



**BASH**  
THE BOURNE-AGAIN SHELL

# BASH SCRIPT

GABRIEL DOS SANTOS MOREIRA  
ROBERTO FERREIRA DE ASSIS FILHO  
RODRIGO DE OLIVEIRA FARIAS

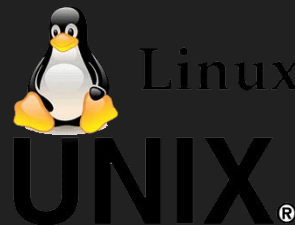
# ÍNDICE

- HISTÓRICO
- CARACTERÍSTICAS GERAIS
- ALGUNS COMANDOS BASH
- HELLO WORLD
- DECLARAÇÃO DE VARIÁVEIS
- ESTRUTURAS DE CONTROLE
  - IF - ELSE
  - LOOPS (WHILE E FOR)
  - CASE / ESAC
- MODULARIZAÇÃO (FUNÇÕES)
- APLICAÇÕES



# HISTÓRICO

- Bash (Bourne-Again Shell) é um interpretador de comandos que vem sendo usado como shell padrão em sistemas Unix e Linux há muitos anos.
- Desenvolvido por Brian Fox e continuado por Chat Ramey, o bash foi lançado em 1989 como um software livre.



# CARACTERÍSTICAS GERAIS

- O Bash é uma das muitas shells disponíveis para sistemas operacionais Unix e Linux.
- É uma implementação aprimorada do interpretador de comandos Bourne Shell (sh)
- O bash está disponível em praticamente todas as distribuições Linux e não exige nenhuma instalação à parte.



```
[~]$ csh
```

```
[~]$ tcsh
```

```
Korn Shell
Welcome to the SUA utilities.
DISPLAY=localhost:0.0
$ ls /dev/fs/C
$Recycle.Bin
BOOTSECT.BAK
Boot
Documents and Settings
PerfLogs
Program Files
ProgramData
$ -
System Volume Information
Users
Windows
autoexec.bat
bootmgr
config.sys
pagefile.sys
```

# CARACTERÍSTICAS GERAIS

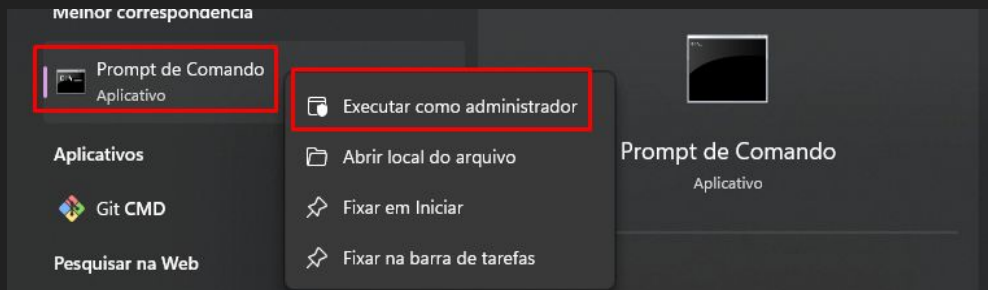
- Paradigma da linguagem é interpretado, por ser interpretada não precisa de compilador, os comandos são escritos em um arquivo de texto e executados diretamente;
- Possui tipagem dinâmica, não é preciso declaração prévia do tipo da variável;
- Possui suporte a variáveis, estruturas de controle de fluxo, funções, matrizes etc;
- Usada para manipulação de arquivos, execução de programas, muito usada para automatizar tarefas.

# Instalação – WSL no Windows

- WSL: É um recurso do windows lançado em 2016, que permite executar binários e scripts de linux diretamente no windows, múltiplas distribuições em uma mesma máquina, cada uma com uma estrutura isolada e independente do sistema e de outras distribuições.
- Diferente de uma máquina virtual, o WSL é integrado com o sistema, permite acessar os arquivos tanto do sistema como os arquivos da distribuição.

