



Capítulo 1: Introdução ao BASH

Type	Leitura
Materials	Capítulo 1.pdf
Reviewed	<input type="checkbox"/>

Introdução ao Bash

Aviso de Comando (Prompt)

- `SHIFT` e pressionando `PGUP` OU `PGDOWN` : Isso é útil para ver textos que rolaram rapidamente para cima.
- `du -hs` : Mostrará o tamanho total (pode usar o `-a` (espaço ocupado por todos os arquivos))
- `bash - version`

Tecclas Úteis

- `Back Space ("--")` : Apaga um caractere à esquerda do cursor.
- `Del` : Apaga o caractere acima do cursor.
- `CTRL+A` : Move o cursor para o início da linha de comandos.
- `CTRL+E` : Move o cursor para o fim da linha de comandos.
- `CTRL+U` : Apaga o que estiver à esquerda do cursor. O conteúdo apagado é copiado para uso com `CTRL+y`.
- `CTRL+K` : Apaga o que estiver à direita do cursor. O conteúdo apagado é copiado para uso com `CTRL+y`.
- `CTRL+L` : Limpa a tela e mantém o texto que estiver sendo digitado na linha de comando (parecido com o comando `clear`).

- **CTRL+Y:** Coloca o texto que foi apagado na posição atual do cursor.

Comandos

- **comando [opções] [argumentos]**
 - [opções]: Modifica o comportamento de um comando.
 - [argumento]: Especifica algo para o qual o comando agirá (como nome de arquivo ou nome de usuário).
 - Exemplo: `ls -lha`

Padrão de Hierarquia do Sistema de Arquivos (FHS)

1. **/bin:** Binários do sistema (*TUDO* que é executável para o Linux).
Ex: `ls`, `vim`, `login`, `exit`, `cd`, `users`.
2. **/boot:** Inicialização do sistema, onde tem o kernel linux etc.
Ex: `grub` (se é linux ou windows), `vmlinuz` (kernel do linux, que varia de 5,1M).
3. **/run:** Tudo que está rodando desde o último boot. É um arquivo temporário.
Exemplo: `network`, `daemons`, `users`.
4. **/media:** Mídia removível que já está montada automaticamente.
5. **/dev:** Tudo que é dispositivo. Tudo no linux é um arquivo, inclusive um HD, SATA (`sda5`), etc.
Ex: `sda` (primeiro disco), `sdb` (segundo disco), `sda5` (onde pode dividir em partições). Temos 2 tipos de dispositivos: Temos dispositivos de bloco (armazém. - HD, PENDRIVE) e caractere (transferindo informação - Primeira porta serial, terminais de texto, um dispositivo que descarta todos os dados escritos nele, dispositivos Random, mouse).
 - Exemplo: `brw-rw----` (bloco), `crw-rw----` (caractere).
6. **/share:** Documentação/ manual/help.
7. **/etc:** Todas as configurações da máquina ou arquivos de configuração.
Ex: nome da máquina (`hostname`), parte de rede para configuração, `log motd`, `samba`, `nginx`, `apache`, etc.
8. **/home:** Contém o diretório de cada usuário e as coisas de cada um.

Ex: /home/john/ (pasta pessoal do usuário "John").

9. **/lib:** Bibliotecas dinâmicas compartilhadas entre os programas.

Ex: /lib/libc.so.6 (biblioteca C compartilhada).

10. **/lost+found:** Usado para armazenar arquivos ou diretórios recuperados pelo utilitário fsck em caso de problemas no sistema de arquivos.

Ex: /lost+found/recovered_file.txt (arquivo recuperado), libdiscover.so.2.0.1, módulos, drivers (módulos de kernel com algumas extensões .ko).

11. **/mnt:** Ponto de montagem temporário para dispositivos e sistemas de arquivos adicionais.

Ex: /mnt/external_hd/ (disco rígido externo montado).

12. **/opt:** Ponto onde instalei o SO e arquivos de terceiros que não são do linux. Pouco utilizado, porque existe gerenciamento de pacotes (como apt, yum, dnf, etc.) que facilitam a instalação e a atualização de software.

Ex: /opt/google/chrome/ (diretório de instalação do Google Chrome), Aplicativos de terceiros com requisitos específicos de instalação. Software personalizado ou de desenvolvedor que não faz parte dos repositórios de pacotes da distribuição.

