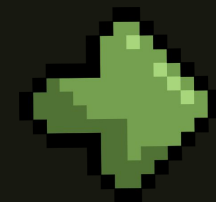


Conhecendo o computador e o terminal

Leonardo de Souza

O que vamos ver hoje



Um pouco de história



Analogia da cozinha e cenário atual



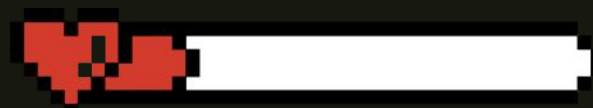
Terminal e comandos



Pare!

Antes de qualquer coisa,
vamos conhecer a turma!





A incrível história dos computadores

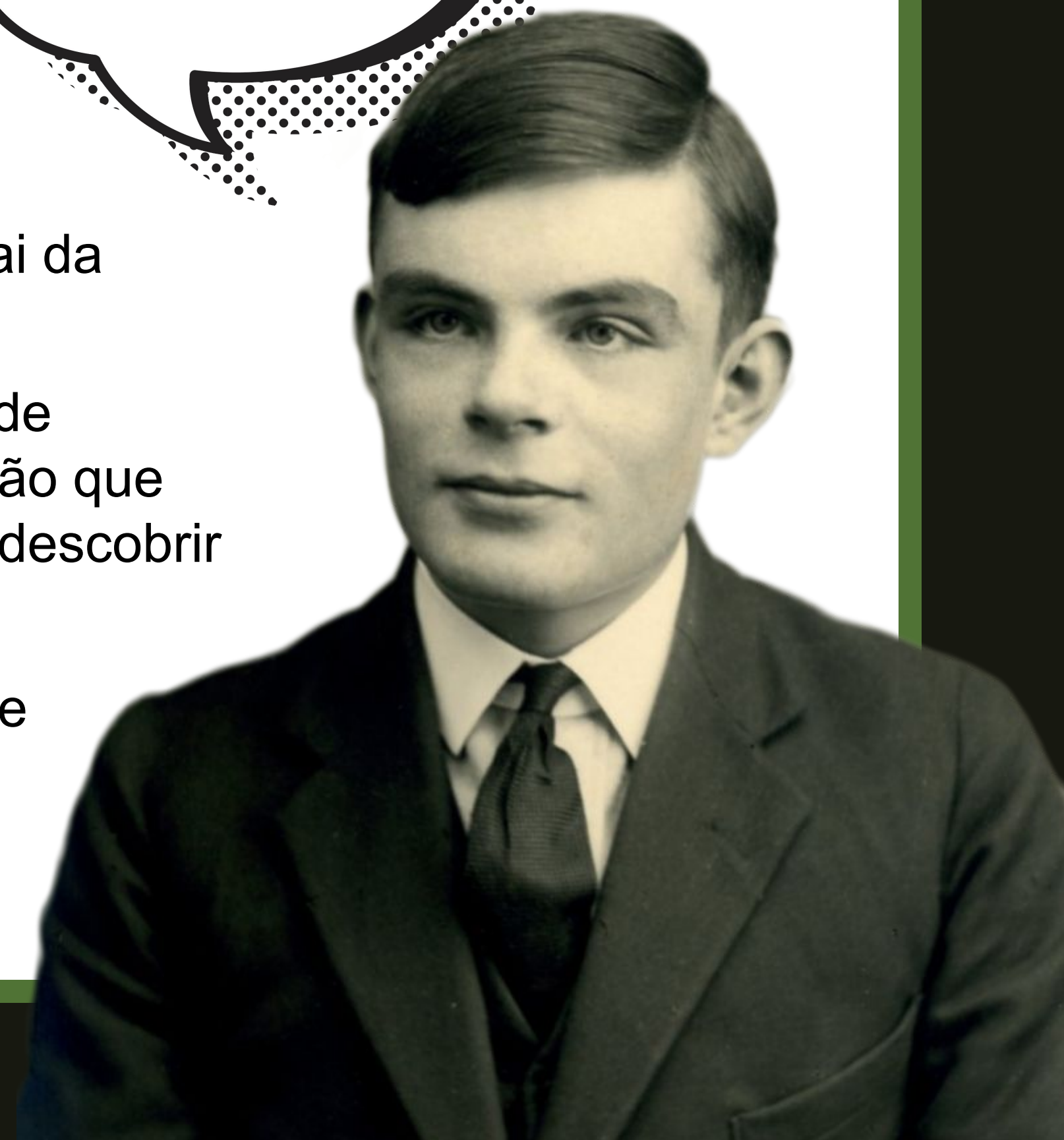
Oi!