# Instalação – WSL no Windows

- 1: Executar o prompt de comando como administrador:



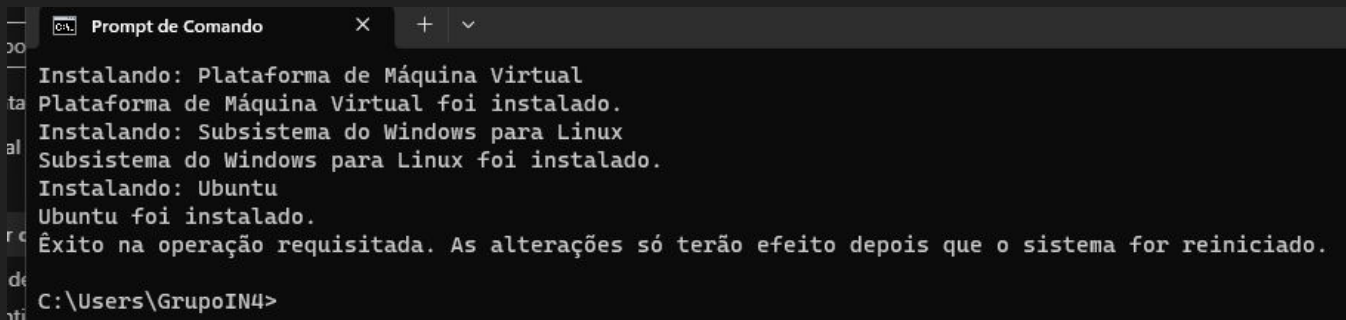
- 2: Digitar `wsl --install` no prompt e apertar enter.
- 3: Irão aparecer as seguintes mensagens no prompt até a instalação acabar:

```
CA: Administrador: Prompt de Comando - wsl --install
Microsoft Windows [versão 10.0.22621.1555]
(c) Microsoft Corporation. Todos os direitos reservados.

C:\Windows\System32>wsl --install
Instalando: Plataforma de Máquina Virtual
Plataforma de Máquina Virtual foi instalado.
Instalando: Subsistema do Windows para Linux
Subsistema do Windows para Linux foi instalado.
Instalando: Ubuntu
[===== 15,0%
```

# Instalação – WSL no Windows

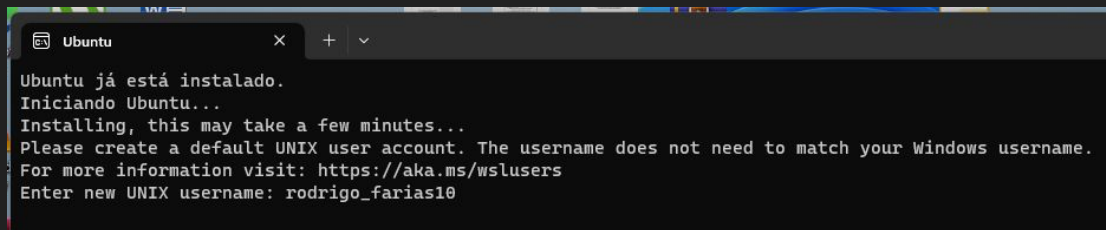
- 4: Assim que a instalação acabar, aparecerá a seguinte mensagem:



```

Prompt de Comando
Instalando: Plataforma de Máquina Virtual
Plataforma de Máquina Virtual foi instalado.
Instalando: Subsistema do Windows para Linux
Subsistema do Windows para Linux foi instalado.
Instalando: Ubuntu
Ubuntu foi instalado.
Êxito na operação requisitada. As alterações só terão efeito depois que o sistema for reiniciado.
C:\Users\GrupoIN4>
```

- 5: Deve-se reiniciar o computador.
- 6: Assim que a máquina é reiniciada, automaticamente o ubuntu vai abrir para concluir a instalação.
- 7: Após instalar o WSL, você precisará criar uma conta de usuário e uma senha para a distribuição do Linux recém instalada:



