# Mentoria #1

Dica: Dê o seu toque pessoal no projeto. Demonstre os desafios que você passou durante o projeto.

# LÓGICA DE PROGRAMAÇÃO ESSENCIAL

## Introdução à Lógica e à Programação

### Entendendo o que é lógica

Programação serve para resolver problemas e não só saber códigos. Você precisa saber resolver o problema e a lógica que será utilizada antes.

A abstração é uma das melhores habilidades de um programador, pois ele consegue resolver um problema de modo mais simples, abstraindo algumas informações desnecessárias.

### O que é algoritmos e pseudocódigo

Algoritmo é uma sequência de passos que resolve um problema.

Pseudocódigo é um passo a passo de uma tarefa que pode ser transformada em código.

### Fluxograma e variáveis

Fluxograma é uma ferramenta utilizada para representar graficamente o algoritmo, isto é, a sequência lógica e coerente do fluxo de dados.

Variáveis na programação é um objeto (uma posição, frequentemente localizada na memória) capaz de reter e representar um valor ou expressão. Elas podem ser numéricas, caracteres, alfanuméricas ou lógicas. Pode ser alterado durante a execução de um algorítimo.

### Tomadas de decisões e expressões

Expressões aritméticas são expressões que utilizam operadores aritméticos e funções aritméticas envolvendo constantes e variáveis.

Expressões literais são expressões com constantes e/ou variáveis que tem como resultado valores literais. Iremos utilizar as expressões literais na atribuição de valor para uma variável ou constante.  
Ex.: nome=”José da Silva”......media=(nota1+nota2)/2

Expressões relacionais são compostas por outras expressões ou variáveis numéricas com operadores relacionais. As expressões relacionais retornam valores lógicos (verdadeiro/falso)”.  
Ex.: 1 > 2 : Falso....2 == 2: verdadeiro

Tomadas de decisão são 60% da programação. Quando escrevemos programas, geralmente ocorre a necessidade de decidir o que fazer dependendo de alguma condição encontrada durante a execução.  
Ex.: Programa um site que só vende em cartão, então terá a programação de que se tem cartão continua, se não tem, exibe mensagem que só vende por cartão.

### Como utilizar a concatenação

Concatenação é um termo utilizado na computação para designar a operação de unir conteúdo de duas strings (sequência de caracteres).

## Introdução ao Portugol

### Estrutura de repetição

Dentro da lógica da programação é a estrutura que permite executar mais de uma vez o mesmo comando ou conjunto de comandos, de acordo com uma condição ou com um contador.

### Linguagem de programação e o Portugol

Linguagem de programação é uma linguagem escrita e formal que especifica um conjunto de instruções e regras usadas para gerar programas (software). Um software pode ser desenvolvido para rodar um computador, dispositivo móvel ou em qualquer equipamento que permita sua execução. A função da linguagem de programação é servir de um meio de comunicação entre computadores e humanos.

**Alto nível:** São aquelas cuja sintaxe se aproxima mais da nossa linguagem e se distanciam da linguagem de máquina (C, PHP, Phyton e etc..).

**Baixo nível:** São linguagem de computador, essas são as que você precisa ter conhecimento direto da arquitetura do computador para fazer alguma coisa (Assembly).

**Compiladas:** Linguagem em que o código fonte, é executado diretamente pelo sistema operacional ou pelo processador, após ser traduzida por meio de um processo chamado compilação (C#, Visual Basic, Delphi, C++). Precisa transformar elas em arquivo executável e rodar sobre o sistema operacional.

**Interpretadas:** Linguagem em que o código fonte é executado por um programa de computador chamado interpretador, que em seguida é executado pelo sistema operacional ou processador. (Java Script, PHP, Pyton, entre outros).

**O que é Portugol? :** É uma pseudolinguagem que permite o leitor desenvolver algoritmos estruturados em português de forma simples e intuitiva, independentemente da linguagem de programação.

### Desvios condicionais e comentários no Portugol

**Desvio condicional “se-senao”:** É utilizada para a condição a ser testada entre parênteses e as instruções que devem ser executadas entre chaves caso o desvio seja verdadeiro. Caso seja falso, irá retornar outra função.

*se (media>=7) {*

*Escreva(“Parabéns!! Você foi aprovado!!”)*

*}*

*Senão {*

*Escreva(“Infelizmente você foi reprovado”)*

*}*

**Desvio condicional “caso”:** Este comando é similar ao “se” e “senão”, e reduz a complexidade na escolha de diversas opções. Ele possui algumas diferenças. Neste comando não é possível o uso de operadores lógicos, ele apenas trabalha com valores definidos.

*Inteiro valor=0*

***Escolha*** *(valor)*

*{*

***Caso*** *1:*

*Escreva(“Ok, abrir Netflix!”)*

***Caso*** *2:*

*Escreva(“Ok, abrir Amazon Prime!”)*

***Caso 3:***

*Escreva(“Ok, abrir HBO GO!”)*

***Pare***

***Caso contrario:***

*Escreva(“Você deve escolher a opção 1,2 ou 3”)*

*}*

**Laços de repetição:** Dentro da lógica de programação é uma estrutura que permite executar mais de uma vez um comando ou um conjunto de comandos, de acordo com uma condição ou contador.