13. **/proc:** Um sistema de arquivos virtual que fornece informações sobre o estado do kernel e processos em execução, informações dinâmicas sobre o sistema. Todo processo que está executando está no /proc.

Ex: cpuinfo, meminfo(total de memória), uptime.

14. **/root:** O diretório pessoal do usuário "root" (superusuário) do sistema.

Ex: /root/ (pasta pessoal do superusuário).

15. **/sbin:** Contém programas usados pelo superusuário (root) para administração e controle do sistema.

Ex: add usuário, deletar, modificar, swapon e off, shutdown, runuser.

16. **/sys:** Similar ao /proc, é um sistema de arquivos virtual usado para interagir com configurações e informações do kernel.

Ex: drivers, firmware, hypervisor, kernel, módulos, power.

17. **/tmp:** Arquivos temporários pelas aplicações etc. Quando reinicia, ele apaga.

Ex: sessões do usuário, impressão temporária, downloads, arquivos de troca (swap), compressão e descompressão, cache de aplicativos.

18. **/usr**: Maior diretório do Linux depois do raiz, tem os libs, sbin, bin, mas que não são essenciais para o sistema.

Ex: games, include, libs, share (documentação, comp dados etc.), src (código fonte dos programas), local (a 3ª hierarquia do Linux).

19. **/var**: Armazena arquivos que são gravados frequentemente pelos programas do sistema, como logs, emails, cache, etc.

Ex: /var/log/syslog (log do sistema).

20. **/srv**: Dados que não são dinâmicos → pouco utilizado.

Ex: páginas do http apache, serviços ftp, repositórios git.

Desligar

- `halt` : Desligar.
- `sudo init 0` : Desliga a máquina.
- `poweroff` : Desliga.
- `shutdown -h now` : Desligar agora.
- `reboot` : Reiniciar.
- `init 6` : Reinicia.
- `shutdown -h 13:00`.
- `shutdown -h +30` "O sistema será desligado para manutenção às 23:00. Por favor, salve seu trabalho."
- `shutdown -c` "Já resolvi. Não sou reiniciar."

Comandos Internos

- Comandos são processados e executados pelo próprio interpretador de comandos (shell) sem invocar um programa externo.
 - *ex: pwd, cd, type, exit, echo, alias*

- **echo**: O comando echo é usado para exibir texto na tela. Ex: `echo "Olá, mundo!"`
- **alias**: O comando alias permite criar atalhos para comandos longos. Ex: `alias nome_do_alias='comando'`

COMANDOS INTERNOS

- Digite o **"type cd"** para saber se é interno ou externo

'echo': O comando echo é usado para exibir texto na tela.

#exemplo: `echo "Olá, mundo!"`

``history``: O comando histor
y mostra o histórico de comandos que você digitou anteriormen

#exemplo: ``history | grep -E "^ *[0-9]+ (ls|cd|mkdir)"``
`#`history | grep "$(date --date='3 days ago' '+%Y-%m`

``history !3`` → Pegar um comando já digitado em específico

`history 3` → apenas os 3 últimos

`histsize=500` → declarar tamanho para o histórico

`!-3` → 3 comando de baixo pra cima

`!!` → último comando

`cal 5 2030` → calendário

`ctrl + r` → para pesquisar caso não queira pelo history

exit ou logout: O comando exit (ou logout) é usado para sair

source ou .: O comando source (ou .) é usado para executar co

ex: source meu_script.sh

Use o comando bg para retomar a execução do processo em segun

HISTTIMEFORMAT='%F %T' history

#se voce quiser que permanentemente, coloque export HISTTIMEF

COMANDOS EXTERNOS

- programas ou scripts que estão localizados fora do shell e não fazem parte dele. Esses comandos são executáveis independentes que o shell chama para realizar uma tarefa específica
 - **ex:** path, ls, grep, cp, mv, vim, tar

PATH

export **PATH**="/novo:\$PATH"

echo \$**PATH** (para ver se funcionou)

Diretórios:

- Local usado para organizar e armazenar conjuntos de arquivos e subdiretórios
- Melhorar a organização e a localização desses dados. "Em essência, é uma pasta que pode conter arquivos e outras pastas."
- **"case sensitivity"** (sensibilidade a maiúsculas e minúsculas): O sistema operacional trata letras maiúsculas e minúsculas de maneira distinta. Em

outras palavras, ele diferencia entre letras maiúsculas e minúsculas nos nomes de arquivos e diretórios.

