

# UMA IMPLEMENTAÇÃO EM LINGUAGEM JAVA DE ALGORITMO DE CLASSIFICAÇÃO BASEADO NO MÉTODO DA INSERÇÃO DIRETA

<sup>1</sup> BIM, Pedro Henrique Marana; <sup>1</sup> RABALDELLI, Diego José Caíres; <sup>1</sup> SOARES, Luis Antonio; <sup>2</sup> TAMAE, Rodrigo Yoshio; <sup>2</sup> MUZZI, Fernando Augusto Garcia

1 - Discentes do Curso Sistemas de Informação - FAEG/Garça

2 - Docente do Curso Sistemas de Informação - FAEG/Garça

[pedrohmbim@yahoo.com.br](mailto:pedrohmbim@yahoo.com.br); [rytamae@yahoo.com.br](mailto:rytamae@yahoo.com.br); [faqmuzzi@yahoo.com.br](mailto:faqmuzzi@yahoo.com.br)

## RESUMO

Apresentado com conceitos extremamente vagos e complexos, o método da inserção direta pode ser facilmente entendido através de aplicações práticas. Sua aceitação é ótima, principalmente perante os demais métodos de classificação interna simples, uma vez que é considerado o mais rápido, dentro de normas adequadas. Neste trabalho foi utilizada a linguagem Java, para o desenvolvimento da aplicação, atendendo a tendências mundiais de aplicabilidade da linguagem.

**Palavras chaves:** algoritmos de classificação, inserção direta e Linguagem Java.

## ABSTRACT

Shown with extremely empty and complex concepts, the method of the insertion sort, can easily be understood through practical applications. Its acceptance is excellent, mainly before the too much methods of simple internal classification, at present is considered fastest, inside of adequate norms. In this work the Java language was used, for the development application, taking care of the world-wide trends of applicability of the language.

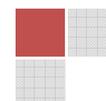
**Keywords:** classification's algorithms, insertion sort and Java Language.

## I-INTRODUÇÃO

No âmbito de estrutura de dados computacionais, os métodos de ordenação são meios extremamente importantes e necessários para classificação e organização de dados, em vetores, uma vez que possibilitam menor esforço, otimizando o processamento do software e máquina em questão, para a localização de dados, ou seja, o objetivo da ordenação é facilitar a localização dos membros de um conjunto de dados, e é uma atividade fundamental e universalmente utilizada para a elaboração de algoritmos mais complexos.

Assim, caracterizados os algoritmos de classificação, subentende-se ordenar seus componentes numa determinada ordem, segundo um critério específico; e é exatamente este critério é que proporciona uma variedade de métodos, entre os quais o da inserção direta.

Paralelamente, classificado como interno, o método da inserção direta, é caracterizado como o mais eficaz dentre os considerados básicos (bubblesort, inserção direta e seleção direta), e baseia-se, conceitualmente, segundo Cardoso (2006), em uma classificação onde inicialmente, considera-se o primeiro elemento



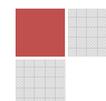
ordenado, o segundo elemento é, então, inserido na sua posição correta em relação ao primeiro, resultando as duas primeiras posições ordenadas. A seguir, o terceiro elemento é inserido na sua posição correta em relação aos dois primeiros, resultando nas três primeiras posições ordenadas; e assim sucessivamente .

A aplicação é descrita em duas classes, sendo a primeira exclusiva do método da inserção direta, expondo minuciosamente a lógica deste método com fins de fixação dos conceitos de algoritmos de ordenação, tão importantes, como visto, à estrutura de dados e posteriormente a aplicação interativa com o usuário, que terá como ponto fundamental à utilização do método da inserção direta, este responsável pela entrada dos elementos a serem ordenados.

## **2 - ALGORITMOS DE CLASSIFICAÇÃO**

Classificação de dados é o processo pelo qual é determinada a ordem que as entradas de uma tabela (os registros de um arquivo) serão apresentadas de forma que obedeça a uma organização previamente determinada por um ou mais campos (CEFETPB, 2006). O campo pelo qual o arquivo é ordenado pode ser chamado de chave de classificação.

A dependência da escolha de um algoritmo quanto à estrutura de dados a ser processada, é tão forte no caso da ordenação que os métodos usados são, em geral, classificados em duas categorias, ordenação de vetores e ordenação de arquivos, onde as duas classes são freqüentemente chamadas de ordenação interna e externa (WIRTH, 1999), ou seja, uma classificação é considerada interna se os registros que ela classificar estiverem na memória principal e externa se alguns dos registros que ela classificar estiverem no armazenamento auxiliar (CEFETPB, 2006). A classificação interna apresenta como principais características classificar apenas o que se encontra na memória principal, gastar o mesmo tempo de acesso para buscar qualquer dos endereços e o tempo médio de acesso não é afetado pela seqüência dos dados. Na classificação externa, os dados são transferidos em blocos para a memória principal para, só então, serem manipulados. Esses dados transferidos num acesso a disco influenciarão na eficiência do processamento e os dados são manipulados bloco a bloco. É importante notar que os métodos de classificação externa envolvem a aplicação de métodos de classificação interna, tomando a cada vez um subconjunto de dados a classificar.