*Função inicio()*

*{*

*Inteiro contador, limite, resultado*

*Contador = 0*

*Limite = 10*

*Faca*

*{*

*Resultado=9\*contador*

*Escreva(“9 X” + contador + “=” + resultado + “\n”)*

*Contador ++*

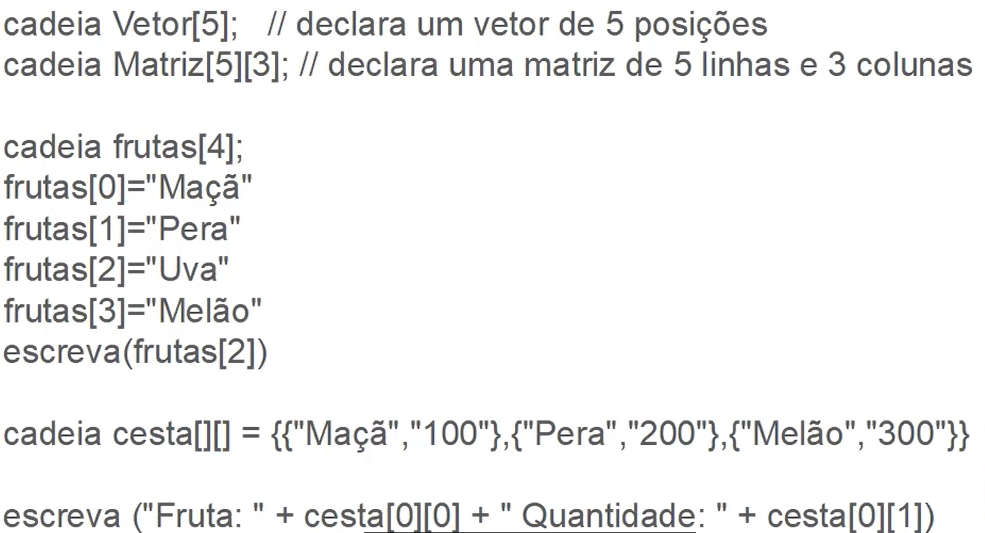
*}enquanto (contador<=limite)*

*}*

### Aplicação prática com matrizes e vetores

**Matriz:** É uma coleção de variáveis de mesmo tipo (você não pode ter dentro de uma matriz caracteres do tipo inteiro e do tipo cadeia ao mesmo tempo), acessíveis com um único nome e armazenados contiguamente na memória.. A individualização de cada variável de um vetor é feita através do uso de índices.

**Vetores:** São matrizes de uma só dimensão.



# Aprenda o que são Estrutura de Dados e Algoritmos

## Conceitos iniciais sobre Estrutura de Eados, Arrays e Registro

### O que é Estrutura de Dados

Estrutura de dados é uma estrutura organizada de dados na memória de um computador ou em qualquer dispositivo de armazenamento, de forma que os dados possam ser utilizados de forma correta.

Essas estruturas encontram muitas aplicações no desenvolvimento de sistemas, sendo que algumas são altamente especializadas e utilizadas em tarefas específicas. Usando as estruturas adequadas através de algoritmos, podemos trabalhar com uma grande quantidade de dados, como aplicações de bancos de dados ou serviços de busca.

Um algoritmo é um conjunto de instruções estruturadas e ordenadas, seu objetivo é realizar uma tarefa ou operação específica.

Os algoritmos são utilizados para manipular dados nas estruturas de várias formas, como por exemplo: inserir, excluir, procurar e ordenar dados.

Em uma estrutura de dados devemos saber como realizar um determinado conjunto de operações básicas, como por exemplo:

* Inserir dados;
* Excluir dados;
* Localizar um elemento;
* Percorrer todos os itens constituintes da estrutura para visualização;
* Classificar, que se resume em colocar os itens de dados em uma determinada ordem (numérica, alfabética, etc.);

**Principais estruturas de dados:**

* Vetores e Matrizes (Arrays);
* Registro;
* Lista;
* Pilha;
* Fila;
* Árvore;
* Tabela Hash;
* Grafos;

### Vetores e Matrizes

Vetores e Matrizes ou Arrays são estruturas de dados simples que podem auxiliar quando há muitas variáveis do mesmo tipo em um algoritmo.

<https://portugol-webstudio.cubos.io/ide>

Vetor ou array uni-dimensional é uma variável que armazena várias variáveis do mesmo tipo. O vetor é uma estrutura de dados indexada, que pode armazenar uma determinada quantidade de valores do mesmo tipo.

Matriz ou array multi-dimensional é um vetor de vetores. Uma matriz é um vetor que possui duas ou mais dimensões.