- **Não podem existir dois arquivos com o mesmo nome em um mesmo diretório.** No entanto, você pode ter arquivos com o mesmo nome em diretórios diferentes, desde que esses diretórios estejam em caminhos diferentes, pois o caminho completo para um arquivo é único.

```
`Por exemplo:`  
  
- `/home/usuario/arquivo.txt`  
- `/var/arquivo.txt`  
  
`seriam considerados diferentes:`  
  
- `/teste`  
- `/Teste`
```

1. Caminho Absoluto:

- Um caminho absoluto especifica o local de um arquivo ou diretório em relação à raiz do sistema de arquivos.
- Começa com uma barra (/) e inclui todos os diretórios necessários para alcançar o arquivo ou diretório desejado a partir da raiz.
- Exemplos de caminhos absolutos:
 - `/home/usuario/arquivo.txt`
 - `/var/log/syslog`
 - `/usr/bin/gcc`

2. Caminho Relativo:

- Um caminho relativo especifica o local de um arquivo ou diretório em relação ao diretório atual (ou de trabalho) em que você está.

- O diretório atual pode ser representado então por um ponto e uma barra em sistemas windows `(.\)` e inclui um diretório ou subdiretórios.

Exemplo

1. Para você entender a hierarquia dos diretórios, vamos apresentar o caminho absoluto do arquivo, que é `C:\Users\Camila\Documents\Documento.tx` ;
2. Eu estou no diretório `C:\Users\Camila>` e quero acessar o arquivo `Documento.txt` . Como faço isso com caminho relativo?
3. Nessas condições, o nome do caminho relativo para o arquivo ficará `.\Documents\Documento.txt` .

MUDANDO HORA E DIA

```
sudo date -s "2024-08-02 00:15:00"
```

MANIPULAÇÃO DE DIRETÓRIOS I

1. `-l` : exibe uma **lista longa** que fornece informações detalhadas sobre os arquivos, incluindo permissões, proprietário, grupo, tamanho e data de modificação.
2. `-L` : Exibe link simbólico
3. `-a` **Mostra arquivos ocultos**
`-A` → Lista todos os arquivos e diretórios, exceto os diretórios especiais "." (atual) e ".." (pai) e seus arquivos.
4. `ls -b` → **ver caracteres especiais em nomes de arquivo.**
5. `h, --human-readable` : Faz com que o ls mostre tamanhos de arquivo **legíveis para humanos**, como "1K", "234M" ou "2G".

6. `i, --inode`: Exibe o número do **índice (inode) de cada arquivo**. Os inodes são números únicos associados a cada arquivo no sistema de arquivos.
7. `t`: Classifica os **arquivos por data de modificação**, mostrando os mais recentes primeiro.
8. `r, --reverse`: Inverte a ordem de classificação, mostrando os arquivos do **mais antigo para o mais novo**.
9. `R, --recursive`: Lista subdiretórios de **forma recursiva**, ou seja, **exibe o conteúdo de subdiretórios também**.
10. `F, --classify`: Adiciona indicadores aos nomes dos arquivos para **indicar o tipo de arquivo (por exemplo, "/" para diretórios)**.
11. `-color[=WHEN]`: Ativa a colorização da saída do ls para melhorar a legibilidade. "WHEN" pode ser "always" (sempre), "auto" (automático) ou "never" (nunca).
12. `G, --no-group`: **Não exibe o nome do grupo dos arquivos** na lista longa.
13. `d, --directory`: **Lista apenas os diretórios, não seu conteúdo**.
14. `s, --size`: **uso de espaço em disco de cada arquivo**, mostrando o número de blocos ocupados.
15. `S, --size`: **Exibe do maior para o menor**.
16. `q, --hide-control-chars`: **Substitui caracteres não gráficos por "?" na saída**.
17. `L, --dereference`: **Mostra informações do arquivo referenciado por links simbólicos**, em vez do link em si.
18. `c`: **Classifica os arquivos com base na data de mudança (ctime)**, em vez da data de modificação.
19. `G, --no-group`: **Não exibe o nome do grupo dos arquivos** na lista longa.
20. `-n` **exibe seus IDs numéricos (UID e GID)**.
21. `-x` **listagem de arquivos por extensão, em ordem alfabética**

