# Nome: Gabriel Garcia Couto RA:00240526 Relatório Trabalho Final

Incialmente foi criado um repositório no github onde será publicado o projeto. Feito isso, criei um projeto chamando de “ordenacao”.

Durante a leitura da proposta, levantei vários requisitos que deveriam ser atendidos para que seja um trabalho válido.

Identifiquei as classes necessárias: Class Bolha, Class Insercao e Class Selecao. Dentro delas declarei dois métodos públicos, um de “ordenacao” onde contém toda a lógica utilizada para organizar o vetor e outro de montar uma variável com todos os elementos do vetor.  
 Na classe main, foi iniciado solicitando ao usuário que informe a quantidade de posições que ele deseja no vetor. Após isso, foi criado uma validação switch-case onde o usuário escolhe se quer preencher manualmente as posições ou de forma automática, através da biblioteca útil.

Após alimentar o vetor, fica disponível ao usuário escolher qual método de ordenação que irá ser utilizado. Com isso, através de um novo switch-case, foram feitas as lógicas de todos os métodos. Primeiro instanciamos a classe, depois armazenamos o vetor em uma String para compará-lo após ordenar e assim chamamos o método de ordenação. Feito isso, o resultado é mostrado em uma String através do “SOUT”. Vale lembrar, que o tempo de execução também é visível para o usuário.

No fim, existe uma opção que mostra ao usuário o resultado dos 3 métodos e assim é mostrado no SOUT, comparando ao vetor e o tempo de execução.

1. Lógica utilizada no método de Inserção:

Esse método percorre o vetor a partir do segundo elemento, em seguida procura inserir na posição correta, comparando com o elemento que está à esquerda, trocando de ligar, caso o elemento for maior.

Para isso foi declarado uma variável chave como “elemento”. Adicionado o primeiro loop, onde iniciava na posição i = 1, ou seja, segunda posição do vetor, esse loop funciona enquanto posição i for menor que o tamanho do vetor.

Antes de abrir um novo loop, o elemento presente na posição i do vetor, é armazenado na variável chave e a variável j, passa a ser j = j – 1. Ao abrir um novo loop, ele vai continuar funcionado somente enquanto o j >= 0 e se elemento do vetor na posição j for maior que o elemento armazenado na variável chave, a cada repetição, o J é diminuído.

Após o segundo loop, o vetor na posição j+1 recebe o elemento da variável chave.

1. Lógica utilizada no método de Seleção:

Esse método guarda a posição do menor elemento e percorre toda a lista buscando por um valor menor. Se esse valor for encontrado, então é feito a substituição dos elementos que estavam sendo comparados. Por último é verificado se a posição do menor elemento é diferente da posição atual, se for verdade, é realizada a troca dos valores, colocando o menor elemento na frente.

Para isso, foi aberto o primeiro loop, onde a posição i = 0, indo até quando i for menor que o tamanho do vetor menos um. Com isso é armazenado o valor da posição i em uma variável de auxílio.

Após isso, é necessário utilizar outro for, que começa na posição onde j = i + 1, indo até quando o j for menor que o tamanho do vetor. Dentro dele é feita uma verificação onde se o vetor na posição j for menor que o vetor na posição auxiliar de i, a posição auxiliar de i recebe j.

Fechando o segundo loop, e antes de fechar o primeiro, é feita as trocas. Onde é declarada uma variável auxiliar que recebe o elemento do vetor na posição auxiliar de i, o vetor na posição auxiliar de i recebe o vetor na posição i e o vetor na posição i recebe o elemento da variável auxiliar.

1. Lógica utilizada no método de Bolha:

Esse método verifica se o elemento da direita é maior, caso o elemento da esquerda for maior, então é feito a troca das posições, esse processo é realizado até que o vetor fique totalmente ordenado.

Para isso, foram declaradas uma variável booleana incialmente como true e outra como auxiliar. Então é feito um while com a condição da variável booleana. Dentro do while, a variável booleana é setada para false e após é aberto um loop for, onde inicia na posição i = 0, enquanto i for menor que o tamanho do vetor. Dentro do for é feito uma verificação, onde olha se o vetor na posição i é maior que o vetor da posição i+1, caso seja verdadeiro, a variável auxiliar recebe o valor do vetor na posição i , o vetor na posição i recebe o vetor na posição i+1 e o vetor na posição i+1 recebe a variável auxiliar e a variável booleana recebe true novamente para poder verificar o loop novamente.