Alan Turing

Quem foi? Um matemático inglês considerado o pai da ciência da computação.

O que ele fez? Ele desenvolveu o que chamamos de “Máquina de Turing”, que fundamentou a computação que conhecemos hoje. Ele também foi responsável por descobrir os códigos de guerra alemães na Segunda Guerra.

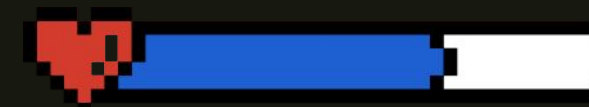
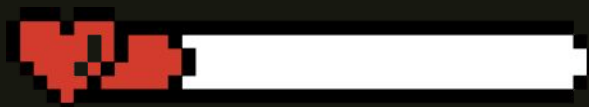
Se quiser saber mais eu recomendo assistir o filme “O Jogo da Imitação”.



Máquina de Turing

O que é? É uma ideia (ou seja, não existe fisicamente) que ajuda a entender como os computadores funcionam. É um computador teórico, composta por uma fita que contém informações, um dispositivo que lê e escreve nessa fita e uma tabela de estados e instruções, que especificam como ela deve se comportar com base no símbolo atual na fita e no estado atual.

O que ela faz? Essa teoria é usada para analisar problemas computacionais e demonstrar o que é e o que não é computacionalmente possível. **Lembrando: não existe uma Máquina de Turing real, é apenas uma ideia.** Na prática, os computadores modernos são muito mais complexos e têm muitos componentes além dos elementos básicos do modelo da máquina de Turing.