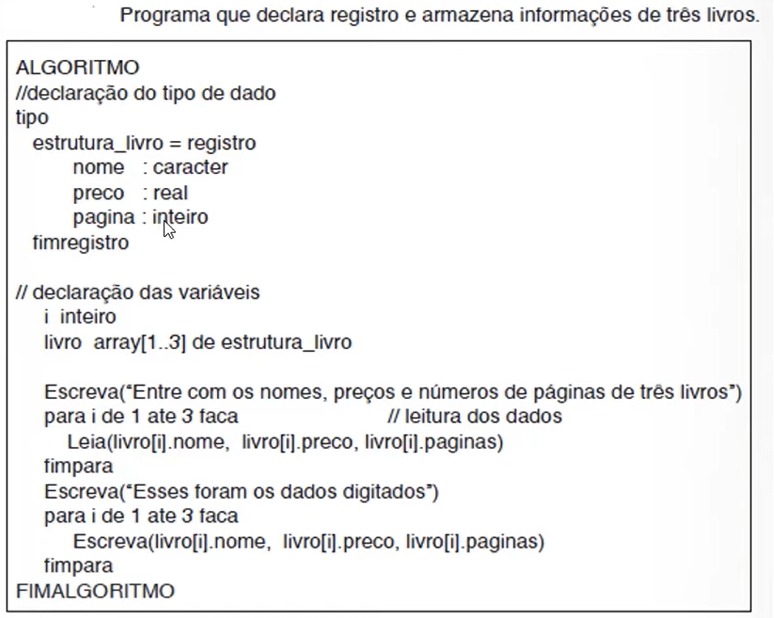
### O que são Registros

Registro é uma estrutura que fornece um formato especializado para armazenar informação em memória. Enquanto Arrays nos permitem armazenar vários dados de um único tipo de dados, o recurso de Registro nos permite armazenar mais de um tipo de dado.   
Ex.: Matriz de caracteres só vai possuir caracteres, já o Registro nós podemos mesclar.

Um Registro é composto por campos que especificam cada uma das informações que compõem. Abaixo podemos ver alguns campos de exemplo que constituem o registro de um cliente:

* CPF;
* Nome;
* Endereço;
* Contato;

Toda estrutura de registro tem um nome (ex: livro), e seus campos podem ser acessados por meio de um operador ponto (.). Por exemplo, para acessar o preço de um livro, poderíamos utilizar a seguinte declaração: *livro.preço*

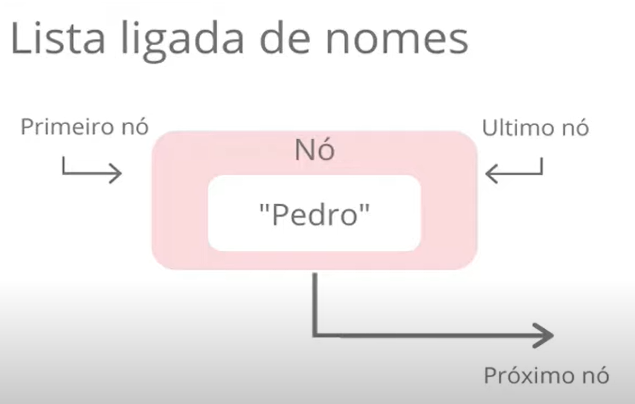


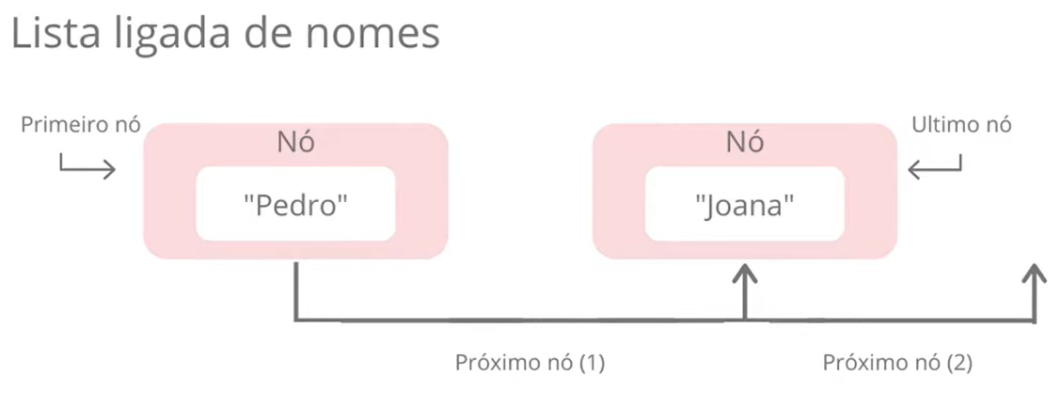
### Entenda o que são Listas, Pilhas e Filas

**Listas**: Éestrutura de dados que armazena dados de um determinado tipo em uma ordem específica. A diferença entre Listas e Arrays(matrizes e vetores) é a de que as listas possuem tamanho ajustável, enquanto Arrays possuem tamanho fixo.

