos

1 #!/bin/bash
2 #Lendo um caracter (-n1 significa um caracter lido)
3 echo "Deseja prosseguir com a operação [s/n]?"
4 read -n1 resposta
5 case $resposta in
6     S | s) echo
7             echo Você escolheu continuar;;
8     N | n) echo
9             echo Operação interrompida;;
10    *) echo
11        echo Opção inválida. Digite apenas s ou n;;
12 esac
```

```
adriana@adriana-VirtualBox: ~/Documentos

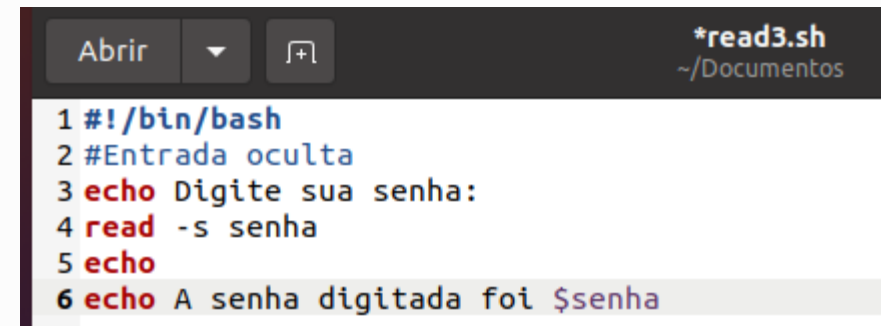
adriana@adriana-VirtualBox:~/Documentos$ gedit read2.sh
adriana@adriana-VirtualBox:~/Documentos$ chmod a+x read2.sh
adriana@adriana-VirtualBox:~/Documentos$ ./read2.sh
Deseja prosseguir com a operação [s/n]?
s
Você escolheu continuar
adriana@adriana-VirtualBox:~/Documentos$ ./read2.sh
Deseja prosseguir com a operação [s/n]?
n
Operação interrompida
adriana@adriana-VirtualBox:~/Documentos$ ./read2.sh
Deseja prosseguir com a operação [s/n]?
y
Opção inválida. Digite apenas s ou n
adriana@adriana-VirtualBox:~/Documentos$
```

Comando read

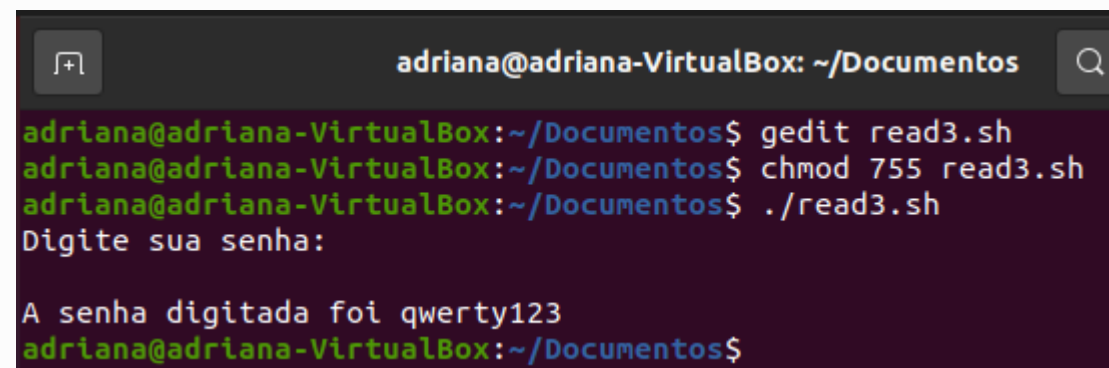
Leitura silenciosa de dados

É possível ler o que o usuário digita, sem que apareçam os caracteres na tela. Esse recurso é útil, por exemplo, para capturarmos senhas. Para isso, usamos a opção -s:

```
#!/bin/bash
#Entrada oculta
echo Digite sua senha
read -s senha
echo
echo A senha digitada foi $senha
```

A screenshot of a code editor window titled '*read3.sh' with the path '~/Documentos'. The editor contains a bash script with six lines: 1 #!/bin/bash, 2 #Entrada oculta, 3 echo Digite sua senha:, 4 read -s senha, 5 echo, and 6 echo A senha digitada foi \$senha. The script is displayed with syntax highlighting.

```
1 #!/bin/bash
2 #Entrada oculta
3 echo Digite sua senha:
4 read -s senha
5 echo
6 echo A senha digitada foi $senha
```

A screenshot of a terminal window with the title 'adriana@adriana-VirtualBox: ~/Documentos'. It shows the execution of the script: 'gedit read3.sh', 'chmod 755 read3.sh', and './read3.sh'. The output shows the prompt 'Digite sua senha:' followed by a blank line (password input), and then 'A senha digitada foi qwerty123'.

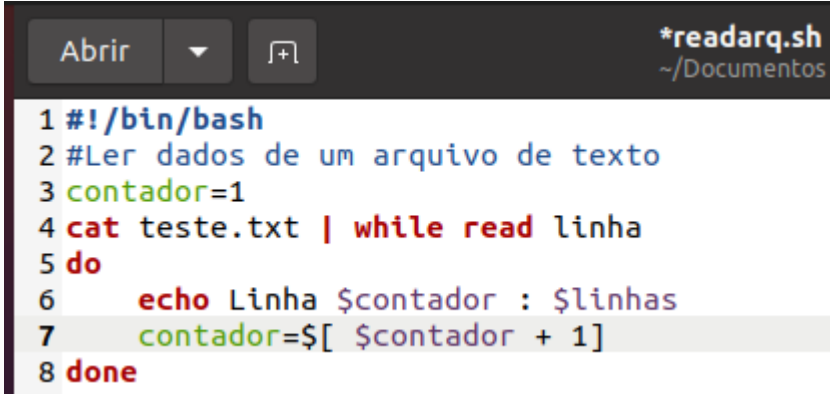
```
adriana@adriana-VirtualBox:~/Documentos$ gedit read3.sh
adriana@adriana-VirtualBox:~/Documentos$ chmod 755 read3.sh
adriana@adriana-VirtualBox:~/Documentos$ ./read3.sh
Digite sua senha:
A senha digitada foi qwerty123
adriana@adriana-VirtualBox:~/Documentos$
```


Comando read

Ler dados de um arquivo

Podemos usar o comando read para ler dados armazenados em um arquivo no computador. Cada chamada ao comando read lê uma linha de texto. Quando não houverem mais linhas para ler, o comando read sairá com status de saída zero.