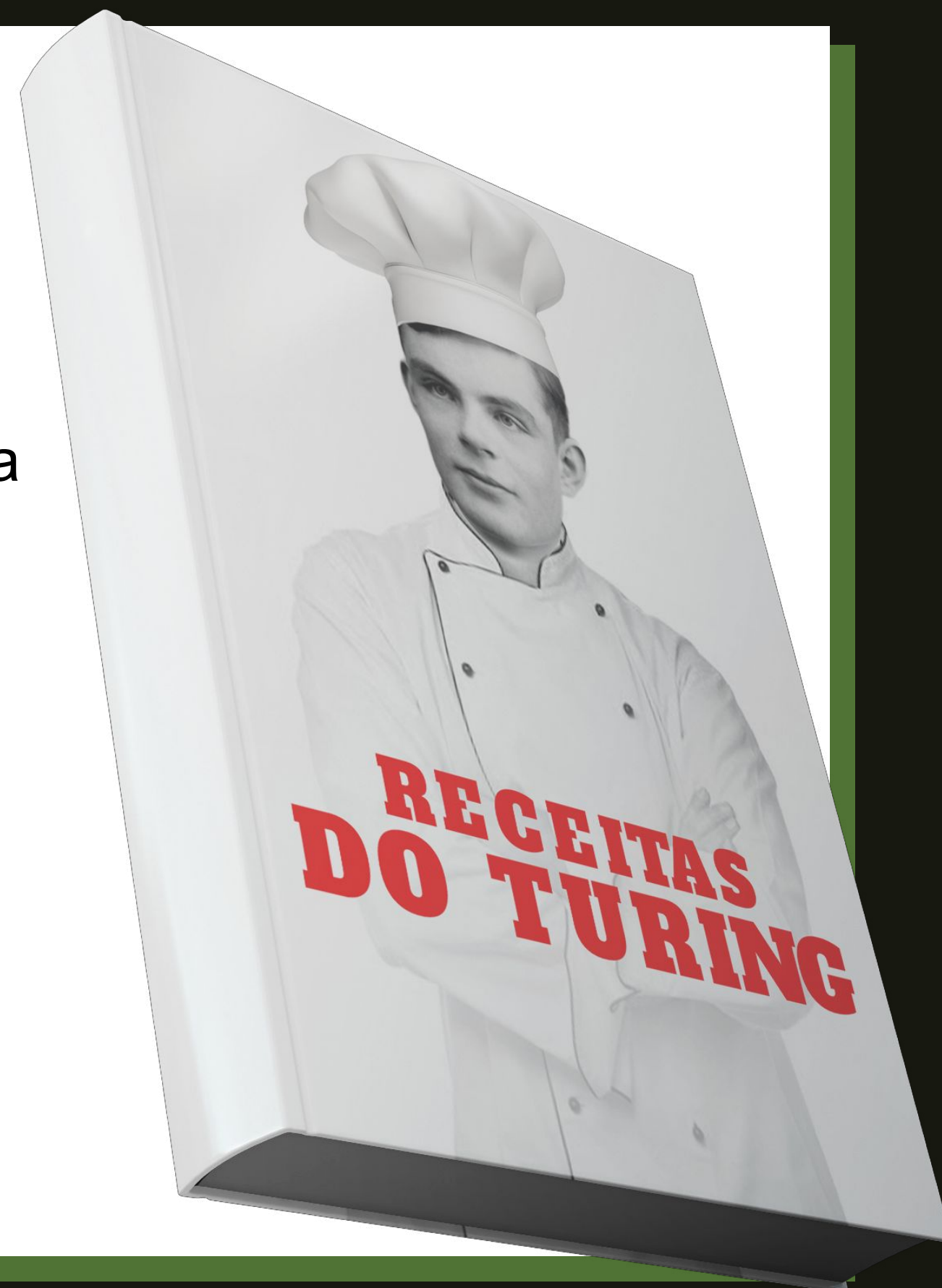


Analogia da cozinha

Programa

Um programa (**ou software, em inglês**) é uma sequência de instruções que o computador executa para chegar a um determinado resultado.

Podemos comparar a uma receita de cozinha, por exemplo, já que ela nada mais é do que uma sequência de ações a serem realizadas a fim de produzir um prato



Memória

A memória, no computador, armazena programas e os dados que são utilizados por eles.

Podemos compará-la a um armário, que é onde armazenamos as receitas e ingredientes que usamos nelas.



Processador

O processador é um componente que lê os programas, e que busca e salva os dados na memória do computador.

Podemos imaginar como sendo o cozinheiro, que lê a receita, pega e guarda ingredientes e prepara os pratos. Quanto mais capaz ele for, mais rápido prepara os pratos.





Evolução dos computadores

Primeiros computadores

Os primeiros computadores eram gigantescos, ocupando uma sala inteira. Usavam tubos de vácuo e interruptores que eram operados a mão para executar cálculos. **Não tinham um programa armazenado**, ou seja, precisavam ser reconfigurados fisicamente para cada tarefa. Eram usados para cálculos científicos e militares. **Não tinha uma tela**: os resultados eram impressos em uma folha de papel.



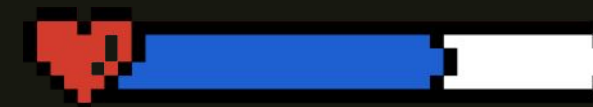
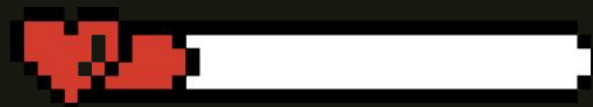
Computadores de hoje

Os computadores de hoje em dia são capazes de **cálculos bem complexos**, fazem análise de dados, rodam inúmeros jogos, servem para trabalho (como produzir textos, planilhas e apresentações), etc.

São simples de usar, **focados para todo o público** e não apenas programadores.

É uma máquina que **executa programas**.

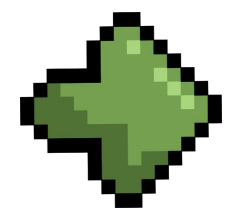




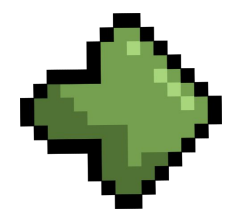
Sistemas operacionais



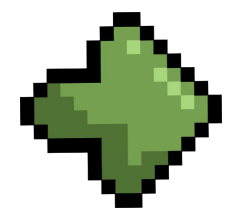
Programas responsáveis por gerenciar o computador