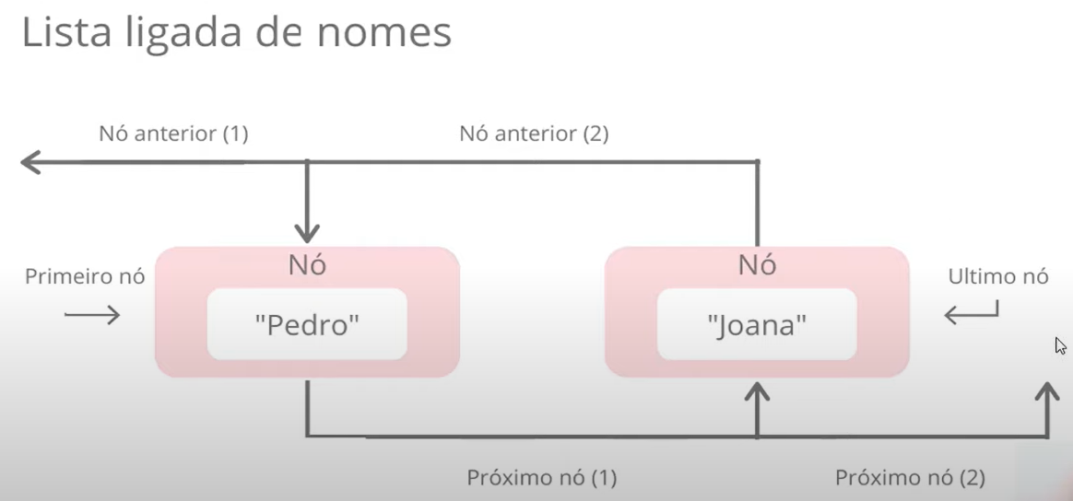
Existem dois tipos de listas:

* **Listas ligadas:** na estrutura do tipo Lista existem os nós onde cada um dos nós conhece o valor que está sendo armazenado em seu interior além de conhecer o elemento posterior a ele: por isso ela é chamada de “Lista ligada”, pois os nós são amarrados com essa indicação de qual é o próximo nó.





* **Listas duplamente ligadas:** Constituem uma nova variação de listas ligadas. A grande diferença das listas duplamente ligadas para as listas ligadas PE que elas são bidirecionais. Vimos que, naturalmente, não conseguimos “andar para trás” em listas ligadas, pois os nós de uma lista ligada sabem somente quem é o próximo elemento. Nas listas duplamente ligadas os nós sabem quem é o próximo elemento e o anterior, o que permite a navegação reversa.

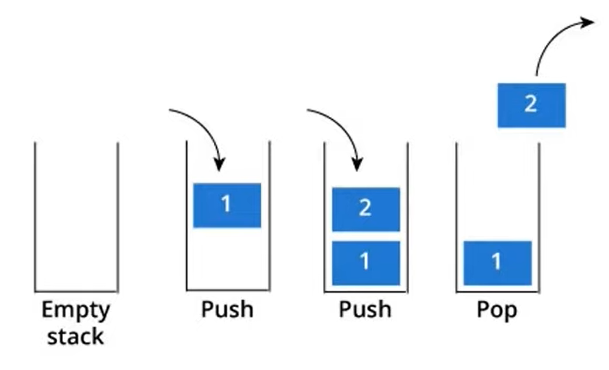


### O que são Pilhas

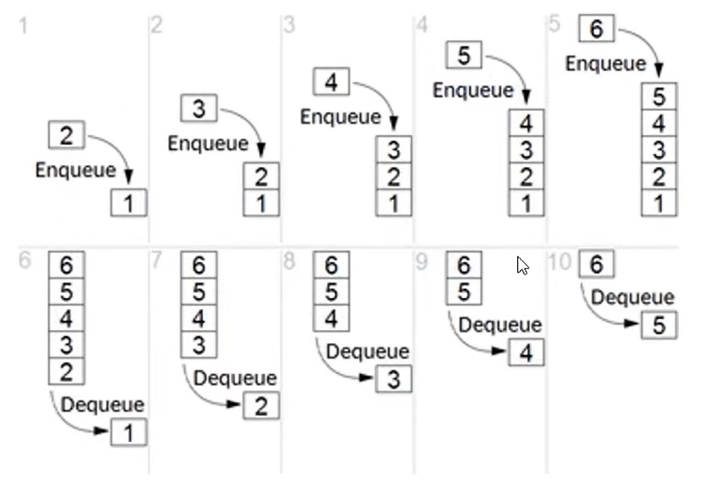
Uma pilha é uma estrutura que serve como uma coleção de elementos, e permite o acesso somente a um item de dados armazenado. O acesso aos itens de uma pilha é restrito – somente um item pode ser lido ou removido por vez.

Tipos de pilhas:

* **LIFO (Last in First Out) ou UEPS (Último que Entra, Primeiro que Sai):** Apresenta o seguinte critério: o primeiro elemento a ser retirado é o último que tiver sido inserido;

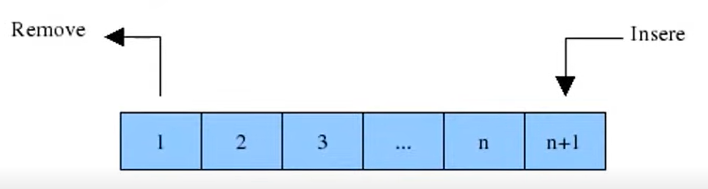
****

* **FIFO (First In First Out) ou PEPS (Primeiro que Entra Primeiro que Sai):** O primeiro elemento a ser retirado é o primeiro que tiver sido inserido;

