Estruturas de dados

-Vetor: Dados do mesmo tipo organizados no mesmo lugar, também conhecidos como arrays.

-Matrizes: São vetores entrelaçados em dimensões diferentes

-Registro: Comporta vários tipos de informações (inteiros, reais caracteres)

-Listas: Podem ter tamanhos ajustados, existem as ligadas (Existem 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**) e a duplamente ligada (Elas são bidirecionais, vimos que naturalmente, não conseguimos andar para atrás 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 também quem é o elemento anterior, o que permite a navegação reversa.**

-Pilhas: Funciona como uma coleção de elementos, e permite o acesso a somente um item de dados armazenado. Tem acesso **restrito- Somente um item pode ser lido ou removido por vez.**

LIFO(Last in First out) UEPS: O primeiro a ser retirado é o ultimo que tiver sido inserido.

FIFO ou PEPS: O primeiro a ser retirado é o primeiro que tiver sido inserido.

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

-Árvore: 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 e folhas. Ela se estrutura como uma árvore com tronco principal e bifurcações. É mais dinâmica.

-Tabela Hash ( Tabela Espalhamento): Uma tabela hash, de dispersão ou espalhamento é uma estrutura de dados especial, que associa chaves de pesquisa a valores.

A tabela hash permite a associação de valores a chaves.

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

Chave: Parte da informação que compõe o elemento a ser manipulado.

-Grafos: Grafos são estruturas que permitem programar a relação entre objetos, os objetos são vértices ou “nós” do grafo e os relacionamentos são as arestas.

Comandos Básicos:

**Dir:** Listar pastas/ ls no linux

**Cd**: Selecionar uma pasta

**Cd..** : Volta uma pasta

**Limpar a tela**: Clear

A tecla tab auto completa o comando

Mkdir: para criar uma nova pasta

Del: apagar arquivo

Apagar uma pasta :rmdir\_\_\_\_/s/q

Del/Rmdir no Windows/Rm/rf no Linux

Git init

Git . :Adicionar todos os arquivos na commit

Git Commit -m: Criar um commit

**Comandos do Git:**

**Dir:** Listar pastas/ ls no linux

**Cd**: Selecionar uma pasta

**Limpar a tela**: Clear

Mkdir: para criar uma nova pasta

rmdir\_\_\_\_/s/q: Apagar uma pasta

Del: apagar arquivo

**git init:** iniciar um novo repositório

**Git Commit -m:** Criar um commit

**git add .** adiciona os arquivos atuais ao próximo commit

**git status** Verificar o status atual dos repositório

**git commit -m** cria um novo commit com uma mensagem

**git push** envia as atualizações para a nuvem na branch atualmente ativa

**git remote add origin:** colocar o link do seus repositório

**git reflog:** Verificar todas as atualizações do git

**git reset –hard:** Colocar o código da commit que você quer retornar

**git branch:** permite listar e ver qual branch está atualmente ativa

**git checkout** **nome-da-branch**: criar uma nova branch

**git checkout -b “nome da branch de origem” nova branch:** permite mudar e criar uma nova branch com base em outra

**git merge “branch a receber merge”:** permitir fazer o merge da branch ativa atualmente com outra branch

**git pull:** atualiza a branch atualmente ativa

**git config –list:** lista de configurações

**git config –global –unset:** mudar configuração

git clone: clonar uma repositório