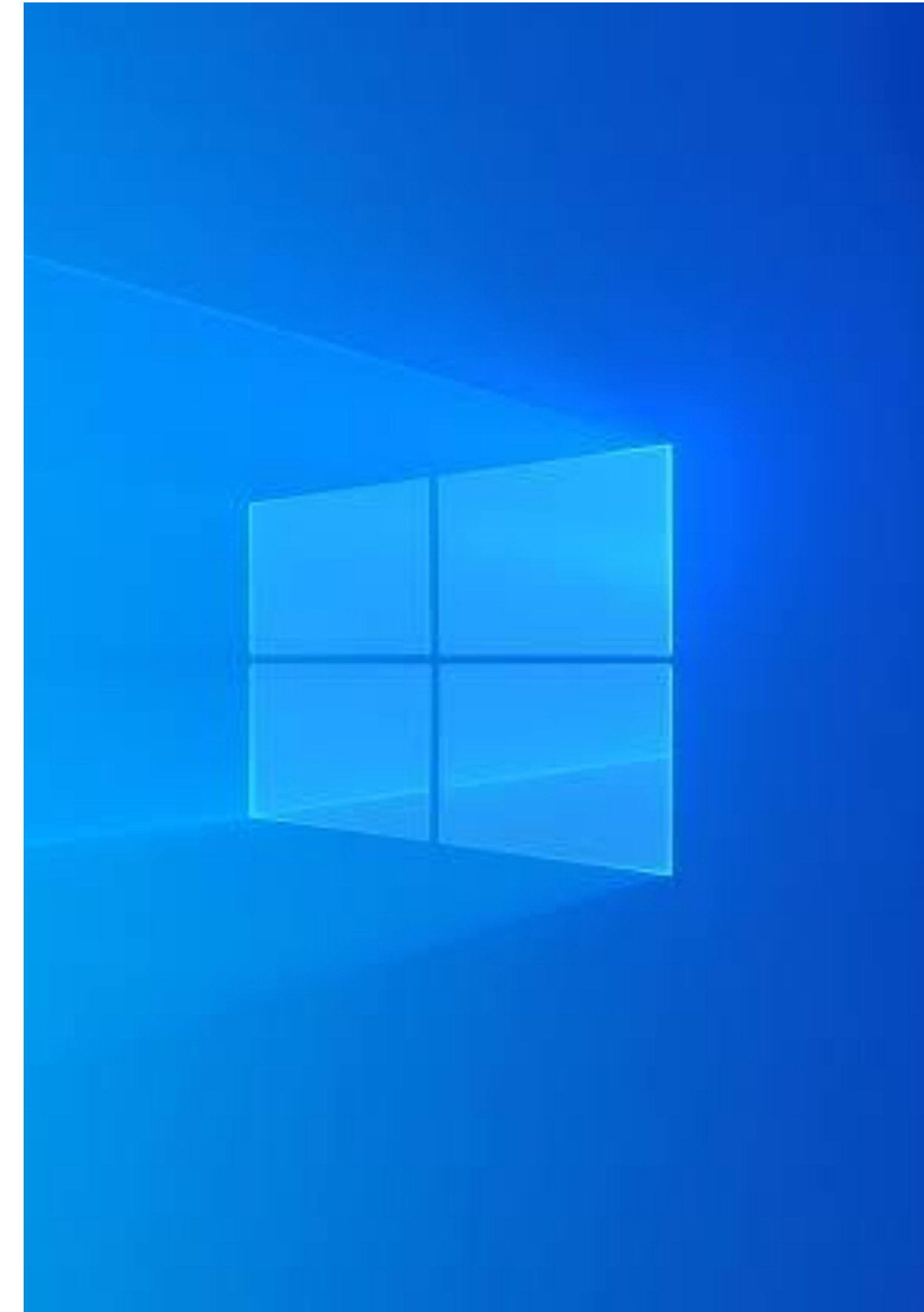
Organiza vários programas rodando ao mesmo tempo



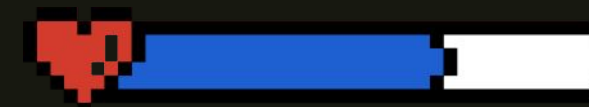
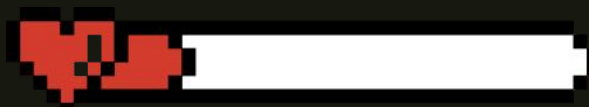
Gerencia o sistema de arquivos e pastas



Apresenta interface para o computador, o que permite interação do usuário







O que é
interface?