- `ls /home/joao como ls` ~ para listar os arquivos de seu diretório home.
- `ls ../../var/log`

- `ls -lS` maior para o menor
- `ls -lrS` menor para o maior (mais antigo)
- `ls -lt` /etc data de criação
- `ls -R /etc` lista recursivamente
- `ls -lha` → arquivos e diretórios com detalhado
- `ls -lh` → lista com os KB
- `ls -a` → todos os arquivos ocultos e diretórios especiais
- `ls -A` → Lista todos os arquivos e diretórios, exceto os diretórios especiais "." (atual) e ".." (pai) e seus arquivos.
- `ls -lha | more (ou less)`
- `ls -b` → **ver caracteres especiais em nomes de arquivo.**
- `cd ../../..`
- `ls -l` → **Informações detalhadas dos diretórios etc.**
- `ls -l -f (ou -p)` → Informações com ordem de criação
- `ls -l -G (ou -o)` → Vai exibir só a coluna do arquivo
- `ls -ln` → converte os nomes do proprietário e do grupo para seus respectivos IDs de usuário (UID) e IDs de grupo (GID) numéricos
- `ls -l -L` → exibirá informações sobre o destino real de um link simbólico, em vez de exibir informações sobre o próprio link.
- `ls -t` → ordena os arquivos por data recente
- `ls -latr` → ordena os arquivos por data antiga
- `ls -lac` → ordem de criação
- `ls -laX (-lX, -lXr)` → arquivos em ordem alfabética por extensão. Isso é útil quando você deseja agrupar arquivos com a mesma extensão juntos na lista.
- `ls -laR` → lista os arquivos e diretórios em todos os níveis da hierarquia de diretórios.

Sintaxe para Criar um Link Simbólico

```
ln -s caminho_do_destino caminho_do_link
```

- `caminho_do_destino` : o arquivo ou diretório que será referenciado.
- `caminho_do_link` : o nome do link simbólico que será criado.

MANIPULAÇÃO DE DIRETÓRIOS II

acessar

- `cd` → diretório ao qual deseja acessar
- `cd -` → **volta para o diretório anterior**
- `cd ../../` → desce 2 pastas
- `cd ../teste2` → **sobe um diretório**
- `cd ~/paulo`
- `cd ~ (ou /)` → para ir para o diretório home
- `pwd` → diretório atual

Criar Diretórios

1. `mkdir [opções] [caminho/diretório] [caminho1/diretório1]`
2. `mkdir /diretorio{subdiretorio1, subdiretorio2, subdiretorio3}`

- `mkdir` → cria diretórios
- `mkdir -p ~/nada/Diretorio1/Diretorio1_2/Diretorio1_3` → *cria uma sequência grande de subdiretórios dentro do diretório inicial (Diretório 1)*

- `ex: mkdir -p ~/backup/diretorio01/diretorio01_1/diretorio01_2
~/backup/diretorio02/diretorio02_1/diretorio02_2`

Remover Diretórios

- `rmdir` → remove DIRETÓRIOS VAZIOS
- `rm ~/nada/diretorio01/pokemon.txt` → remove o arquivo "pokemon.txt"
- `rmdir ~/nada/diretorio01 ~/nada/diretorio02/diretorio02_2 ~/nada/diretorio02` → remove os diretórios e subdiretórios
- `tree -A` → ver a árvore de diretórios criada

visualizar conteúdo

- `cat` → visualizar o conteúdo de um arquivo
- `cat -n teste` → número da linha, inclusive as linhas em branco
- `cat -s teste` → ocultar linhas em branco repetidas
- `cat -b teste` → enumera apenas as linhas que contêm conteúdo
- `cat -E teste` → adiciona um '\$' nos espaços
- `cat -T teste` → converte o tab (espaço) em '^'
- `zcat teste.gz` → para ver o arquivo sem precisar descompactar
- `bzcat teste` → visualizar com extensão bz
- `tac teste` → imprime as linhas de um arquivo da última para a primeira

removendo arquivos e diretórios

- `rm` → remove arquivos ou diretórios
- `rm -r Diretorio1` → melhor do que o `rmdir`, porque ele solicita a confirmação para cada subdiretório
- `rm -rf Diretorio17/` → remover todo o diretório (subdiretório etc.)

