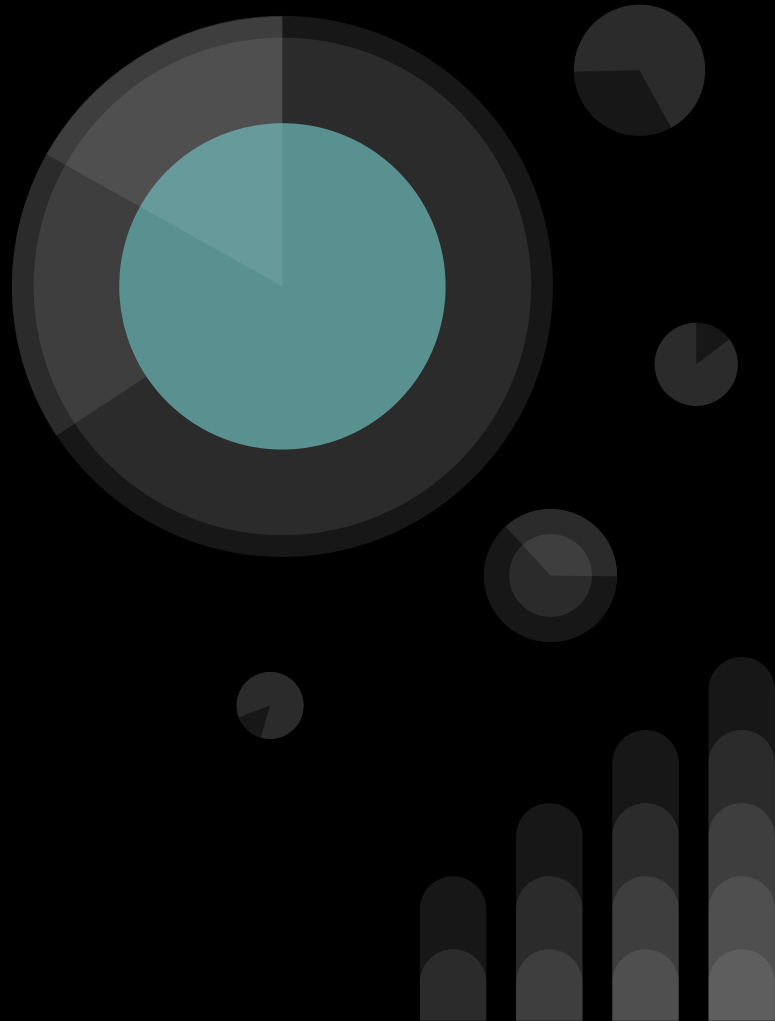


42Bootcamp

Aula-1

Professor : Alfredo Manuel





Compilador

Um compilador é um software que converte código-fonte de uma linguagem de programação em um formato que um computador pode entender e executar, geralmente traduzindo-o para código de máquina ou outra forma executável.



Shell vs Bash

"Shell" e "Bash" são termos relacionados a ambientes de linha de comando em sistemas operacionais Unix e Unix-like. Aqui está a diferença entre eles:

Shell:

- O termo "shell" refere-se ao ambiente de linha de comando que permite a interação do usuário com o sistema operacional.
- É uma interface entre o usuário e o núcleo do sistema operacional.

Bash:

- "Bash" é um tipo específico de shell, sendo uma abreviação de "Bourne Again SHell".
- É um shell de texto que funciona como uma melhoria e extensão do shell original do Unix, chamado "sh" (Bourne Shell).



Nano vs vim

nano e vim são ambos editores de texto para terminais Unix-like, mas têm abordagens e funcionalidades diferentes.

Nano:

- Interface mais simples e amigável para usuários iniciantes.
- Usa atalhos na parte inferior da tela para comandos.
- Possui uma interface mais intuitiva e fácil de usar, sem muitos recursos avançados.
- É uma boa escolha para usuários que estão começando a programar ou editar arquivos de texto no terminal.