CLI

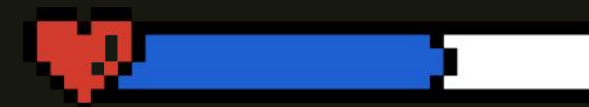
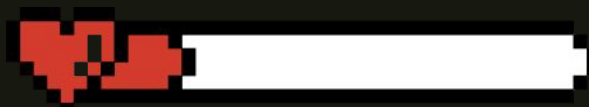
Antigamente, nos computadores, você não via nada gráfico, nem mesmo a ponteira do mouse. Era como nesta foto ao lado: se você quisesse "mexer" no computador você precisava inserir comandos em texto, por meio do que chamamos de Command Line Interface (CLI) que significa Interface de Linha de Comando.



GUI

Com o tempo, foram desenvolvidas interfaces gráficas, ou seja, para usar o computador você não precisa escrever linhas de comando. Você abre uma pasta clicando com o mouse, por exemplo. Essas interfaces são chamadas de Graphical User Interface (GUI), ou Interface Gráfica de Usuário.





Conhecendo o terminal

O que é

Programa que permite interação com o computador por meio de comandos (CLI), ou seja, ao invés de operarmos o computador da maneira convencional, nós o operamos através dos comandos, como explicado anteriormente. **Eles dependem do sistema operacional.**



Bash



PowerShell



Zsh



**GitBash - Acessar
comandos Unix**



CMD



Sistemas Unix (Linux e MacOS)

Herdaram o terminal da época em que não existiam interfaces gráficas



Bash: Bash/Zsh: um terminal muito poderoso

Windows

Foi desenvolvido com foco na interface gráfica. Os terminais disponíveis são, em geral, menos interessantes para nós.



GitBash: simula o Bash no Windows. Usaremos ele em aula.



Porque usar o terminal?

- Alguns programas não possuem interface gráfica.
- Permite automatização de tarefas. Exemplo: fazer backup automático de arquivos por meio de um script.
- Força maior entendimento por parte do usuário.
- É bem mais rápido e eficiente que a interface gráfica.

Fixação

- O processador é responsável pela execução de programas, acessando a memória da máquina
- Sistemas Operacionais são programas que criam interfaces para o usuário
- Terminal é um ambiente onde nós devs executamos comandos para trabalhar mesmo sem interface gráfica



Conhecendo os comandos

O que é?

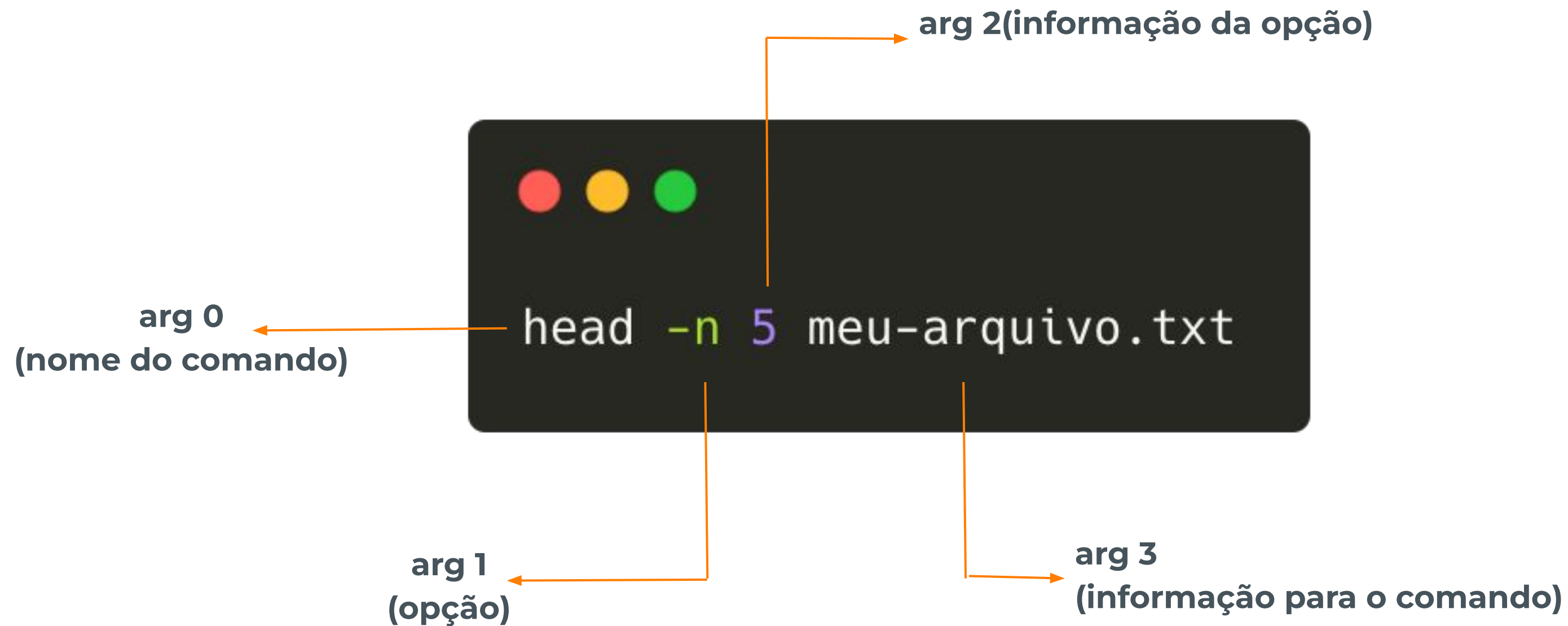
Um comando é uma sequência de palavras e letras que executam uma determinada ação. Cada elemento dessa sequência é chamado de **argumento**.