- `rm -i` → o sistema solicitará uma confirmação antes de efetivamente excluir o arquivo
- `rm -rf *` → remove tudo
- `rm -rf a*` → todos que começam com a letra 'a' são removidos
- `rm -- -` → remove arquivos que tem o '-'

exemplo:

Remova o arquivo teste2/arquivo1.txt. Anote o comando
`rm ~/exercicio1/teste2/arquivo1.txt`

Renomeie o arquivo teste1/arquivo2.txt para teste1/arquivo_re
 comando.

`mv ~/exercicio1/teste1/arquivo2.txt ~/exercicio1/teste1/arqui`

Remova o diretório teste3, com todos os seus arquivos, utiliz
 Anote o comando.

`rm -r ~/exercicio1/teste3`



copiar

- `cp [origem] [destino]` → copia de um lugar pro outro
- `cp ~/exercicio1/teste2/arquivo1.txt ~/exercicio1/teste1/arquivo2.txt` → move para outro lugar e mudando o nome do arquivo
- `cp -a ~/origem ~/destino` → fazendo uma **cópia completa**, incluindo subdiretórios e mantendo todas as informações dos arquivos e diretórios originais.
- `cp -r *Pasta Pasta2` → **copia todos os arquivos** da Pasta para Pasta 2
- `cp -p ~/origem ~/destino` → **preservação de arquivo de informações** como a data de modificação, as permissões de arquivo e o dono do arquivo. o arquivo "copia_do_arquivo.txt" terá as mesmas permissões e timestamps (data de modificação e acesso) que o arquivo original.
- `cp -v ~/origem ~/destino` → exibirá na tela **informações sobre os arquivos sendo copiados**
- `cp -u ~/origem ~/destino` → a cópia será criada apenas se o arquivo de origem for mais recente do que a cópia existente no destino.
- `cp -uv`
- `cp documentos/* backup` → copia todos os arquivos de documentos para backup
- `cp */tmp` → **copia todos os arquivos do diretório atual** para o /tmp
- `cp -vrx * diretorio/` → não copiar arquivos dentro de pastas que já estão em outros sistemas de arquivos
- `cp -rp diretorio_original/ copia_do_diretorio/` → o diretório "copia_do_diretorio" conterá todos os arquivos e subdiretórios do diretório original, e **todas as permissões e timestamps serão mantidas**.

exemplo:

Copie o arquivo1.txt para o diretório teste1, com o nome arqu.

```
cp ~/exercicio1/teste2/arquivo1.txt ~/exercicio1/teste1/arqui
```

Copie os arquivos arquivo1.txt e arquivo2.txt para o diretório nome. Anote o comando.

```
cp ~/exercicio1/teste1/arquivo2.txt ~/exercicio1/teste3/ && c
```



mover

- Apaga o arquivo de origem , e ai ele move
 - `mv` → move da origem para o destino, mas a origem é apagada
 - `touch nome_do_arquivo && mv nome_do_arquivo /caminho/para/a/pasta/`
 - `mv -p arquivo.txt /caminho/do/novo/diretorio`
 - `mv -f arquivo.txt /caminho/do/novo/diretorio/`
 - `mv -r diretorio_origem/ /caminho/do/novo/diretorio/`

X11

- O X11, também conhecido como "X Window System," é uma parte do Linux que lida com a parte gráfica. É responsável por mostrar janelas, ícones, botões e tudo que você vê na tela quando usa o computador.

Aqui estão algumas coisas importantes que o X11 faz:

Criar e Listar Diretório:

Para criar um diretório, use o comando mkdir (abreviação de "make directory"):

- `mkdir nome_do_diretorio`
- `mkdir Pastas_Testes/Pastas_Testes2`
- `ls -d Pasta_Testes` → Listar a pasta e não o conteúdo
- `ls Pasta1 Pasta2`

Criar Arquivo:

- Para criar um diretório, use o comando mkdir (abreviação de "make directory"):

```
touch local/nome_do_arquivo
```

ex: touch Pastas_Testes/arquivo.txt

Renomear Diretório ou Arquivo:

- Para renomear um diretório ou arquivo, use o comando mv (abreviação de "move"):

```
mv nome_atual novo_nome
```