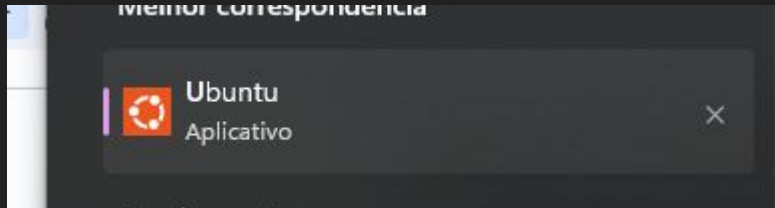
```

Ubuntu
Ubuntu já está instalado.
Iniciando Ubuntu...
Installing, this may take a few minutes...
Please create a default UNIX user account. The username does not need to match your Windows username.
For more information visit: https://aka.ms/wslusers
Enter new UNIX username: rodrigo_farias10
```



# Instalação – WSL no Windows

- 8: A distribuição do UBUNTU será instalada por padrão na máquina:



- 9: Execute o comando wsl.exe no prompt e pronto! Já podemos utilizar os comandos em bash

script:

```
root@DESKTOP-16NTK17: /mnt/c/Users/Rodrigo#  
Microsoft Windows [versão 10.0.22621.1555]  
(c) Microsoft Corporation. Todos os direitos reservados.  
  
C:\Users\Rodrigo>wsl.exe  
Welcome to Ubuntu 22.04.2 LTS (GNU/Linux 5.15.90.1-microsoft-standard-WSL2 x86_64)  
  
* Documentation:  https://help.ubuntu.com  
* Management:    https://landscape.canonical.com  
* Support:        https://ubuntu.com/advantage  
  
This message is shown once a day. To disable it please create the  
/root/.hushlogin file.  
root@DESKTOP-16NTK17: /mnt/c/Users/Rodrigo#
```

# Alguns Comandos Bash

Listar o armazenamento do diretório atual:

```
ls
```

Visualizar qual shell está utilizando:

```
echo $SHELL
```

Visualizar o histórico de comandos:

```
history
```

# SINTAXE BÁSICA

- Cada script bash deverá ter em sua primeira linha:
  - `#!/bin/bash`
- Declaração de variáveis
- Estruturas condicionais
  - Instrução *If else*
- *Estruturas de repetição*
  - Loop *While*
  - Loop *For*
- *Funções*
- *Comentários começam com o símbolo #, menos na primeira linha do script*

# HELLO WORLD

Criar um script:

```
nano <nome do script>
```

Mudar as permissões do arquivo e torná-lo executável:

```
chmod +x <caminho do arquivo>
```

Executar um script:

```
<caminho do arquivo>
```

# HELLO WORLD

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

```
echo "hello world :)"
```

# DECLARAÇÃO DE VARIÁVEIS

- ❑ Atribuição de valor
- ❑ Cuidado com espaços
- ❑ Case-sensitive

```
variaveis.sh
```

```
num=1
```

```
echo $num
```

```
str="string"
```

```
echo $str
```

```
data_atual=$(date +%Y-%m-%d)
```

```
echo $data_atual
```

# DECLARAÇÃO DE VARIÁVEIS

- ❑ As variáveis também podem ser inicializadas a partir do resultado de uma expressão

```
data_atual=$(date +%Y-%m-%d)
```

- ❑ É possível acessar o valor de uma variável usando o prefixo "\$"
- ❑ É possível acessar o valor de uma variável usando o prefixo "\$"

```
mensagem="Olá, mundo!"  
echo $mensagem
```

# DECLARAÇÃO DE VARIÁVEIS

- ❏ Adicionando parâmetros para o script na linha de comando

```
argumentos.sh *  
#!/bin/bash  
echo "Olá, " $1  
# $1 : primeiro parâmetro  
echo "Olá, " $2  
# $2 : segundo parâmetro  
echo "Olá, " $@  
# $@ : todos os parâmetros
```



# ESTRUTURAS DE CONTROLE

## ❏ Cláusula if-then

```
GNU nano 6.2          if1.sh *
num=200

if [ $num -eq 200 ]
then
    echo "A condição é verdadeira"
fi
```