Cada comando pode possuir **opções** e **parâmetros**:

opções: mudam o comportamento do comando e possuem o caractere “-” ou “--” como prefixo

parâmetros: são informações atribuídas ao próprio comando ou às opções

- **Exemplo:** Visualizar as primeiras 5 linhas de um arquivo chamado *meu-arquivo.txt*



- **echo**: Imprime algo no terminal



```
echo "Hello World" # imprime Hello World no terminal
```

- **whoami**: Mostra o nome do usuário na tela



```
whoami # retorna o nome do usuário atual ex: 'seuUsuario'
```

- **clear:** Limpa tudo que está aparecendo no terminal. Só serve pra ajudar na leitura e organização



```
clear #limpa tudo que está no terminal
```

- **pwd:** Sigla para print working directory (mostrar diretório de trabalho). Mostra o endereço completo do diretório/pasta em que estamos trabalhando



pwd # retorna a pasta que o terminal está atualmente ex: '/c/Users/seuUsuario'

ls: O nome vem de list (lista). Lista os arquivos e pastas do diretório em que estamos



```
ls # retorna nome de arquivos e pastas presentes na pasta atual  
ls -a # retorna nome de arquivos e pastas, incluindo os ocultos (cujo nome começa com `.`)  
ls -l # retorna informações de arquivos e pastas, incluindo tamanho, proprietário e outras informações  
ls -la # soma dos dois modificadores anteriores
```


- **cd:** Sigla para change directory (mudar diretório). Muda o diretório/pasta em que estamos



```
cd ./minha-pasta # troca o diretório atual para a subpasta "minha-pasta"  
cd # vai para a pasta "home" do usuário atual  
cd ../ # vai para a pasta acima da atual
```

- **mkdir:** Sigla para make directory (criar diretório).
Cria um novo diretório



```
mkdir minha-pasta # cria uma pasta chamada 'minha-pasta' no diretório atual
```

- **touch:** Cria um novo arquivo.



```
touch index.html # criará um arquivo chamado index.html na pasta atual
```

- **rm:** Vem da palavra remove. Possibilita apagar arquivos. Ele apaga de uma maneira IRREVERSÍVEL e SEM PEDIR CONFIRMAÇÃO



```
rm ./meu-arquivo-gigante.txt # remove imediatamente o arquivo 'meu-arquivo-gigante.txt'  
rm -r ./minha-pasta # remove Recursivamente todos os arquivos e sub-pastas da 'minha-pasta'
```

- **mv:** Vem da palavra move. Permite mover arquivos de um diretório para outro. Pode ser usado também para renomear arquivos



```
mv ./meu-arquivo-gigante.txt ./minha-sub-pasta # move 'meu-arquivo-gigante' para 'minha-sub-pasta'  
mv ./meu-arquivo-gigante.txt ./meu-gigante.txt # renomeia 'meu-arquivo-gigante.txt' para 'meu-gigante.txt'
```

- **cp:** Vem da palavra copy (copiar). Copia arquivos de um diretório para outro



```
cp ./meu-arquivo-gigante.txt ./minha-sub-pasta # copia 'meu-arquivo-gigante' para 'minha-sub-pasta'
```

- **cat:** Vem da palavra concat (concatenar - unir). Ele concatena tudo que está no arquivo e imprime no terminal



```
cat meu-arquivo-gigante.txt # imprime o conteúdo do arquivo 'meu-arquivo-gigante.txt'
```