Os métodos de classificação interna são divididos em cinco grupos de acordo com a técnica empregada, são eles: 1. Classificação por Inserção (inserção direta e shell); 2. Classificação por Troca (bubblesort e quicksort); 3. Classificação por Seleção (seleção direta e heapsort); 4. Classificação por Distribuição (distribuição de chaves e radixsort); e 5. Classificação por Intercalação (mergesort).

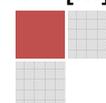
### **3 - MÉTODO DA INSERÇÃO DIRETA**

Dentre os métodos de inserção, o método da Inserção Direta é o mais simples, porém, é o mais rápido, entre os outros métodos considerados básicos – Bubblesort e Seleção Direta (PASSARO, 2006). Neste algoritmo, o próprio vetor é utilizado no processo de ordenação, não consumindo, portanto, memória para a separação dos segmentos do vetor, consome-se um pouco de memória somente para o armazenamento de variáveis auxiliares (PASSARO, 2006). Todavia, a eficácia do método de inserção está intrinsecamente ligada a uma adequação aos seguintes fatores: o número de registros a serem classificados; se todos os registros caberão ou não na memória interna disponível; e o grau de classificação já existente;

No entanto, a ordenação por inserção tem duas vantagens. Primeiro, ela se comporta naturalmente, ou seja, trabalha menos quando a matriz já está ordenada e o máximo quando a matriz está ordenada no sentido inverso. Isso torna a ordenação por inserção excelente para listas que estão quase em ordem. A segunda vantagem é que ela não rearranja elementos de mesma chave. Isso significa que um vetor que é ordenado por duas chaves permanece ordenado por ambas as chaves após uma ordenação por inserção (CCET-VIRTUAL, 2006).

Muito embora o número de comparações possa ser razoavelmente baixo para certos conjuntos de dados, o vetor precisa ser deslocado a cada vez que um elemento é colocado na sua posição correta, ou seja, o número de movimentações pode ser significativo. Há de se destacar ainda que o método proporciona que o próprio vetor seja usado no processo de ordenação, não consumindo memória para a separação dos segmentos do vetor.

Quanto a seu funcionamento, conceitualmente o método da inserção pode ser entendido como “a classificação de um conjunto de registros que é feita inserindo registros num sub-arquivo classificado anteriormente, ou seja, a inserção de um elemento é feita na posição correta dentro de um sub-arquivo classificado [...]”



(CEFETPB, 2006), ou ainda “[...] Os elementos são conceitualmente divididos em uma seqüência destino  $A_1...A_{i-1}$  e uma seqüência fonte  $A_i...A_n$ . Em cada passo, iniciando-se com  $i=2$  e incrementando-se  $i$  de uma em uma unidade, o  $i$ -ésimo elemento da seqüência vai sendo retirado e transferido para a seqüência destino, e inserido na posição apropriada [...]” (WIRTH, 1999) e [...] “... o vetor C é dividido em dois segmentos. Inicialmente, o primeiro segmento contém apenas o primeiro elemento, estando, portanto, classificado, enquanto o segundo segmento contém os elementos  $C[2], C[3], \dots, C[n]$ , sendo  $n$  o número de elementos do vetor. Por meio de iterações sucessivas cada elemento do segundo segmento é inserido ordenadamente no primeiro, até que todos os sejam (VELOSO, 1983).

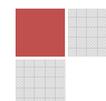
Todavia, apesar de fundamental o embasamento teórico, deixa clara a necessidade de uma visualização e comentário de uma implementação prática, uma vez que os conceitos pouco auxiliam na fixação e pleno entendimento dos conceitos do método da inserção direta.

#### **4 - IMPLEMENTAÇÃO DO MÉTODO CLASSIFICAÇÃO EM LINGUAGEM JAVA**

A aplicação procura demonstrar o funcionamento dos algoritmos de classificação, enfocando o método da inserção direta, apresentando suas vantagens, como seu desempenho superior aos demais algoritmos de classificação denominados básicos (PASSARO, 2006) e limitações como necessidade de pré-requisitos (vetor relativamente pequeno, ordenação prévia). Entretanto sua aplicabilidade e funcionamento, não são visíveis, apesar de fundamentais para qualquer vertente no campo da computação, pois qualquer estrutura levemente ordenada otimiza o funcionamento do processo em questão.

O desenvolvimento da aplicação fundamentou-se, na criação de duas classes, sendo a primeira exclusiva para descrever o método da inserção direta e a classe principal, assim denominada, por ser vista e utilizada pelo usuário e que tem como essência funcional à utilização do algoritmo do método de inserção direta.