****

### O que são Filas

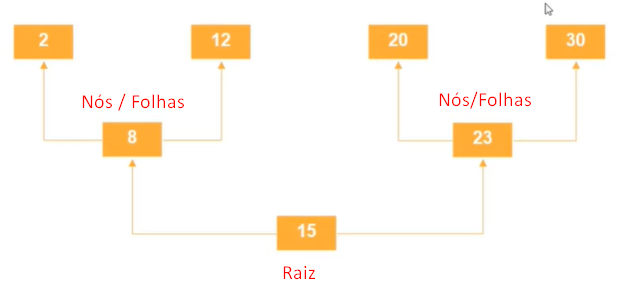
A estrutura do tipo Fila admite remoção de elementos e inserção de novos sujeita à seguinte regra de operação: O elemento a ser removido é o que justamente está na estrutura há mais tempo, ou seja, o primeiro objeto inserido na fila é também o primeiro a ser removido seguindo o conceito FIFO.



## Estruturas de dados do tipo Árvore, Tabela Hash e Grafos

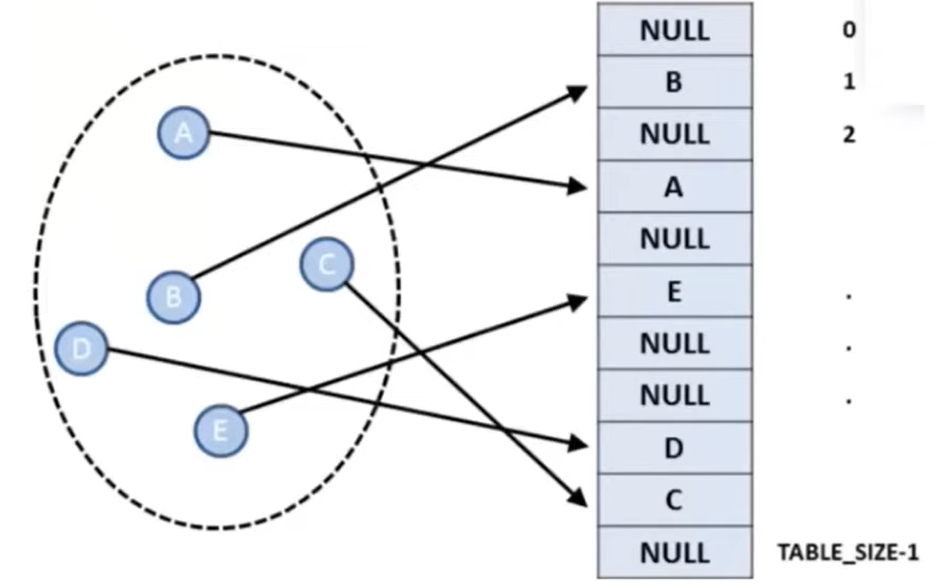
### O que são Árvores

É uma estrutura de dados que organiza seus elementos de forma hierárquica, onde existe um elemento que fica no topo da árvore, chamado de raiz e existem os elementos subordinados a ele, que são chamados de nós ou folhas.



### O que são Tabela Hash

Uma tabela hash, de dispersão ou espelhamento é uma estrutura de dados especial, que associa chaves de pesquisa a valores. Uma tabela hash é uma generalização da ideia de array, porém utiliza uma função denominada Hashing para espalhar os elementos, fazendo com que os mesmo fiquem de forma não ordenada dentro do “array” que define a tabela.



A tabela hash permite a associação de “valores” a “chaves”.

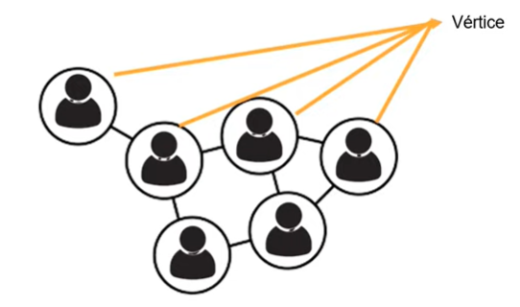
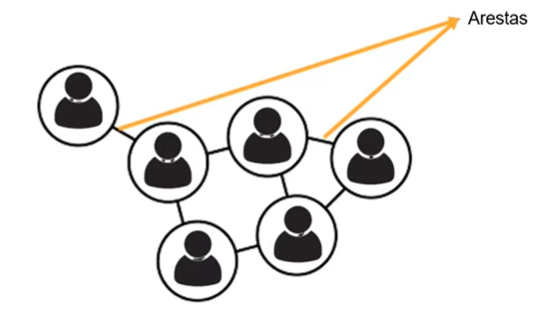
Valores: é a posição ou índice onde o elemento de encontra.

Chaves: parte da informação que compões o elemento a ser manipulado.

Espalhar facilita a busca na estrutura de dados, pois a partir de uma chave podemos acessar de forma rápida uma posição do “array”.

### O que são grafos

Grafos são estruturas que permitem programar a relação entre objetos. Os objetos são vértices ou “nós” do grafo. Os relacionamentos são arestas.

# Introdução do Git e ao GitHub

## Introdução ao Git

### Entendendo o que é Git e a sua importância

Git é um sistema de versionamento de código distribuído criado por Linus Torvald(criador do LINUX), que ajuda você a monitorar e criar diferentes versões do seu código.