- **head:** Head significa "cabeça". Imprime as 10 primeiras linhas de um arquivo. A opção -n permite indicar quantas linhas queremos ver



```
head meu-arquivo-gigante.txt # imprime as 10 primeiras linhas do arquivo 'meu-arquivo-gigante.txt'  
head -n 20 meu-arquivo-gigante.txt # imprime as 20 primeiras linhas do arquivo 'meu-arquivo-gigante.txt'
```


- **tail:** Tail significa "cauda". Imprime as 10 últimas linhas de um arquivo. A opção -n permite indicar quantas linhas queremos ver



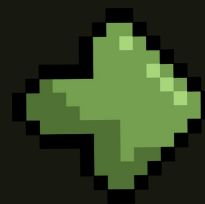
```
tail meu-arquivo-gigante.txt # imprime as 10 últimas linhas do arquivo 'meu-arquivo-gigante.txt'  
tail -n 20 meu-arquivo-gigante.txt # imprime as 20 últimas linhas do arquivo 'meu-arquivo-gigante.txt'
```

- **grep:** Permite buscar um determinado texto no conteúdo de um arquivo
- -A x imprime x linhas após o texto
- -B y imprime y linhas antes do texto

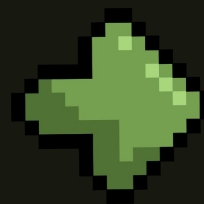


```
grep Future4 ./lista-de-empresas.txt # Busca pela palavra Future4 no arquivo lista-de-empresas.txt e  
imprime toda a linha encontrada
```

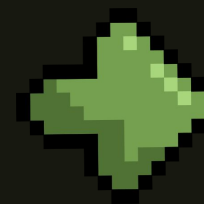
Fixação



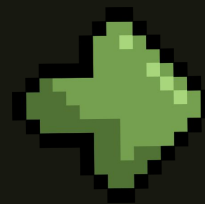
whoami



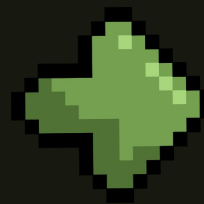
cd



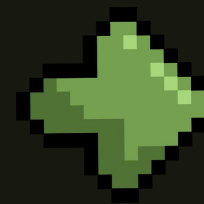
cp



echo



touch



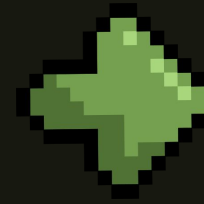
cat



clear



mkdir



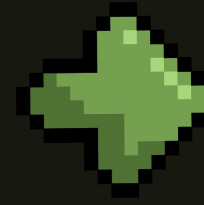
head



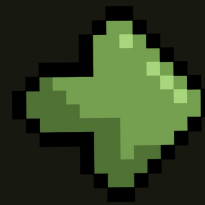
pwd



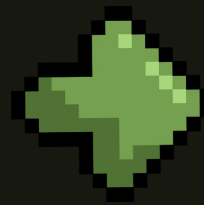
rm



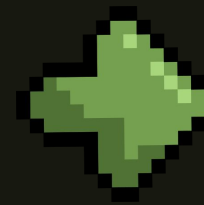
tail



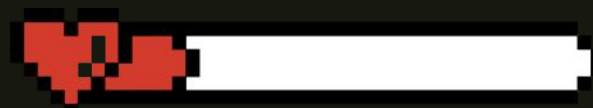
ls



mv



grep



Hora
da prática!

Passo 1

Abrir o terminal dentro da pasta do template

Dica: botão direito do mouse em qualquer parte dentro da pasta

Opcional: em vez de abrir direto na pasta, navegar da raiz (root) até ela manualmente com o comando 'cd'.

Passo 2

Ler o conteúdo do arquivo de texto pokemons.txt

Dica: comando 'cat'

Descobrir qual é o número do Pikachu

Dica: comando 'grep'

Passo 3

Descobrir os dois pokemons que vêm antes do Pikachu

Dica: comando 'grep' com opção -B

Descobrir os três pokemons que vêm depois do Pikachu

Dica: comando 'grep' com opção -A

Passo 4

Mostrar apenas os pokemons da primeira geração (do 1 ao 151)

Dica: comando 'head' com opção -n

Mostrar apenas os 100 últimos pokemons da lista

Dica: comando 'tail' com opção -n

Muito
obrigado!

Programa
3000 TALENTOS TI
Obrigado(a)!