Pode-se ver na Figura 1 o algoritmo desenvolvido como objeto deste trabalho na linguagem Java:



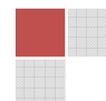
```
1 class insercao{
2 public static void insert (int vet [])
3 {
4
5     for (int j = 1; j < vet.length; j++){
6
7         int i;
8         int aux;
9         aux = vet[j];
10        i = j - 1;
11
12        while (i >= 0 && vet[i] > aux){
13
14            vet[i + 1] = vet[i];
15            i = i - 1;
16        }
17        vet[i + 1] = aux;
18    }
19 }
20 }
```

Figura1-Código da Aplicação em Java, Algoritmo de inserção

Esta classe inserção baseia-se numa ordenação simples, que se inicia com a declaração da classe inserção (linha 1), posteriormente o método insert (linha 2), onde o laço (linhas 5 a 16), que se inicia na segunda posição do vetor utilizando um indexador auxiliar, até a sua posição final, com o indexador auxiliar acrescido de um, e ocorre a declaração das variáveis a utilizar (linhas 7 e 8). A seguir, uma variável auxiliar recebe o vetor na posição inicial do laço (linha 9), e há a inicialização do vetor na primeira posição, já utilizando o indexador principal do método (linha 10).

Assim, condiciona-se a organização à que, o indexador seja maior ou igual à posição zero, e que o vetor nesta posição (indexador) seja maior que a variável auxiliar, previamente definida (linha 12). Desta forma, ocorre o deslocamento, onde o vetor na posição indexada na condição incrementado de um, receberá a variável na posição indexada (linha 14) e o decréscimo do indexador, ateia-se à necessidade de ordenação total do vetor, atrelando-se a condição (linha 15). Posteriormente, o que possibilita a ordenação total do vetor, à saída plena ou não cumprimento da condição, ocorre à inserção, com o vetor na posição indexada incrementado de um recebendo a variável auxiliar (linha 17).

Entretanto, a classe acima, por si, não terá utilidade sem uma complementação estrutural, uma vez que necessita de números a serem ordenados. Assim, faz-se necessário que uma classe principal faça a chamada do método de classificação anteriormente descrito, como tradicionalmente ocorre em programas escritos na linguagem Java.



## 5 - CONCLUSÃO

Os conceitos fundamentais da estrutura de dados usando os algoritmos de classificação apresentam-se necessários, apesar de não possuírem funções explícitas na construção de sistemas mais complexos, pois qualquer processo ou estrutura é notoriamente mais simples de ser executada ou processada, se seus elementos estiverem ordenados.

O método da inserção direta, apresenta-se, dentre os métodos considerados simples (seleção direta, bubblesort e inserção direta), como comprovadamente o mais rápido (dentro de alguns requisitos, como tamanho de vetor reduzido, vetor relativamente ordenado). Desta forma, como somente sua definição conceitual pouco auxilia no entendimento do método entre lógica e aplicação, surgiu a necessidade de implementação, detalhada e prática do algoritmo, que foi fundamentado teoricamente em algoritmos em linguagens Pascal, C e pseudocódigos, devido a didática que estes métodos possibilitam além de sua similaridade com a linguagem Java, utilizada para a implementação prática, devido a sua grande aceitação e tendências mundiais, além de ser uma linguagem de propósito geral, para o meio acadêmico e com forte aceitação no mercado.

## 6 - REFERÊNCIAS BIBLIOGRÁFICAS

CARDOSO, A.P. **Métodos de Ordenação**. Disponível em:

<http://www.inf.unisinos.br/~anibal/prog2ordena.pdf>. Acessado em: 15/08/2006

CCET-VIRTUAL. **Vetor**. Disponível em: <http://ccet.ucs.br/dein/napro/java/> Acesso em: 18 ago. 2006.

CEFETPB. **Classificação de Dados-Métodos de Classificação Interna**. Disponível em: <http://arquivos.coinfo.cefetpb.edu.br/~fred/>. Acessado em: 18 ago. 2006.

GONÇALVES, R. A. L. **Algoritmos e Estruturas de Dados I- Técnicas de Programação Usando a Linguagem Pascal**. Disponível em:

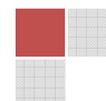
<http://www.din.uem.br/~ronaldo/LivroAED-Capitulos-1-2-3-IntrodPascal.pdf>.

Acessado em 14 set. 2006.

PÁSSARO, Â. **Projeto e Análise de Algoritmos. Algoritmos de Ordenação**.

Disponível em: [http://www2.brazcubas.br/professores1/arquivos/12\\_angelopassaro](http://www2.brazcubas.br/professores1/arquivos/12_angelopassaro).

Acesso em :18 ago. 2006.



VELOSO, P. **Estruturas de Dados**. Editora Campus, 15ª Edição, 228 p. 1983.

WIRTH, N. **Algoritmos e estruturas de dados**. Tradução de Cheng Mei Lee. Livros técnicos e científicos, Editora S.A. Rio de Janeiro, 1999, 255p.

