**Módulo 01 Desenvolvimento 01**

**Aluno: Tomás de Farias Ribeiro Caldas**

**Aula 01**: Usar a lógica para realizar ações simples e definir o passo a passo das ações.

**Introdução do Projeto**

Construa um algoritmo de ordenação, utilizando o método *bubble sort* estudado. (Lembre-se que se trata de uma série de instruções que devem ser seguidas passo a passo).

Para isso, você deve criar um método em que o tamanho do vetor seja dez e deve estar em ordem crescente.

O vetor deverá:

- comparar seus elementos dois a dois adjacentes;

- se os elementos não estiverem em ordem, então ordenar;

- senão, avançar para o próximo par;

- repetir a operação até que nenhuma troca possa ser feita no vetor inteiro.

**Realize essa atividade no WORD ou no Bloco de Notas, suba esse arquivo para algum repositório e compartilhe o link no campo ao lado para que outros desenvolvedores possam analisá-lo.**

**Resolução:**

Passos do algoritmo podem ser seguidos da seguinte forma.

1. Percorrer o vetor até a penúltima posição executando os passos 2 e 3.
2. Comparar casa posição i com a posição seguinte (i + 1)
3. Se o elemento da posição i for maior que o seguinte, então os 2 são trocados de posição.
4. Quando todo o vetor tiver sido percorrido, verificar:
5. Se foi realizada ao menos uma troca , voltar ao passo 1.
6. senão, finalizar.

Abaixo apresento a codificação em Portugol. Criei um vetor inteiro de tamanho 10, e trêsa variáveis inteiras , i para a estrutura de repetição para, auxiliar para realizar a troca de posição dos elementos do vetor e, por fim, trocou, para sinalizar se devemos percorrer o vetor mais uma vez ou não.

Em seguida, fazemos uma repetição para sortear números entre 1 e 100 e popular o nosso vetor, imprimindo assim o vetor gerado, e aí então iniciarmos o processo de ordenação, realizando exatamente assim  um dos passos descritos acima.

1. Enquanto houver ao menos uma troca ( o valor 1 na variável trocou possui essa informação), a variável trocou, que iniciou com o valor 1 , recebe o valor 0
2. Todo o vetor é percorrido  até o penúltima posição.
3. Se o  elemento da posição i for maior que o seguinte, troca os dois elementos de posição e alterar o valor da variável trocou para 1.

por fim , escrevemos outro para para imprimir o vetor agora ordenado.

Código

programa

{

  inclua biblioteca Util

  funcao inicio()

  {

    inteiro vet[15], i, auxiliar, trocou = 1

    para(i = 0; i < 15; i++){

      vet[i] = Util.sorteia(1, 100)

    }

    escreva("Vetor gerado: ")

    para(i = 0; i < 15; i++){

      escreva(vet[i], " ")

    }

    escreva("\n")

    enquanto(trocou == 1){

      trocou = 0

      para(i = 0; i < 15 - 1 ; i++){

        se(vet[i] > vet[i+1]){

          auxiliar = vet[i]

          vet[i] = vet[i+1]

          vet[i+1] = auxiliar

          trocou = 1

        }

      }

    }

    escreva("\nVetor ordenado: ")

    para(i = 0; i < 15; i++){

      escreva(vet[i], " ")

    }

    escreva("\n")

  }

}