Imagine que você é um escritor e então você tem uma ideia e passa ela pro papel, depois pede pro seu editor revisar, então ele sugere alterações, então esse arquivo deixa de ser o principal e passa a ser o “Antigo”, logo após você altera e realiza um novo documento e então o editor revisa novamente e logo esse “novo” vira o “Antigo II”. O Git é uma ferramenta dessa.

Tenha em mente de que um software não é feito de maneira solo, mas sim de maneira colaborativa, assim como o LINUX.

Benefícios de aprender Git e GitHub:

* Controle de Versão
* Armazenamento em nuvem
* Trabalho em equipe
* Melhorar seu código
* Reconhecimento

## Navegação via command line interface e instalação

### Comandos básicos para um bom desempenho no terminal

O Git é um software CLI(Command Line Interface), diferente dos comuns que são GUI(Graphical Use Interface), ou seja, é por linha de comando que você trabalha nele.

Comandos Windows:

* **Dir**: Me traz todas as pasta do diretório
* **Cd /** : Me leva ao diretório C
* **Cd (nome da pasta que quer):** Ele leva até a pasta
* **Cd** **..**: Ele volta um diretório
* **Cls**: Limpa a tela do prompt
* **Tecla TAB**: Ele auto completa o nome da pasta que você quer, desde que a pasta esteja dentro do seu diretório.
* **Mkdir**: Ele cria uma pasta no diretório
* **Echo**: Ele printa de volta na linha de baixo do CMD um texto que você escreveu depois de echo
* Ex. **echo hello > hello.txt**: Ele vai criar um arquivo com nome hello em txt e dentro do arquivo vai estar escrito “hello”, que foi o que colocou antes do >.
* **Del (nome da pasta):** ele irá deletar tudo o que existir de arquivo dentro da pasta.
* **↑**: Ele navega entre os comandos digitados anteriormente.
* **Rmdir**: Remove o diretório e tudo o que está dentro.

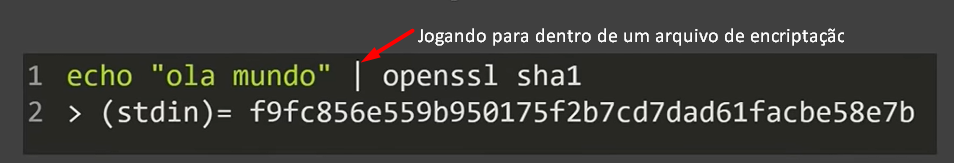
### Instalação do Git

Git-cms.com

## Entendendo como o Git funciona por baixo dos panos

### Tópicos fundamentais para entender o funcionamento do Git

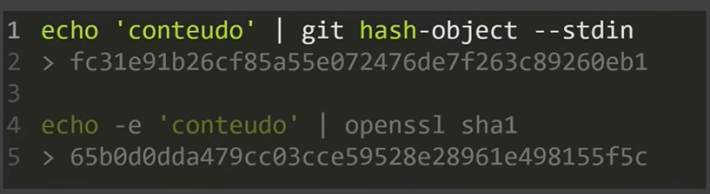
**SHA1 –** A sigla SHAsignifica Secure Hash Algorithm (Algoritmo de Hash Seguro), é um conjunto de funções hash criptográficas projetadas pela NSA (Agência de Segurança Nacional dos EUA). A encriptação gera um conjunto de caracteres identificador de 40 dígitos. É uma forma curta de representar um arquivo.

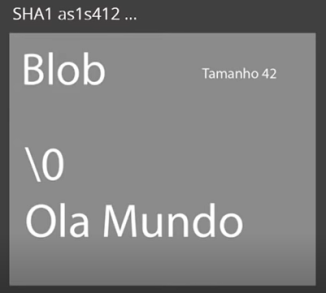


Se você alterar uma virgula do arquivo, ele irá gerar outro conjunto de caracteres, porém se você voltar como estava o texto antes e gerar de novo, ele voltará a gerar o mesmo conjunto que gerou primeiro.

