



escola  
britânica de  
artes criativas  
& tecnologia

# Front-End



# Comandos para Mac

- Comandos básicos do terminal
- Instalando GIT



# Comandos básicos do terminal



# Comandos básicos do terminal

Abra o terminal do MAC, abra o finder, vá em utilitários e depois clique em Terminal.



## ls

Permite a visualização de arquivos do diretório, sendo pastas ou arquivos que estejam presentes no diretório onde o comando for aplicado.

```

havokk — zsh — 80x24
Last login: Wed May 10 05:16:11 on ttys000
havokk@MacBook-Pro-de-Havokk ~ % ls
Desktop      Downloads    Movies       Pictures     teste
Documents    Library     Music        Public
havokk@MacBook-Pro-de-Havokk ~ %
  
```



# Comandos básicos do terminal

## • pwd

Apresenta no terminal o caminho do sistema onde estamos localizados nesse momento.

```

havokk@MacBook-Pro-de-Havokk ~ % pwd
/Users/havokk
havokk@MacBook-Pro-de-Havokk ~ % █
  
```

## • mkdir

Cria uma pasta diretório (lembrando que será criada a pasta no diretório onde estaremos neste momento).

```

havokk@MacBook-Pro-de-Havokk ~ % mkdir projetos
havokk@MacBook-Pro-de-Havokk ~ % ls
Desktop      Downloads    Movies       Pictures      projetos
Documents    Library      Music        Public        teste
havokk@MacBook-Pro-de-Havokk ~ % █
  
```



# Comandos básicos do terminal

## • cd

Permite a navegação entre pastas (detalhe: temos formas de utilizar o CD, veja a seguir).

## • cd nomedodiretorio

Entra no diretório com o nome que você estiver colocando em “nomedodiretorio”. Nota, ele já deve existir previamente para que o comando funcione.

```

havokk@MacBook-Pro-de-Havokk ~ % cd projetos
havokk@MacBook-Pro-de-Havokk projetos % ls
havokk@MacBook-Pro-de-Havokk projetos %
  
```

**Nota:** o ls retornou nada, pois a pasta havia sido criada recentemente e está vazia.



# Comandos básicos do terminal

## • **cd ..**

Colocando dois pontos finais após o **cd**, ele irá retornar uma pasta anterior. Ou seja, em um caso que estivermos na pasta Desktop e usarmos **cd ..**, iremos retornar para a pasta anterior, a do sistema!

```

[havokk@MacBook-Pro-de-Havokk projetos % cd ..
[havokk@MacBook-Pro-de-Havokk ~ % ls
Desktop      Downloads    Movies       Pictures     projetos
Documents    Library      Music        Public      teste
havokk@MacBook-Pro-de-Havokk ~ %
  
```



# Comandos básicos do terminal

## • **cd -**

Ao usar esse comando, iremos retornar ao diretório anterior ao que estávamos ao usar o comando `cd`. Por exemplo: Entramos em uma pasta onde não queríamos estar, mas esquecemos que pasta estávamos antes.

```

[havokk@MacBook-Pro-de-Havokk ~ % cd -
~/projetos
havokk@MacBook-Pro-de-Havokk projetos %
  
```





# Comandos básicos do terminal

## • touch

Comando utilizado para criar arquivos. Basta usar touch nomedoarquivo.extensão para que possamos gerar um arquivo no sistema.

Exemplo: touch nome.txt

```

havokk@MacBook-Pro-de-Havokk projetos % touch nome.txt
havokk@MacBook-Pro-de-Havokk projetos % ls
nome.txt
havokk@MacBook-Pro-de-Havokk projetos %
  
```



# Comandos básicos do terminal

## • Clear

Limpa o histórico apresentado no terminal.



# Comandos básicos do terminal

## ● **nano nomedoarquivo.extensão**

Abrimos o arquivo, principalmente de texto ou parâmetros de configurações, como CSS ou .ini, para editar seu conteúdo com o editor de texto chamado NANO. Com ele, teremos opções extras para a edição do arquivo.

```

[havokk@MacBook-Pro-de-Havokk projetos % ls
nome.txt
[havokk@MacBook-Pro-de-Havokk projetos % nano nome.txt
[havokk@MacBook-Pro-de-Havokk projetos %
  