Copiar Diretório ou Arquivo:

```
cp nome_origem nome_destino
```

Excluir diretório ou arquivo:

- Para excluir um arquivo: `rm nome_do_arquivo`
- Para excluir um diretório e seu conteúdo: `rm -r nome_do_diretorio`

Renomear em Lote (Batch Rename):

```
rename 's/padrao/novo_nome/' arquivos
```

- `rename 's/.txt$/bak/' *.txt`

Visualizar conteúdo de Arquivo:

- Para visualizar o conteúdo de um arquivo, use o comando

```
cat
```

```
less
```

- `(/nome_que_quero_procurar)`: faz uma busca no texto
- `n` : para avançar (next)
- `shift + n` (para voltar no text)

```
more (b para voltar e f para avançar)
```

```
head & tail
```

- para ver o início ou final do texto
 - `tail -n` : para definir o numero de linhas

VARIÁVEL DE AMBIENTE

- Para alterar e remover uma variável de ambiente
- Obs: Para ver as variáveis de ambiente, basta dar um `"env"`

VARIAVEL COM ASPAS E SEM ASPAS

```
echo $greeting mostra o conteudo sem aspas
echo "$greeting" = mostra o conteudo com aspas
echo '$greeting' = mostra o '$greeting'
```

Editar Arquivo:

- Para visualizar o conteúdo de um arquivo, use o comando **cat**, **less**, **more**, ou **nano**, dependendo de suas preferências e necessidades:

```
nano nome_do_arquivo
```

Vim:

- **Abrir um arquivo:** `vim nome_do_arquivo`
- **Modo de inserção:** Pressione `"i"` para entrar no modo de inserção, onde você pode digitar e editar o texto.
- **Salvar alterações:** No modo de inserção, pressione `"Esc"` para sair do modo de inserção e, em seguida, digite `:w` e pressione `"Enter"` para salvar o arquivo.
- **Colocar um nome para salvar** `':saveas meu_novo_arquivo.txt'`
- **Sair do Vim:** No modo de inserção, pressione `"Esc"` para sair do modo de inserção e, em seguida, digite `:q` e pressione `"Enter"` para sair do Vim. Se você fez alterações no arquivo, você pode usar `:q!` para sair sem salvar.
- **Salvar e sair:** Para salvar e sair ao mesmo tempo, digite `:wq` e pressione `"Enter."`

MCEDIT (Editor do Midnight Commander):

- **Abrir um arquivo com MCEDIT:** Pressione `"F4"` enquanto estiver no Midnight Commander para editar um arquivo no MCEDIT.
- **Modo de edição:** Você pode editar o texto diretamente no MCEDIT.

- **Salvar alterações:** No MCEDIT, pressione `"F2"` para salvar as alterações.
- **Sair do MCEDIT:** Para sair do MCEDIT, pressione `"F10."`

Nano:

- **Abrir um arquivo:** `nano nome_do_arquivo`
- **Modo de edição:** O Nano é bastante intuitivo e permite que você edite o texto diretamente na interface.
- **Salvar alterações:** Pressione `"Ctrl" + "O"` (letra "O" maiúscula) para salvar o arquivo, e depois pressione `"Enter."`
- **Sair do Nano:** Pressione `"Ctrl" + "X"` para sair do Nano. Se você fez alterações, o Nano perguntará se você deseja salvá-las antes de sair.

PERMISSÕES ATRIBUÍDAS AO ARQUIVO/COMANDO PARA O PROPRIETÁRIO, GRUPO E OUTROS

```
rwxr-xr-x 1 root root 527376 Jan 24 2022 /usr/bin/wget
```

Para o proprietário (dono) do arquivo (que neste caso é root):

Leitura (r)

Escrita (w)

Execução (x)

Para o grupo ao qual o arquivo pertence (que também é root neste caso):

Leitura (r)

Execução (x)

Para outros usuários (qualquer outro usuário no sistema):

Execução (x)

FUNÇÃO (SHELL SCRIPT)

```
mostrar_informacoes() {  
    > # Limpar a tela  
    > clear  
  
    > # Imprimir a data atual  
    > echo "Data atual: $(date)"  
  
    > # Imprimir o hostname  
    > echo "Hostname: $(hostname)"  
  
    > # Imprimir o historico  
    > echo -e "\nHistórico de comandos executados:"  
    > history
```