```
#!/bin/bash
#Ler dados de um arquivo de texto:
contador=1
cat teste.txt | while read linha
do
    echo Linha $contador : $linha
    contador=$(( $contador + 1 ))
done
```



```
1 #!/bin/bash
2 #Ler dados de um arquivo de texto
3 contador=1
4 cat teste.txt | while read linha
5 do
6     echo Linha $contador : $linha
7     contador=$(( $contador + 1 ))
8 done
```

Comando read

Ler dados de um arquivo

```
Abrir ▼ +l *readarq.sh ~/Documentos
1 #!/bin/bash
2 #Ler dados de um arquivo de texto
3 contador=1
4 cat teste.txt | while read linha
5 do
6     echo Linha $contador : $linhas
7     contador=$(( $contador + 1 ))
8 done
```

```
adriana@adriana-VirtualBox: ~/Documentos
adriana@adriana-VirtualBox:~/Documentos$ gedit readarq.sh
adriana@adriana-VirtualBox:~/Documentos$ chmod 755 readarq.sh
adriana@adriana-VirtualBox:~/Documentos$ ./readarq.sh
Linha 1 : Castelo De Areia
Linha 2 : Num dia de verão, estava na praia,
Linha 3 : observando duas crianças brincando na
Linha 4 : areia. Elas trabalhavam muito,
Linha 5 : construindo um castelo de areia, com torres,
Linha 6 : passarelas e passagens internas.
Linha 7 : Quando estavam quase acabando, veio uma
Linha 8 : onda e destruiu tudo, reduzindo o
Linha 9 : castelo a um monte de areia e espuma.
Linha 10 : Achei que, depois de tanto esforço e cuidado,
Linha 11 : as crianças caíam no choro.
Linha 12 : Mas tive uma surpresa.
adriana@adriana-VirtualBox:~/Documentos$
```

Funções

1ª sintaxe:

```
function nome_função {  
    comandos da função  
}
```

2ª sintaxe:

```
nome_função() {  
    comandos da função  
}
```

Funções

```
func.sh
1 #!/bin/bash
2 #Função em um script so shell
3
4 function mensagem {
5     echo "Aula ICC"
6 }
7
8 #Chamando a função dez vezes usando um contador:
9 contador=1
10 while [ $contador -le 10 ]
11 do
12     mensagem
13     contador=$(( contador + 1 ))
14 done
15
16 #Outro texto, indicando o fim do script:
17 echo "Função executada com sucesso!"
```

```
adriana@adriana-VirtualBox: ~
adriana@adriana-VirtualBox:~$ gedit func.sh
adriana@adriana-VirtualBox:~$ chmod a+x func.sh
adriana@adriana-VirtualBox:~$ ./func.sh
Aula ICC
Aula ICC
Aula ICC
Aula ICC
Aula ICC
Aula ICC
Aula ICC
Aula ICC
Aula ICC
Aula ICC
Função executada com sucesso!
adriana@adriana-VirtualBox:~$
```

Funções

```
Abrir  funcao.sh  Salvar
1 #!/bin/bash
2 #Function Output
3
4 function quadrado {
5     read -p "Digite um número entre 20 e 30: " numero
6     echo $[ $numero * $numero ]
7 }
8
9 #Atribuindo o valor da função à variável do shell valor:
10 valor=`quadrado`
11
12 #Mostrando o valor:
13 echo "O quadrado do número é $valor"
```

```
adriana@adriana-VirtualBox: ~
adriana@adriana-VirtualBox:~$ gedit funcao.sh
adriana@adriana-VirtualBox:~$ chmod 755 funcao.sh
adriana@adriana-VirtualBox:~$ ./funcao.sh
Digite um número entre 20 e 30: 21
O quadrado do número é 441
```

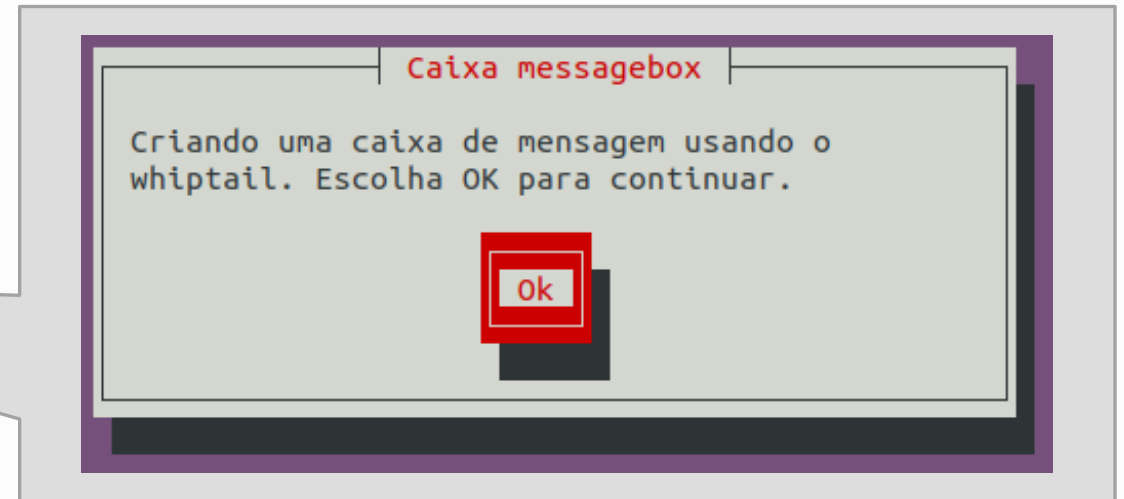
Nome da função entre crases (backtick`)

Caixas de diálogo com whiptail

```
1 #!/bin/bash
2 whiptail --title "Caixa messagebox" --msgbox "Criando uma caixa de mensagem usando o whiptail. Escolha OK para continuar." --fb 10 50
```

Linhas Colunas

```
adriana@adriana-VirtualBox:~$ gedit msg.sh
adriana@adriana-VirtualBox:~$ chmod 755 msg.sh
adriana@adriana-VirtualBox:~$ ./msg.sh
adriana@adriana-VirtualBox:~$
```



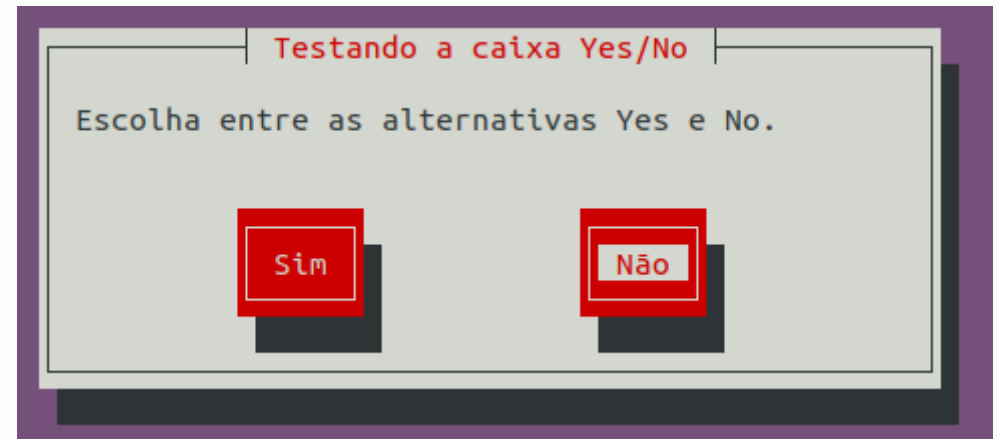
Pressione a tecla enter para voltar

Caixas de diálogo com whiptail