### Objetos internos do Git

**Blobs –** Ele contém os metadados do SHA1. Os blobs só guardam os SHA’s.

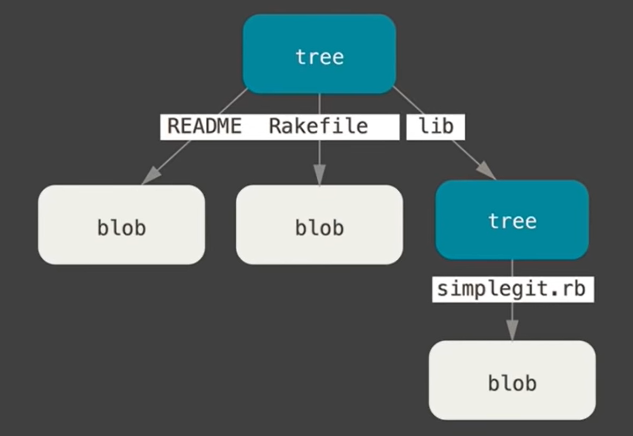




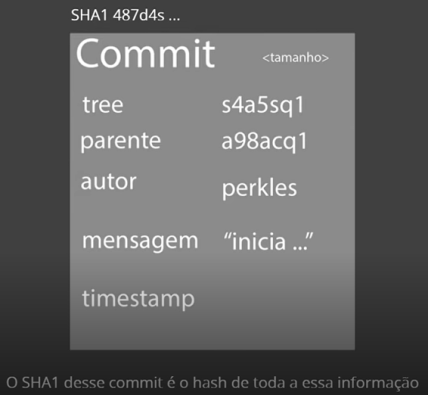


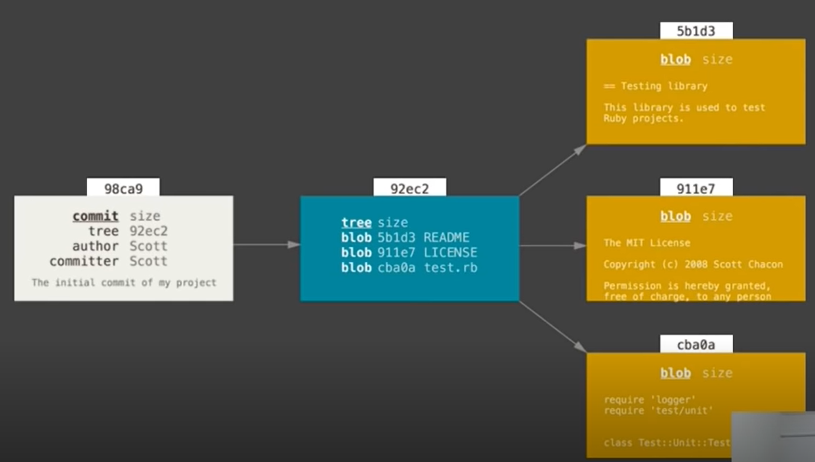
**Tree –** Armazenam Blobs. Ela também contém metadados do Git. Ela é responsável por montar toda a estrutura e onde estão localizados os arquivos.

Uma tree pode apontar para outra tree, como um diretório pode estar dentro de outro diretório.

**Commit –** Ele é o objeto que vai juntar tudo, que vai dar sentido a alteração.





### Chave SSH e Token

A Chave SSH é uma forma de estabelecer uma conexão segura e encriptada entre duas máquinas. Sempre haverá duas chaves, uma pública e uma privada.

O GitHub pode utilizar essa chave para poder manter a conexão segura, assim você não precisa se conectar a ele utilizando usuário e senha. Para pegar uma chave e inserir no GitHub deve fazer o seguinte, no windows:

* Vá ao Git Bash
* Ssh-keygen –t ed25519 –C (seu email)[eduardokawamurajr@gmail.com](mailto:eduardokawamurajr@gmail.com)
* Enter
* Insere a senha
* Chave gerada
* Ctrl+l = limpa tela
* Cd /c/Users/user(nome do usuário do PC)/.ssh/
* Ls
* Cat id\_ed25519.pub
* Então será demonstrada a sua chave pública, copia ela e cola dentro do Github
* Volta pro Cli
* Ls
* Pwd
* Eval $(ssh-agent –s)
* Ls
* Ssh-add id\_ed25519
* Coloca a senha

Pronto, tudo sincronizado...você pode apagar a chave no Github e depois você pode fazer o processo de novo para gerar uma nova chave.

**Token de acesso pessoal –** É outro método para um acesso no Github sem precisar ficar utilizando usuário e senha:

* Dentro do GitHub
* Developer settings
* Personal Access tokens
* Generate new token
* Configura a data de expiração e seleciona o “repo”
* Generate token
* Copia ele e salva em algum lugar que você tenha acesso

Agora se você for copiar um repositório pessoal do Git pra máquina, ele vai solicitar esse token para finalizar o processo.