Exemplo de uso:

```
nano nome do arquivo.txt
```



Nano vs vim

Vim:

- Editor poderoso e altamente configurável, mas pode ter uma curva de aprendizado íngreme para iniciantes.
- Divide a tela em modos diferentes (normal, inserção, visual, etc.).
- Oferece uma ampla gama de recursos e extensões.
- Requer algum tempo para aprender os comandos, mas oferece uma eficiência significativa quando você os domina.

Exemplo de uso:

```
vim nome do arquivo.txt
```



Git vs Github

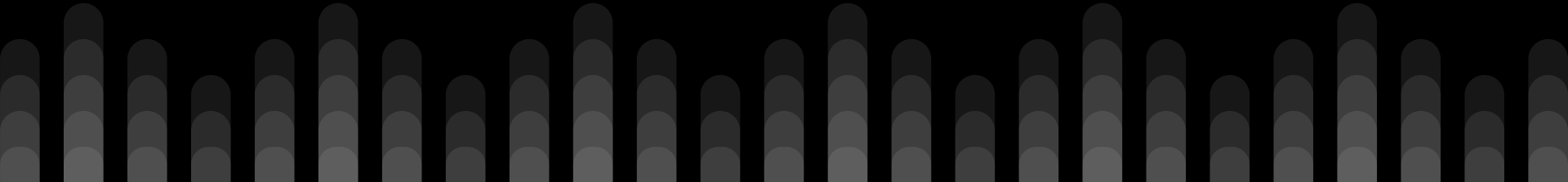
Git:

- Git é um sistema de controle de versão distribuído.

GitHub:

- GitHub, por outro lado, é uma plataforma de hospedagem de código-fonte que utiliza o Git.

Comandos Bash





- **Comandos de Navegação:**

- a. `cd`: Muda o diretório atual.
 - i. Exemplo: `cd /caminho/do/diretorio`.
- b. `ls`: Lista os arquivos e diretórios no diretório atual.
 - i. Exemplo: `ls -l` (lista detalhada).

- **Operações em Arquivos e Diretórios:**

- a. `cp`: Copia arquivos ou diretórios.
 - i. Exemplo: `cp arquivo.txt destino/`.
- b. `mv`: Move ou renomeia arquivos ou diretórios.
 - i. Exemplo: `mv arquivo.txt novo_nome.txt`.
- c. `rm`: Remove arquivos ou diretórios.
 - i. Exemplo: `rm arquivo.txt`.



- **Manipulação de Diretórios:**

- a. `mkdir`: Cria um novo diretório.

- i. Exemplo: `mkdir novo_diretorio`.

- b. `rmdir`: Remove um diretório vazio.

- i. Exemplo: `rmdir diretorio_vazio`.

- **Visualização de Conteúdo:**

- a. `cat`: Exibe o conteúdo de um arquivo.

- i. Exemplo: `cat arquivo.txt`.

- b. `less` ou `more`: Permite visualizar o conteúdo de um arquivo de forma paginada.



- **Comandos de Texto:**
 - a. `echo`: Exibe uma mensagem na tela ou redireciona para um arquivo.
 - i. Exemplo: `echo "Olá, Mundo!" > arquivo.txt`.
 - b. `grep`: Procura padrões em arquivos de texto.
 - i. Exemplo: `grep "palavra" arquivo.txt`.
- **Comandos de Ajuda:**
 - a. `man`: Exibe o manual do usuário para um comando específico.
 - i. Exemplo: `man ls`.
 - b. `--help`: Exibe informações de ajuda para um comando específico.
 - i. Exemplo: `ls --help`.
- **Comandos de Usuário e Permissões:**
 - a. `whoami`: Exibe o nome do usuário atual.
 - b. `chmod`: Modifica as permissões de um arquivo ou diretório.
 - i. Exemplo: `chmod +x script.sh` (torna o script executável).

Fim