```
simounao.sh
Abrir  Salvar  -  [X]

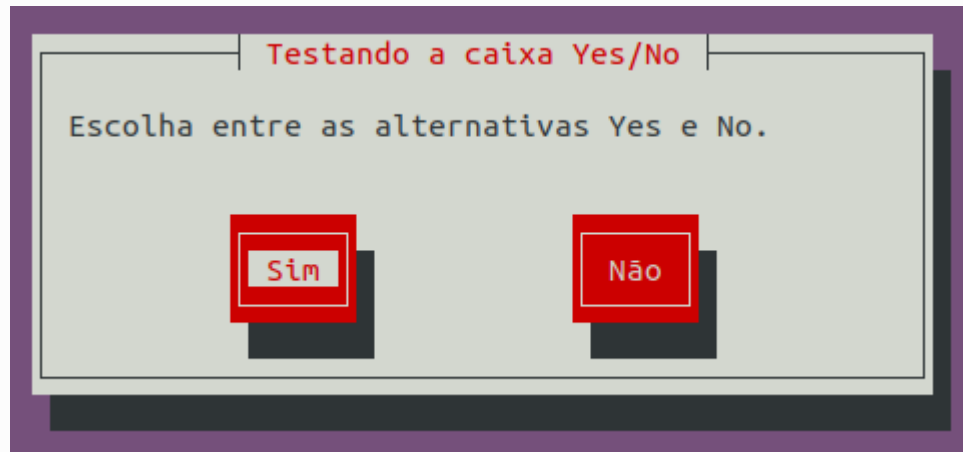
1 #!/bin/bash
2 if whiptail --title "Testando a caixa Yes/No" --yesno "Escolha entre as
  alternativas Yes e No." --fb 10 50
3 then
4     echo "Você escolheu Yes. O status de saída é $?."
5 else
6     echo "Você escolheu No. O status de saída é $?."
7 fi
```

```
adriana@adriana-VirtualBox: ~
adriana@adriana-VirtualBox:~$ gedit simounao.sh
adriana@adriana-VirtualBox:~$ chmod a+x simounao.sh
adriana@adriana-VirtualBox:~$ ./simounao.sh
Você escolheu No. O status de saída é 1.
adriana@adriana-VirtualBox:~$
```



Utilize as teclas ← e → e pressione a tecla enter.

Caixas de diálogo com whiptail



```
adriana@adriana-VirtualBox: ~  
adriana@adriana-VirtualBox:~$ ./simounao.sh  
Você escolheu Yes. O status de saída é 0.  
adriana@adriana-VirtualBox:~$
```

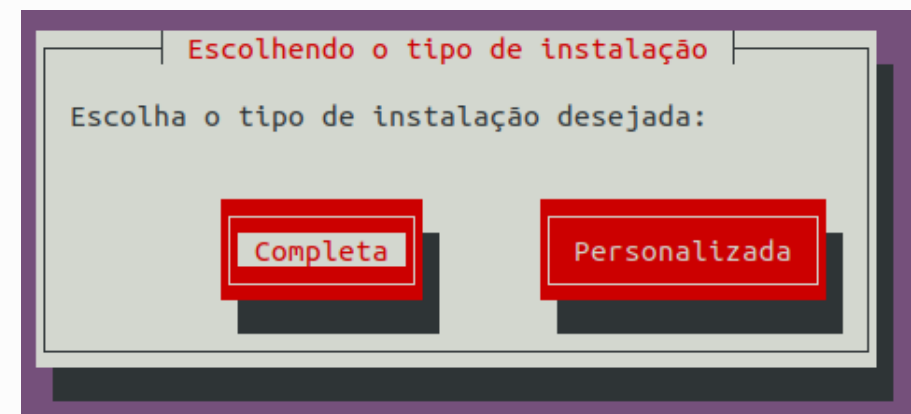
A opção **-fb** permite o uso de botões completos (por padrão, o whiptail usa uma versão simplificada dos botões).

Utilize as teclas ← e → e pressione a tecla enter.

Caixas de diálogo com whiptail

```
Abrir  btnperso.sh  Salvar  -  X
1 #!/bin/bash
2 if whiptail --title "Escolhendo o tipo de instalação" --yes-button "Completa"
  --no-button "Personalizada" --yesno "Escolha o tipo de instalação desejada:"
  --fb 10 50
3 then
4     echo "Você escolheu completa. O status de saída é $?."
5 else
6     echo "Você escolheu personalizada. O status de saída é $?."
7 fi
```

```
adriana@adriana-VirtualBox: ~
adriana@adriana-VirtualBox:~$ gedit btnperso.sh
adriana@adriana-VirtualBox:~$ chmod 755 btnperso.sh
adriana@adriana-VirtualBox:~$ ./btnperso.sh
Você escolheu completa. O status de saída é 0.
adriana@adriana-VirtualBox:~$
```

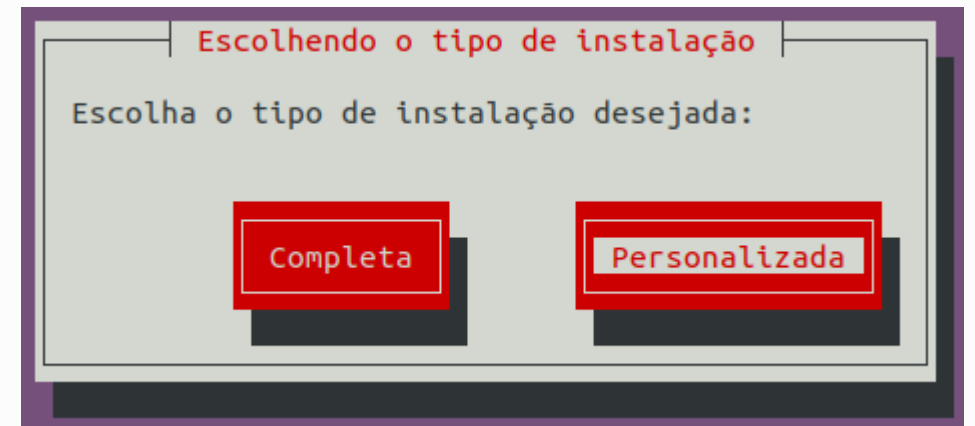


Utilize as teclas ← e → e pressione a tecla enter.

Caixas de diálogo com whiptail

