

# Conhecendo o Computador e o Terminal

# O que vamos ver hoje?

- Um pouco de ***história***
- ***Analogia*** da cozinha e cenário atual
- ***Terminal*** e ***comandos***

# Um pouco de história

# Surgimento 🎉

- **Alan Turing** criou um modelo **teórico** de uma máquina que seria capaz de **seguir instruções** baseadas em 0 ou 1, realizando toda e qualquer operação computacional
- Turing também se envolveu na construção de máquinas **físicas** capazes de quebrar códigos de guerra alemães durante a segunda guerra mundial
  - **Filme: O jogo da imitação**

# Surgimento 🎉

- **Máquina de Turing**
  - Fita escrita que pode ser **lida**, **escrita** ou **movimentada**
  - Utiliza uma tabela lógica para realizar estas operações
- Com essa estrutura, é possível **calcular qualquer coisa** que os computadores atualmente conseguem 🤖

# Analogia da Cozinha

# Analogia da cozinha 🍝

- **Programa** ou **software** é sequência de instruções a serem executadas a fim de produzir um resultado
- **Receita** é sequência de ações a serem realizadas a fim de produzir um prato

# Analogia da cozinha 🍝

- **Memória:** armazena programas e dados utilizados por eles
- **Armário:** armazena receitas e ingredientes utilizados por elas



# Analogia da cozinha 🍪

- **Processador:** lê os programas, buscando e salvando dados na memória
- **Cozinheiro:** lê a receita; e prepara os pratos, pegando e guardando ingredientes no armário

# Computadores hoje

# E hoje em dia?

- **Máquinas poderosas**
  - Cálculos complexos
  - Análise de dados
  - Jogos
  - Trabalho (textos, planilhas e apresentações)
- **Simples e intuitivo**, focado em usuários "leigos"
- 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

# Interfaces

- Todos os **comandos** eram por **texto**, por meio de uma *Command Line Interface* (CLI)
  - Terminal
- Com o tempo, foram desenvolvidas interfaces **mais amigáveis** e **intuitivas**, chamadas de *Graphical User Interface* (GUI)

# Terminal

# Terminal

- Programa que permite interação com o computador por meio de comandos (**CLI**)
- Existem vários terminais, alguns exemplos abaixo:
  - Bash
  - Zsh
  - CMD
  - PowerShell
  - GitBash - Acessar comandos Unix

# Terminal

- Os terminais dependem do **Sistema Operacional**
- **Sistemas Unix (Linux e MacOS)**
  - Herdaram o terminal da época em que não existiam interfaces gráficas
  - **Bash/Zsh**: um terminal muito poderoso



# Terminal

- **Windows**

- Foi desenvolvido com foco na **interface gráfica**
- Os **terminais** disponíveis são, em geral, menos interessantes para nós
- Usaremos, então, o **GitBash** que simula o bash no Windows

# Terminal

- **Por que usar o terminal?**
  - Alguns programas **não** possuem **interface gráfica**
  - Permite **automatização** de tarefas
  - 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**