```



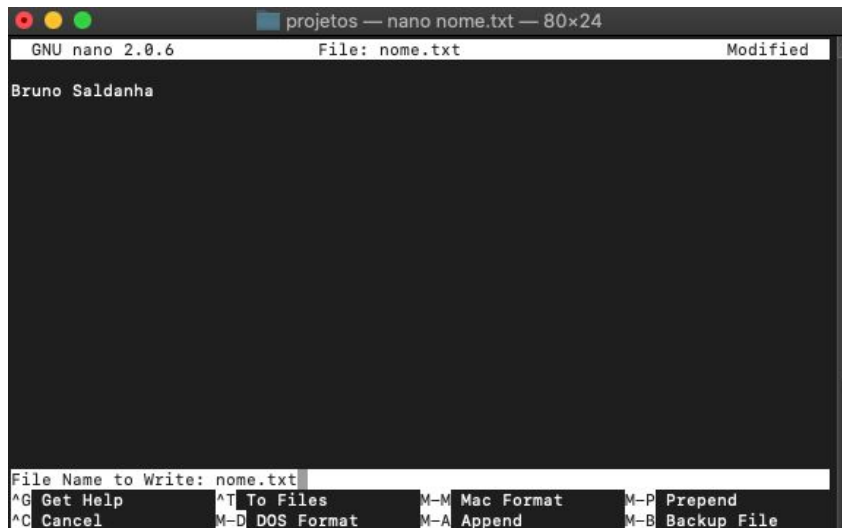
**Nota:** temos comandos abaixo. Utilizando a tecla CTRL + alguma tecla, teremos acessos a novos comandos. Veja os principais a seguir.



# Comandos básicos do terminal

## • CTRL + O

Salvará o conteúdo editado, perguntando o nome do arquivo que deseja fazer a edição.



```

projetos — nano nome.txt — 80x24
GNU nano 2.0.6      File: nome.txt      Modified
Bruno Saldanha


File Name to Write: nome.txt
^G Get Help      ^T To Files      M-M Mac Format    M-P Prepend
^C Cancel        M-D DOS Format    M-A Append        M-B Backup File
  
```



# Comandos básicos do terminal

Aperte ENTER e será salvo no arquivo que estamos editando.

Com isso, o editor retorna para as opções anteriores, bastando apesar usar o comando **CTRL + X**, que sairá do NANO.




# Comandos básicos do terminal

## • Cat

Apresenta o conteúdo daquele arquivo para a gente sem a necessidade de abrir ou editar o mesmo.

```

[havokk@MacBook-Pro-de-Havokk projetos % cat nome.txt
Bruno Saldanha
havokk@MacBook-Pro-de-Havokk projetos %
  
```



# Instalando GIT

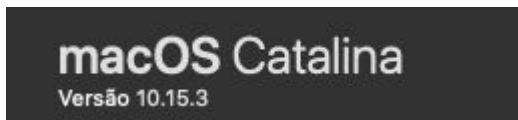


# Instalando GIT

Aqui, temos de checar a versão do seu sistema operacional para termos a certeza de qual gerenciador de pacotes ter em mãos.

Clique no ícone da maçã no canto esquerdo superior e clique em “Sobre este Mac”. Uma janela aparecerá, mostrando a versão do seu Mac.

Neste caso, é esse:



Tendo essa informação em mãos, iremos decidir qual gerenciador de pacotes baixar.

O git fornece duas opções: O Macports (Para sistemas abaixo do 11, essa é a opção) e o Homebrew (opção para versões 11 para cima).





# Instalando GIT

• Homebrew

<https://brew.sh/>

• MacPorts

<https://www.macports.org/install.php>

**Nota:** o Homebrew será instalado via Terminal, enquanto o MacPorts será um executável que será baixado pelo site e instalado como um pacote para o sistema.



# Instalando GIT

Tendo quaisquer um dos dois instalados, basta usar o comando abaixo para baixar o Git:

 **brew install git**  
Para Homebrew

 **sudo port install git**  
Para MacPorts



# Instalando GIT

Pode aparecer um erro no terminal e pedir a instalação do xcodebuild:



Caso isso ocorra, instale-o normalmente. Se seu Mac for uma versão mais antiga, deve procurar a versão do Xcode equivalente para realizar o download.



# Instalando GIT

- Lista de suporte por versão

<https://developer.apple.com/pt/support/xcode/>

- Lista de download de versões mais antigas

<https://developer.apple.com/download/all/?q=xcode>



# Instalando GIT

Com o git instalado, é imprescindível que seja feito o seguinte comando:

• **git config --global core.editor "nano"**

Esse comando permite que o editor NANO, utilizado nas aulas, possa ser usado normalmente nos comandos git também!



# Bons estudos!