```
Abrir  btnperso.sh  Salvar  -  X
1 #!/bin/bash
2 if whiptail --title "Escolhendo o tipo de instalação" --yes-button "Completa"
  --no-button "Personalizada" --yesno "Escolha o tipo de instalação desejada:"
  --fb 10 50
3 then
4     echo "Você escolheu completa. O status de saída é $?."
5 else
6     echo "Você escolheu personalizada. O status de saída é $?."
7 fi
```

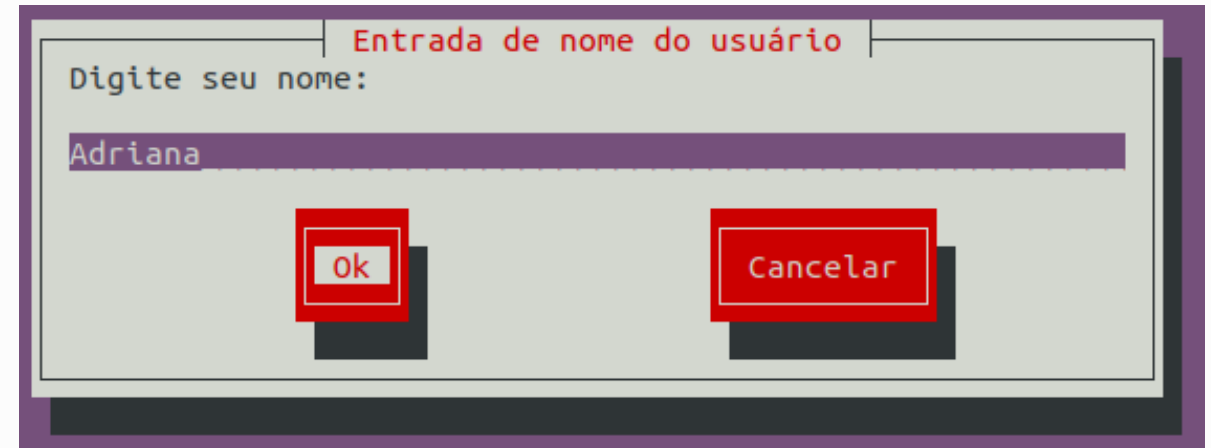
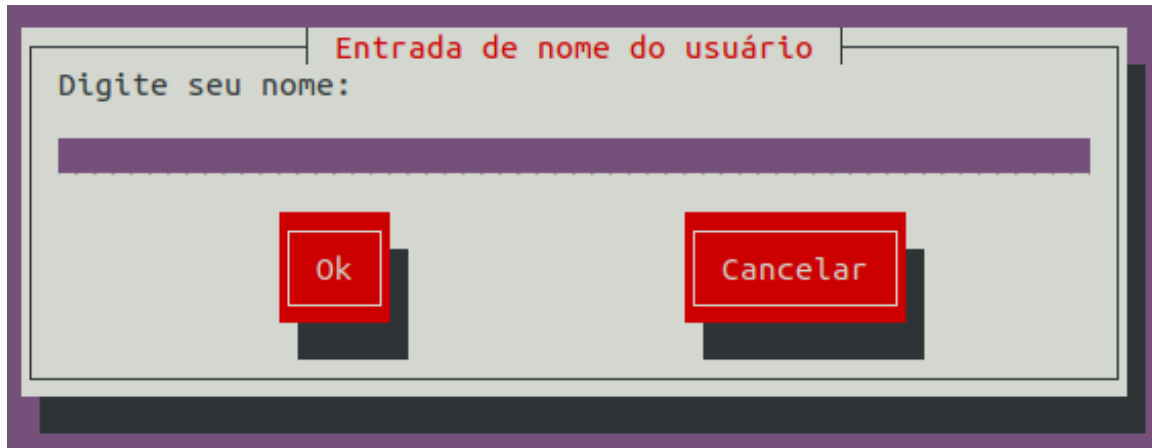
```
adriana@adriana-VirtualBox: ~
adriana@adriana-VirtualBox:~$ ./btnperso.sh
Você escolheu personalizada. O status de saída é 1.
adriana@adriana-VirtualBox:~$
```



Utilize as teclas ← e → e pressione a tecla enter.

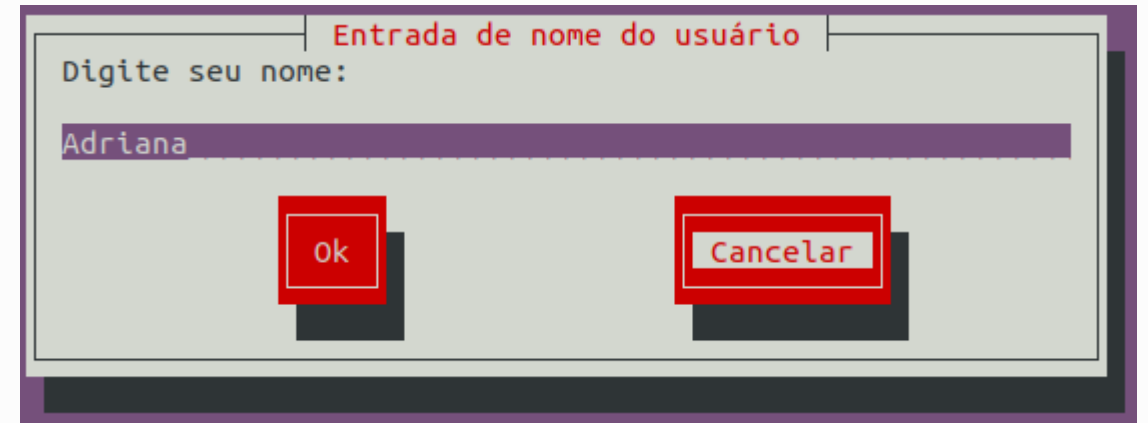
Caixas de diálogo com whiptail