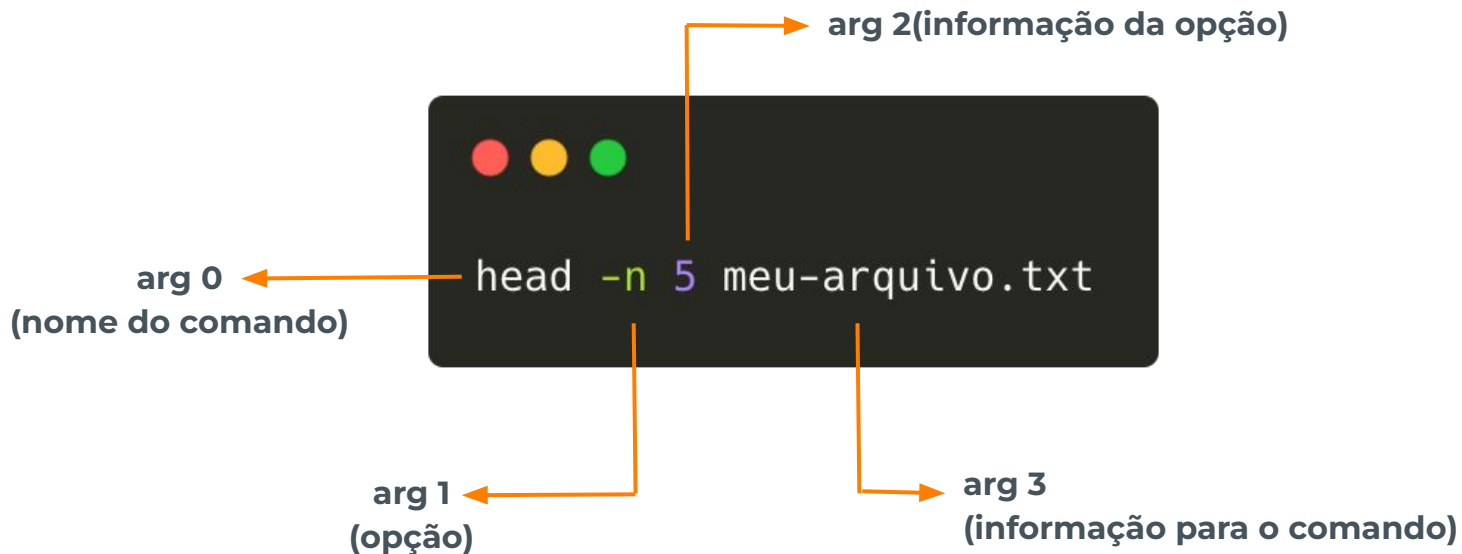
# Comandos

# O que são comandos?

- 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

# Comandos

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



# Comandos Básicos

- **echo**
  - Imprime algo no terminal



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

# Comandos Básicos

- **whoami**
  - Imprime o nome do usuário na tela


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



# Comandos Básicos

- **clear**

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



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

# Comandos Básicos

- **pwd**
  - Sigla para *print working directory*
  - Mostra o endereço completo do diretório/pasta em que estamos trabalhando



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

# Comandos Básicos

- **ls**
  - O nome vem de *list*
  - 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
```

# Comandos Básicos

- **cd**
  - Sigla para *change directory*
  - 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
```

# Comandos Básicos


- **mkdir**
  - Sigla para *make directory*
  - Cria um novo diretório



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

# Comandos Básicos

- **touch**
  - Cria um novo arquivo

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

# Comandos Básicos

- **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'
```

# Comandos Básicos

- **mv**
  - Vem da palavra ***move***
  - Permite mover arquivos de um diretório para outro
  - Pode ser usado 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'
```



# Comandos Básicos

- **cp**
  - Vem da palavra ***copy***
  - Copia arquivos de um diretório para outro

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

# Comandos Básicos

- **cat**
  - Vem da palavra *concat*
  - 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'
```

# Comandos Básicos

- **head**
  - 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'
```

# Comandos Básicos

- **tail**
  - 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'
```

# Comandos Básicos

- **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

Vimos os seguintes comandos:

- *whoami*
- *echo*
- ***clear***
- *pwd*
- ***ls***
- ***cd***
- ***touch***
- ***mkdir***
- ***rm***
- *mv*
- *cp*
- ***cat***
- ***head***
- ***tail***
- ***grep***

# Vamos Praticar!

# Praticando

## 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'



# Praticando

Ler o conteúdo do arquivo de texto pokemons.txt

- *Dica: comando 'cat'*

Descobrir qual é o número do Pikachu

- *Dica: comando 'grep'*

# Praticando

**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**

# Praticando

**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**

# Resumo

# Resumo



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

# Resumo

- Vimos os seguintes comandos:
  - *whoami*
  - *echo*
  - ***clear***
  - *pwd*
  - ***ls***
  - ***cd***
  - ***mkdir***
  - ***touch***
  - ***rm***
  - *mv*
  - *cp*
  - ***cat***
  - ***head***
  - ***tail***
  - ***grep***

# Dúvidas?



Programa  
**3000 TALENTOS TI**  
Obrigado(a)!