# ESTRUTURAS DE CONTROLE

- ❑ Cláusula if - elif - then - else
- ❑ Operadores binários para realizar comparações:  
-eq, -ne, -lt, -le, -gt e -ge

```
GNU nano 6.2    if2.sh

num=200

if [ $num -eq 200 ]
then
    echo "Número = 200"
elif [ $num -lt 200 ]
then
    echo "Número < 200"
elif [ $num -gt 200 ]
then
    echo "Número > 200"
else
    echo "Expressão inválida"
fi
```

# Estrutura de Controle - While

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

```
myvar=1
```

```
while [ $myvar -le 10 ]  
do
```

```
    echo $myvar
```

```
    myvar=$(expr $myvar + 1)
```

```
    sleep 0.5
```

```
done
```

```
echo "Saiu do loop"
```

# ESTRUTURAS DE CONTROLE - For

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

```
for (( count=10; count>0; count-- ))  
do  
    echo $count  
done  
  
echo "saiu do for loop"
```

# ESTRUTURAS DE CONTROLE - For

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

```
nomes=("Anya Forger" "Maria" "Pedro" "Zoe Legend")
```

```
for nome in "${nomes[@]}"
```

```
do
```

```
    echo $nome
```

```
done
```

# ESTRUTURAS DE CONTROLE - case/esac

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

```
read day
```

```
case $day in
```

```
    "segunda" | "terca" | "quarta" | "quinta" | "sexta")
```

```
        echo "é um dia útil";;
```

```
    "sabado" | "domingo")
```

```
        echo "É um dia de fim de semana.";;
```

```
    *)
```

```
        echo "Dia inválido.";;
```

```
esac
```

# MODULARIZAÇÃO (FUNÇÕES)

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

```
exibir_mensagem_de_boas_vindas() {  
    echo "Bem-vindo ao meu script!"  
}
```

```
exibir_data_e_hora_atual() {  
    echo "A data e hora atual são: $(date)"  
}
```

```
main() {  
    exibir_mensagem_de_boas_vindas  
    exibir_data_e_hora_atual  
}
```

```
main
```

# MODULARIZAÇÃO (FUNÇÕES)

*funcoes.sh*

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

```
exibir_mensagem_de_boas_vindas() {  
    echo "Bem-vindo ao meu script!"  
}
```

```
exibir_data_e_hora_atual() {  
    echo "A data e hora atual são: $(date)"  
}
```



# MODULARIZAÇÃO (FUNÇÕES)

*main.sh*

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

```
source funcoes.sh
```

```
function main() {  
    exibir_mensagem_de_boas_vindas  
    exibir_data_e_hora_atual  
}
```

```
main
```

# APLICAÇÕES

- ❑ Automação de tarefas repetitivas
- ❑ Scripting de sistema
- ❑ Programação em Shell
- ❑ Administração de servidores



# APLICAÇÕES

```
GNU nano 6.2                                atsistema.sh
#!/bin/bash
TIME=1
clear
while true;do
echo " "
echo "SEJA BEM VINDO AO $0 DO UBUNTU!"
echo " "
echo "Escolha uma opção abaixo para começar!"
    1 - Verificar repositório do sistema
    2 - Mostrar atualizações do sistema
    3 - Instalar atualizações do sistema
    4 - Limpar o sistema
    5 - Remover pacotes não necessários
    0 - Sair do sistema"
echo " "
echo -n "Opção escolhida: "
```

# APLICAÇÕES

```
GNU nano 6.2          atsistema.sh
read opcao
case $opcao in
    1)
        echo Verificando o sistema por atualizações...
        sleep $TIME
        apt update
        ;;
    2)
        echo Mostrando as atualizações do sistema...
        sleep $TIME
        apt list --upgradable
        ;;
    3)
        echo Instalando atualizações do sistema...
        sleep $TIME
        apt upgrade -y
        ;;
```

# APLICAÇÕES

```
GNU nano 6.2                               atsistema.sh
4)
    echo Limpando o sistema...
    apt clean
    apt autoclean
    ;;
5)
    echo Removendo pacotes desnecessários do sistema...
    sleep $TIME
    apt autoremove -y
    ;;
0)
    echo Saindo do sistema...
    sleep $TIME
    exit 0
    ;;
*)
    echo Opção inválida, tente novamente!
    ;;
esac
done
```