```
1 #!/bin/bash
2 nome=$(whiptail --title "Entrada de nome do usuário" --inputbox "Digite seu
  nome:" --fb 10 60 3>&1 1>&2 2>&3)
3 statussaida=$?
4 if [ $statussaida = 0 ]
5 then
6     echo "O nome digitado foi: $nome"
7 else
8     echo "Entrada cancelada pelo usuário"
9 fi
```



Caixas de diálogo com whiptail

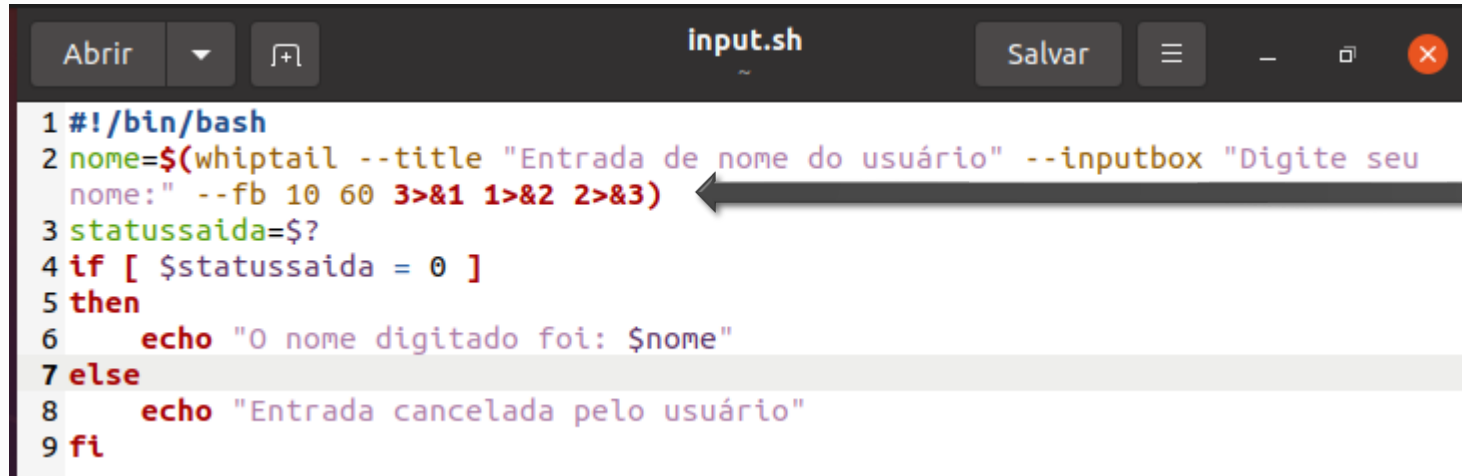
```
adriana@adriana-VirtualBox: ~  
adriana@adriana-VirtualBox:~$ gedit input.sh  
adriana@adriana-VirtualBox:~$ chmod 755 input.sh  
adriana@adriana-VirtualBox:~$ ./input.sh  
O nome digitado foi: Adriana  
adriana@adriana-VirtualBox:~$
```



```
adriana@adriana-VirtualBox: ~  
adriana@adriana-VirtualBox:~$ ./input.sh  
Entrada cancelada pelo usuário  
adriana@adriana-VirtualBox:~$
```

Utilize as teclas ←, → e ↓ para mover.

Caixas de diálogo com whiptail



