**E-mail: profa.jaqueline@gmail.com**

**Aula 15/02:**

- Métodos de ordenação;

- Tipo abstrato de dados:

Exemplo: Pilha; Fila 🡪 conceito atrelado a um processo;

- Hash 🡪 Método para criptografia

- Lista encadeada 🡪 Não possui tamanho fixo 🡪 Conceito semelhante de vetor para JavaScript;

- Lista dinâmica;

- Árvore 🡪 Estrutura não-linear 🡪 Deu origem ao banco de dados;

- Recursividade 🡪 Uma função chamando ela mesma;

**Aula 20/02:**

- Métodos de ordenação:

- Seleção; Inserção e troca 🡪 Selection sort; Bubble sort e Quick sort; Insert sort;

Bubblesort 🡪 Método mais simples de ordenação 🡪 não é muito eficiente

25 57 48 37 12 92 86 33

25 57 48 37 12 92 86 33

25 48 57 37 12 92 86 33

25 48 37 57 12 92 86 33

25 48 37 12 57 92 86 33

25 48 37 12 57 92 86 33

25 48 37 12 57 86 92 33

25 48 37 12 57 86 33 92 🡪 primeira iteração

25 48 37 12 57 86 33 92

25 48 37 12 57 86 33 92

25 37 48 12 57 86 33 92

25 37 12 48 57 86 33 92

25 37 12 48 57 86 33 92

25 37 12 48 57 86 33 92

25 37 12 48 57 33 86 92 🡪 Segunda iteração

25 37 12 48 57 33 86 92

25 37 12 48 57 33 86 92

25 12 37 48 57 33 86 92

25 12 37 48 57 33 86 92

25 12 37 48 57 33 86 92

25 12 37 48 33 57 86 92 🡪 Terceira iteração

25 12 37 48 33 57 86 92

12 25 37 48 33 57 86 92

12 25 37 48 33 57 86 92

12 25 37 48 33 57 86 92

12 25 37 33 48 57 86 92 🡪 Quarta iteração

12 25 37 33 48 57 86 92

12 25 37 33 48 57 86 92

12 25 37 33 48 57 86 92

12 25 33 37 48 57 86 92 🡪 Quinta iteração

12 25 33 37 48 57 86 92

12 25 33 37 48 57 86 92

12 25 33 37 48 57 86 92 🡪 Sexta iteração

12 25 33 37 48 57 86 92

12 25 33 37 48 57 86 92 🡪 Sétima iteração 🡪 Fim do código;

Selection sort 🡪 Busca pelo menor e troca com o primeiro da fila

10 20 12 5 8 15

5 20 12 10 8 15

5 8 12 10 20 15

5 8 10 12 20 15

5 8 10 12 20 15

5 8 10 12 15 20

25 57 48 37 12 92 86 33

12 57 48 37 25 92 86 33

12 25 48 37 57 92 86 33

12 25 33 37 57 92 86 48

12 25 33 37 57 92 86 48

12 25 33 37 48 92 86 57

12 25 33 37 48 57 86 92

12 25 33 37 48 57 86 92

Insertion sort 🡪 Insere o valor na sua posição certa;

25 57 48 37 12 92 86 33

25 57 48 37 12 92 86 33

25 48 57 37 12 92 86 33

25 37 48 57 12 92 86 33

12 25 37 48 57 92 86 33

12 25 37 48 57 92 86 33

12 25 37 48 57 86 92 33

12 25 33 37 48 57 86 92 🡪 Vetor final.

**Aula 12/03**

**Quicksort:**

Dividir o problema e ordenar, escolher um pivô, a esquerda do pivô, ficam todos os menores que ele, a direita ficam todos os maiores;

25 57 48 37 12 92 86 33 🡪 pivo

i j

25 57 48 37 12 92 86 33 🡪 troca

I j

25 33 48 37 12 92 86 57 🡪 i para

I j

25 33 48 37 12 92 86 57

I j

25 33 48 37 12 92 86 57 🡪 j para 🡪 troca

I j

25 33 12 37 48 92 86 57 🡪 Divide o vetor em 2 na recursão

25 33 12 🡪 pivo

48 92 86 57 🡪 pivô

25 33 12 🡪 j para

i j

25 33 12 🡪 troca

i j

25 12 33 🡪 divide vetor

25 12 🡪 pivô

I j

25 12 🡪 para j e troca;

12 25

Segundo vetor:

48 92 86 57 🡪 j para

I j

48 92 86 57

I j

48 57 86 92 🡪 divide o vetor de novo

48 57 86 🡪 pivô

48 57 86

I j

48 57 86 🡪 para

I j

Vetor final:

12 25 33 37 48 57 86 92 🡪 ordenado

**Aula 26/03:**

HeapSort 🡪 define o heap como sendo metade do tamanho do vetor;

ShellSort 🡪 define um gap e compara pulando esse valor pelo vetor.

Big O 🡪 calculo da ordem de complexidade de um algoritmo (2^n, n^3, n^2, nlog2n, n, log2n)