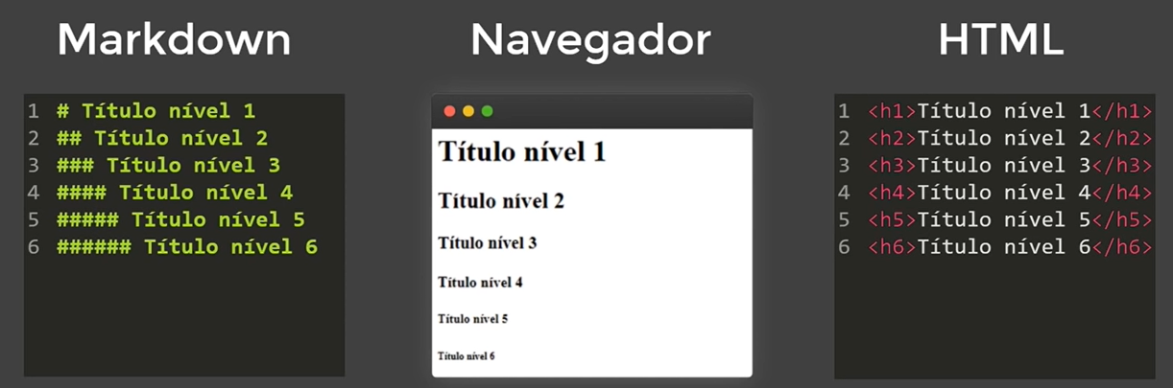
## Primeiros passos com o Git

### Iniciando Git e criando um commit

Comando para iniciar um git dentro de um repositório no Git Bash = git init

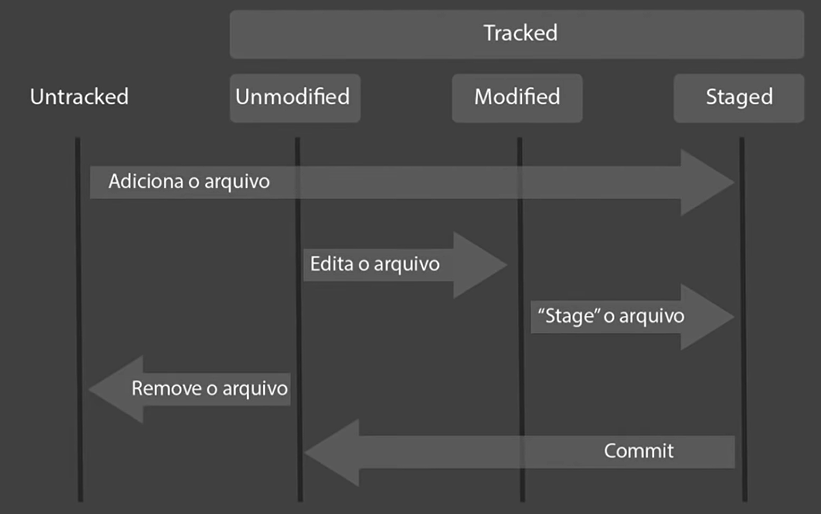
A flag “-a” mostra arquivos ocultos.

Podemos adicionar o Markdown no arquivo Git para “humanizar” mais a forma de escrever HTML dentro do Git.



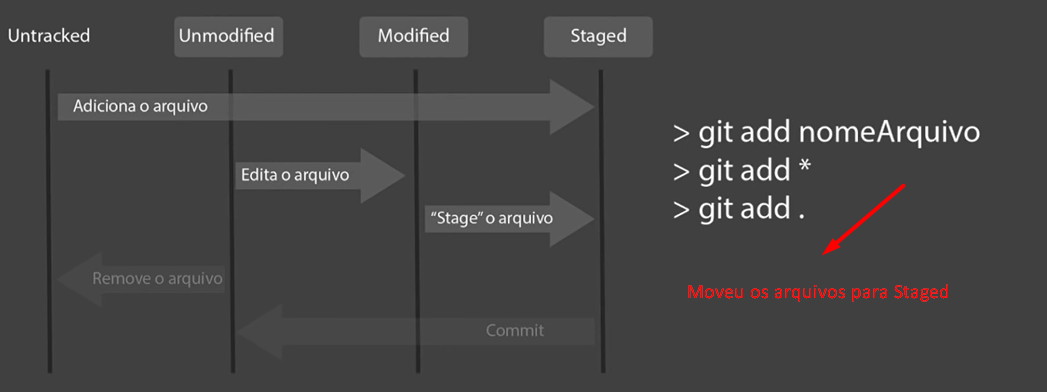
## Ciclo de vida dos arquivos Git

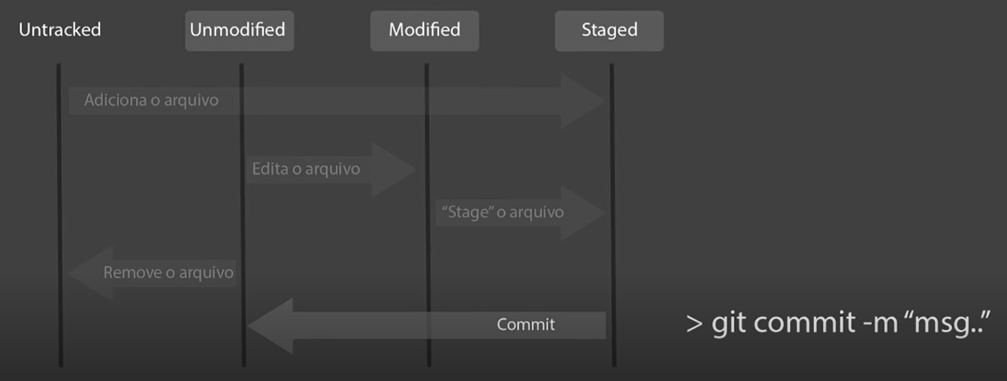
### Passo a passo no ciclo de vida



**Git Status –** Vai nos ajudar a monitorar o status dos arquivos, se ele está Untracked, Unmodified, Modified ou Staged.

**Git add \* ou . -** Ele vai adicionar tudo o que têm no seu diretório para o staged para posteriormente commit.







## Introdução ao GitHub

### Trabalhando com o GitHub

* Verificar os arquivos se estão sincronizados(commitados) ou não – **git status**
* Para adicionar o arquivo para *staged* – **git add .** ou **\***
* Para indicar para o repositório remoto – **git commit –m “**(uma mensagem sobre o commit/atualização que está fazendo**”**
* Para mandar de fato para o repositório remoto – **git push origin main**
* Para buscar o repositório/clonar do GitHub – **git clone** (colar o HTTP)