```
1 #!/bin/bash
2 nome=$(whiptail --title "Entrada de nome do usuário" --inputbox "Digite seu
  nome:" --fb 10 60 3>&1 1>&2 2>&3)
3 statussaida=$?
4 if [ $statussaida = 0 ]
5 then
6     echo "O nome digitado foi: $nome"
7 else
8     echo "Entrada cancelada pelo usuário"
9 fi
```

O que significa **3>&1 1>&2 2>&3** em um script do Shell no Linux?

3>&1 – Cria um novo descritor de arquivos e o redireciona para a saída padrão (1)

1>&2 – Redireciona a saída padrão para o erro padrão (2)

2>&3 – Redireciona o erro padrão para o descritor criado, que agora irá redirecionar para a saída padrão.

<http://www.bosontreinamentos.com.br/shell-script/o-que-significa-31-12-23-em-um-script-do-shell-no-linux/>

Caixas de diálogo com whiptail

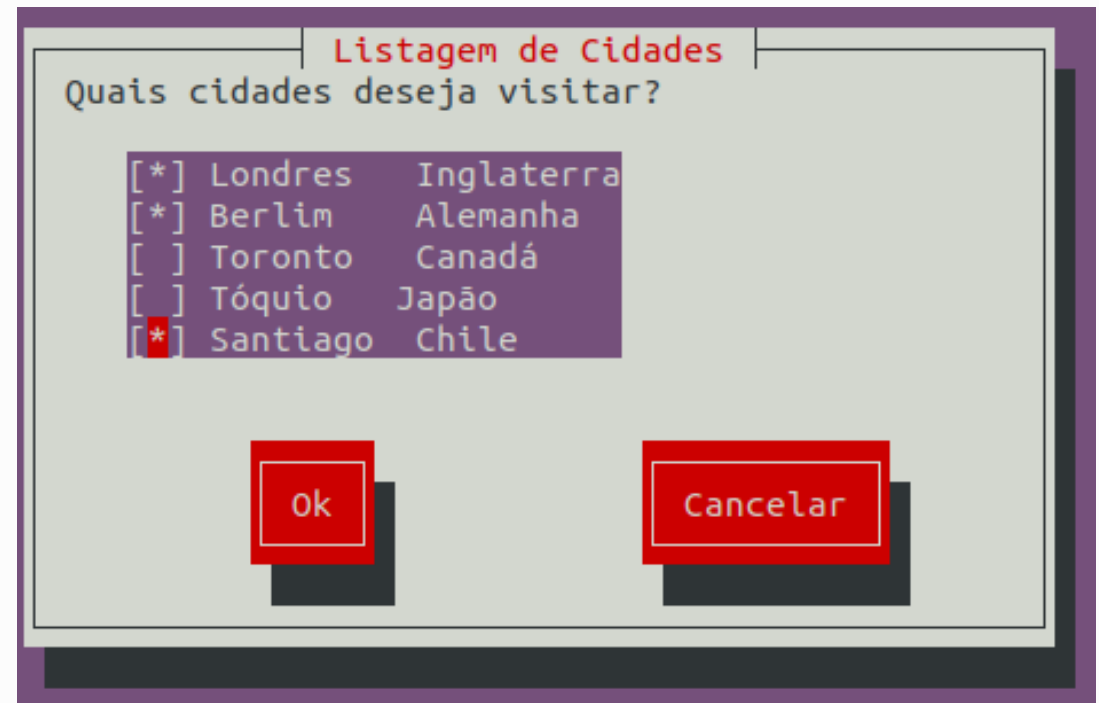
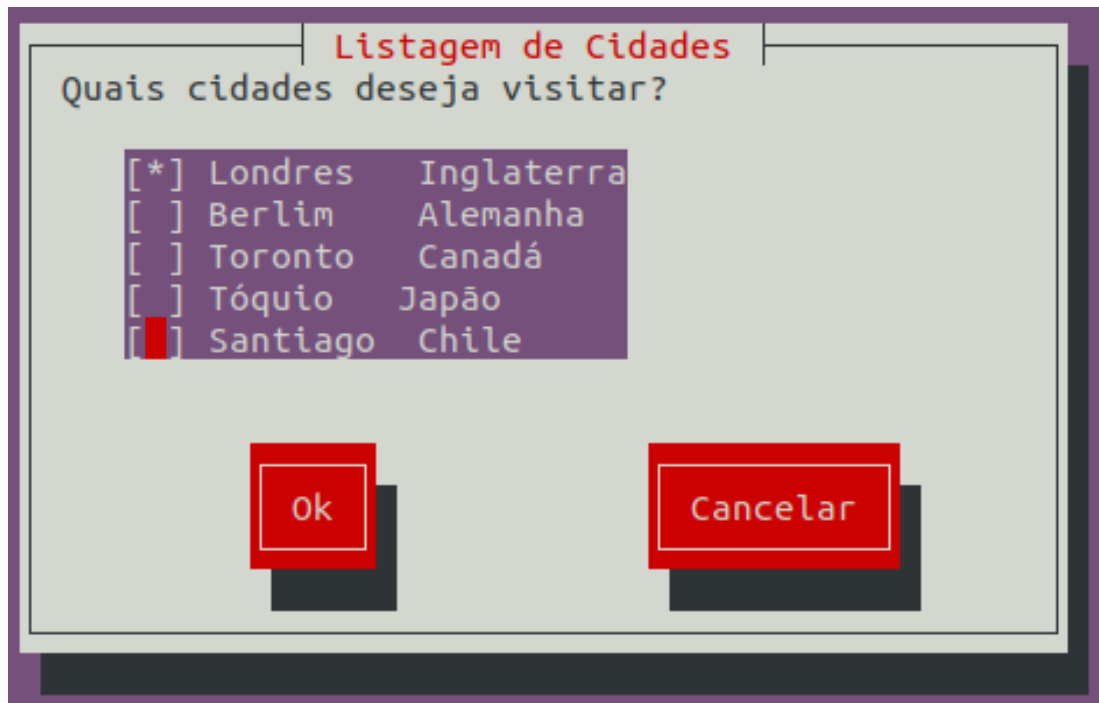
```
1 #!/bin/bash
2 cidade=$(whiptail --title "Listagem de Cidades" --checklist --fb \
3 "Quais cidades deseja visitar?" 15 50 5 \
4 "Londres" "Inglaterra" ON \
5 "Berlim" "Alemanha" OFF \
6 "Toronto" "Canadá" OFF \
7 "Tóquio" "Japão" OFF \
8 "Santiago" "Chile" OFF 3>&1 1>&2 2>&3)
9 status=$?
10 if [ $status = 0 ]
11 then
12     echo "As cidades escolhidas foram: " $cidade
13 else
14     echo "Entrada cancelada pelo usuário."
15 fi
```

As \ indicam que continua na linha abaixo.

Londres é tag
Inglaterra é item

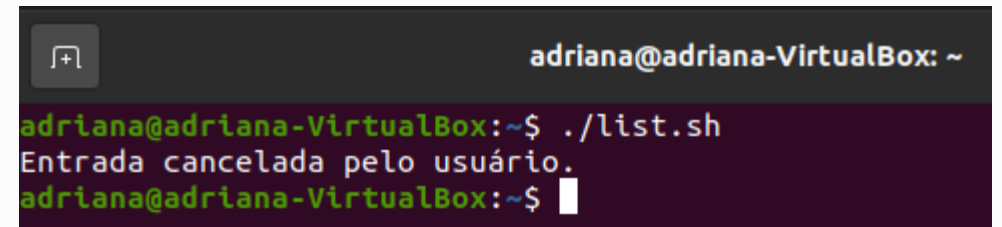
```
adriana@adriana-VirtualBox: ~
adriana@adriana-VirtualBox:~$ gedit list.sh
adriana@adriana-VirtualBox:~$ chmod a+x list.sh
adriana@adriana-VirtualBox:~$ ./list.sh
As cidades escolhidas foram: "Londres" "Berlim" "Santiago"
adriana@adriana-VirtualBox:~$ ./list.sh
As cidades escolhidas foram: "Londres"
adriana@adriana-VirtualBox:~$
```

Caixas de diálogo com whiptail



Utilize as teclas ←, → e ↓ para mover. E espaço para selecionar.

Caixas de diálogo com whiptail



Utilize a tecla tab para mover.

Caixas de diálogo com dialog

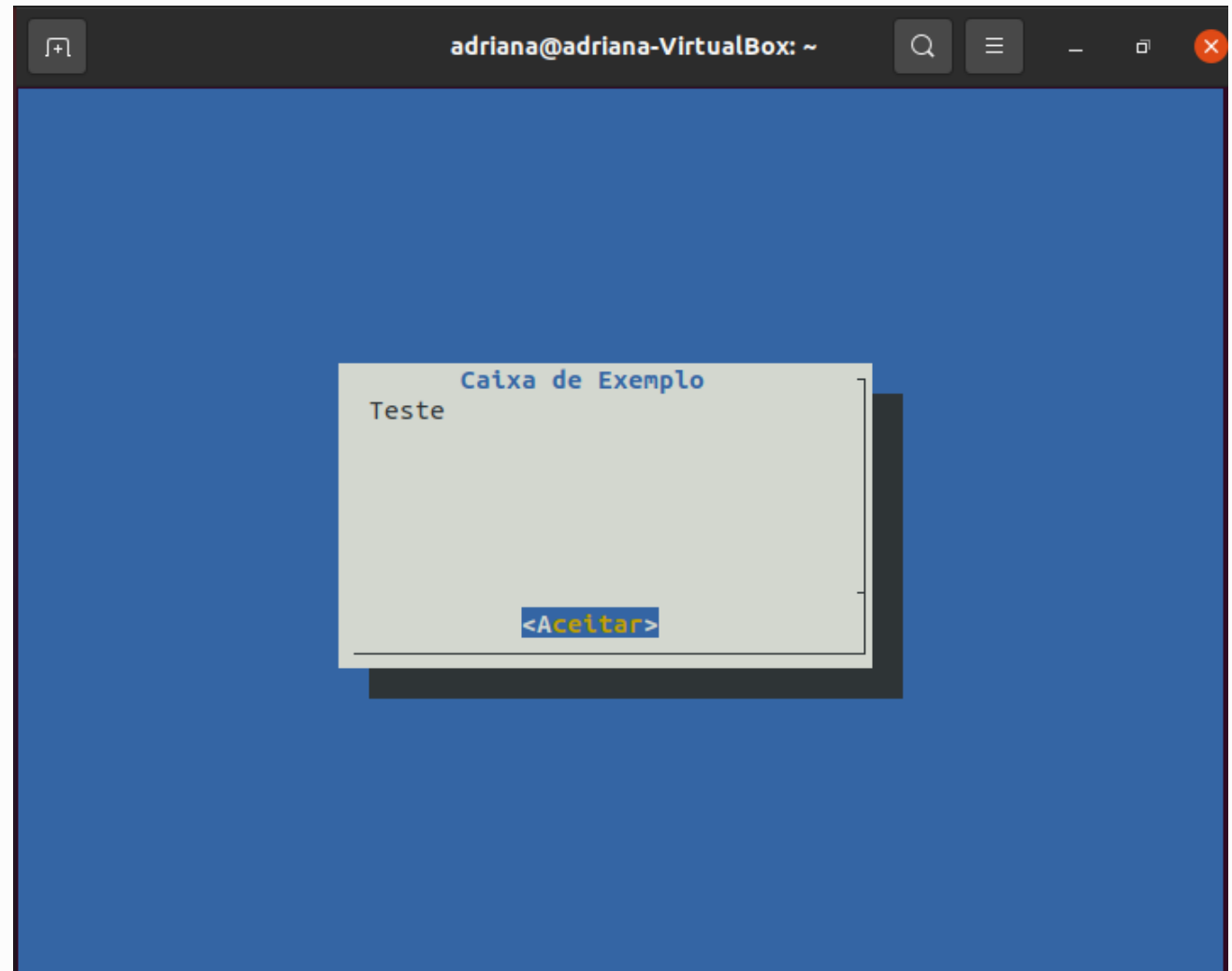
O comando dialog utiliza parâmetros de linha de comando para determinar que tipo de widget de janela deve ser criada. Um widget é um tipo de elemento de janela.

Widget	Descrição
calendar	Fornece um calendário que permite selecionar datas
checklist	Mostra entradas múltiplas onde cada entrada pode ser ligada ou desligada
form	Permite construir um formulário com labels e campos de texto para serem preenchidos
fselect	Fornece uma janela de seleção de arquivos
gauge	Mostra uma barra de progresso
infobox	Mostra uma mensagem sem esperar por uma resposta
inputmenu	Fornece um menu editável
menu	Mostra uma lista de seleções para escolha
msgbox	Mostra uma mensagem e pede que o usuário pressione um botão OK
passwordbox	Mostra uma caixa de texto simples que esconde o texto digitado
radiolist	Fornece um grupo de itens de menu onde apenas um item pode ser selecionado
tailbox	Mostra o texto de um arquivo em uma janela com rolagem usando o comando tail
textbox	Mostra o conteúdo de um arquivo em uma janela com rolagem
timebox	Fornece uma janela para selecionarmos uma hora, minuto e segundo
yesno	Fornece uma mensagem simples com botões Yes e No

Caixas de diálogo com dialog

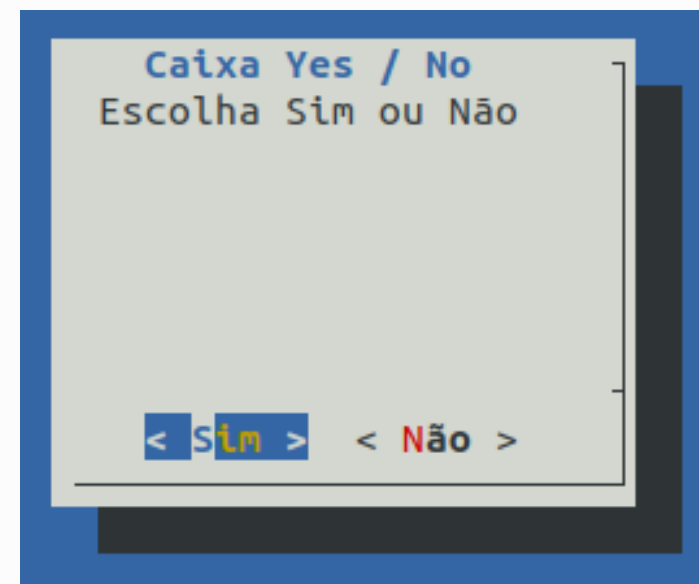
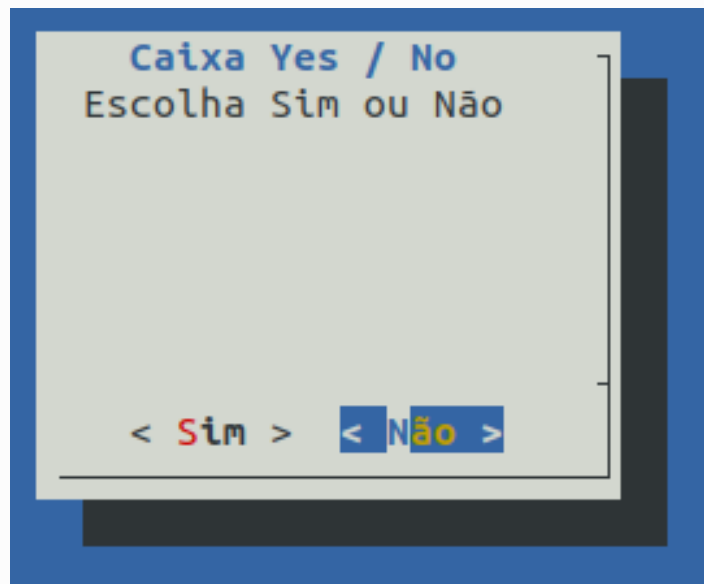
```
adriana@adriana-VirtualBox:~$ sudo apt-get install dialog
```

```
adriana@adriana-VirtualBox: ~  
adriana@adriana-VirtualBox:~$ dialog --title "Caixa de Exemplo" --msgbox "Teste"  
" 10 35
```



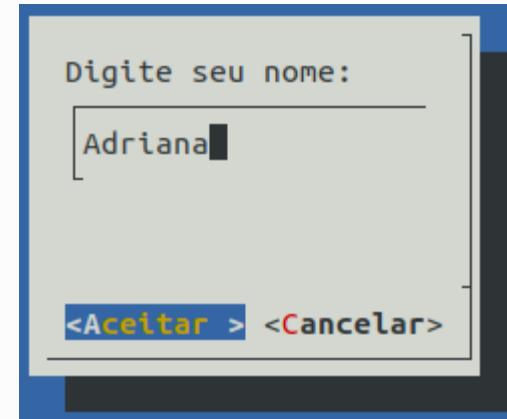
Caixas de diálogo com dialog

```
adriana@adriana-VirtualBox: ~  
adriana@adriana-VirtualBox:~$ dialog --title "Caixa Yes / No" --yesno "Escolha  
Sim ou Não" 10 25
```



Caixas de diálogo com dialog

```
adriana@adriana-VirtualBox: ~  
adriana@adriana-VirtualBox:~$ nome=$(dialog --inputbox "Digite seu nome:" 10 25  
--stdout)
```

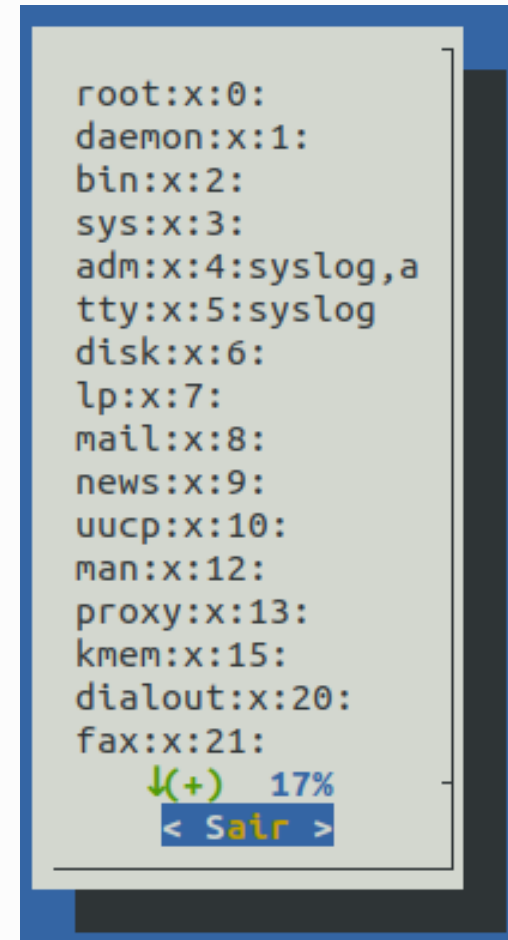


```
adriana@adriana-VirtualBox: ~  
adriana@adriana-VirtualBox:~$ echo $?  
0  
adriana@adriana-VirtualBox:~$
```

```
adriana@adriana-VirtualBox: ~  
adriana@adriana-VirtualBox:~$ echo $nome  
Adriana  
adriana@adriana-VirtualBox:~$
```

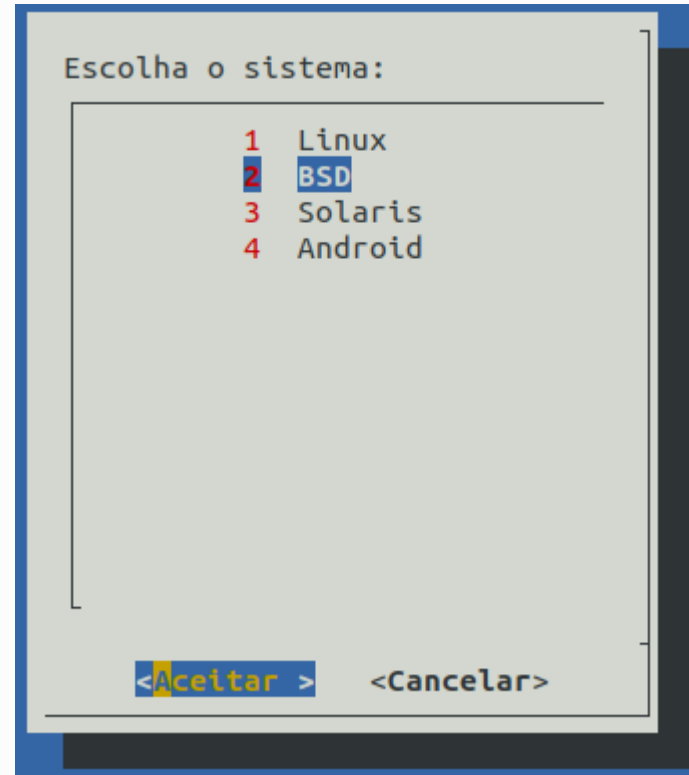
Caixas de diálogo com dialog

```
adriana@adriana-VirtualBox: ~  
adriana@adriana-VirtualBox:~$ dialog --textbox /etc/group 20 20
```



Caixas de diálogo com dialog

```
adriana@adriana-VirtualBox: ~  
adriana@adriana-VirtualBox:~$ sistema=$(dialog --menu "Escolha o sistema:" 20 3  
5 15 \  
> 1 "Linux" \  
> 2 "BSD" \  
> 3 "Solaris" \  
> 4 "Android" --stdout)
```



```
adriana@adriana-VirtualBox: ~  
adriana@adriana-VirtualBox:~$ echo $sistema  
2  
adriana@adriana-VirtualBox:~$
```

Caixas de diálogo com dialog

```
adriana@adriana-VirtualBox: ~  
adriana@adriana-VirtualBox:~$ data=$(dialog --calendar "Escolha a data" 0 0 --stdout)
```

Escolha a data

Mês Ano

abril 2021

↑(-)

	dom	seg	ter	qua	qui	sex	sá
14					1	2	3
15	4	5	6	7	8	9	10
16	11	12	13	14	15	16	17
17	18	19	20	21	22	23	24
18	25	26	27	28	29	30	

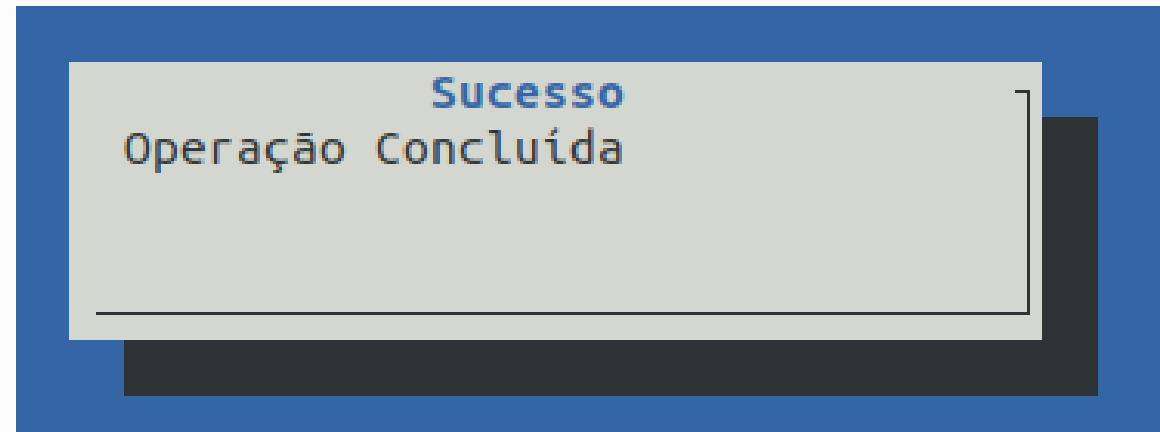
↓(+)

<Aceitar> <Cancelar>

```
adriana@adriana-VirtualBox: ~  
adriana@adriana-VirtualBox:~$ echo $data  
30/04/2021  
adriana@adriana-VirtualBox:~$
```

Caixas de diálogo com dialog

```
adriana@adriana-VirtualBox: ~  
adriana@adriana-VirtualBox:~$ dialog --title "Sucesso" --infobox "Operação Concluída" 5 35; sleep 4; clear
```



Caixas de diálogo com dialog

```
adriana@adriana-VirtualBox: ~  
adriana@adriana-VirtualBox:~$ itens=$(dialog --radiolist "Escolh a cor:" 15 30  
4 \  
"Verde" "" OFF \  
"Branco" "" OFF \  
"Vermelho" "" ON \  
"Rosa" "" OFF --stdout)
```



```
adriana@adriana-VirtualBox:~$ echo $itens  
Vermelho  
adriana@adriana-VirtualBox:~$
```



```
adriana@adriana-VirtualBox:~$ echo $itens  
Verde  
adriana@adriana-VirtualBox:~$
```

Referências

PRITCHARD, S.; PESSANHA, B. G.; LANGFELDT, N.; STANGER, J.; DEAN, J. 2007. **Certificação Linux LPI Rápido e Prático. Guia de Referência nível 1: Exames 101 e 102.** 2ª Ed. Rio de Janeiro: Editora Alta Books.

Curso de Shell Scripting – Bóson Treinamentos

<http://www.bosontreinamentos.com.br/curso-de-shell-scripting/>