



FUNDAÇÃO UNIVERSIDADE FEDERAL DO PAMPA  
ENGENHARIA DE SOFTWARE  
CAMPUS ALEGRETE

Cassio Ceolin Junior,  
Daniel Oliveira de Freitas,  
Emanuelle Leães Alves,  
Everton Levi.

## RELATÓRIO DE ESTRUTURA DE DADOS

ALEGRETE

2021

# Sumário

## Sumário

1 Introdução.....	2
2 Desenvolvimento .....	3
2.1 Introdução à estrutura de dados .....	3
2.2 Tipo de dados.....	3
2.3 Tipo de dados primitivos .....	4
2.4 Abstração de dados .....	5
2.5 Tipo Abstrato de Dados .....	5
2.6 Como os TAD são empregados .....	7
2.7 Problema real .....	7
3 Considerações Finais .....	8
4 Referências .....	8

# 1 Introdução

Estrutura de dados é o ramo da computação que estuda os diversos mecanismos de organização de dados para atender aos diferentes requisitos de processamento. As estruturas de dados definem a organização, métodos de acesso e opções de processamento para a informação manipulada pelo programa, também é uma maneira de armazenar e relacionar informações, sendo na maioria das vezes sequencial mas tendo grande importância para se armazenar quando necessário um conjunto de dados em um software.

Existem diversas estruturas de dados utilizados pela programação, mas as principais são vetores, pilhas, listas, filas e pilhas sendo largamente utilizadas na implementação de aplicações.

Segundo Preiss (2000, p. 83): A abstração é “a ideia de uma qualidade ou propriedade independente de qualquer objeto ou coisa real em particular que tenha tal qualidade ou propriedade”. Por exemplo, podemos pensar no tamanho de um objeto sem conhecê-lo. Da mesma forma, podemos pensar sobre o modo de dirigir um carro sem conhecermos a marca ou o modelo. A abstração é usada para suprimir os detalhes irrelevantes e, ao mesmo tempo, enfatizar os relevantes. A abstração facilita ao programador pensar sobre o problema que ele tem para resolver.

Ainda, de acordo com Preiss (2000, p. 84): No projeto de programas orientados a objetos, uma das principais preocupações do programador é desenvolver um conjunto de abstrações apropriado para a aplicação que ele tem em mãos, e então definir tipos abstratos de dados adequados para representar tais abstrações. Ao fazer isso, o programador deve estar consciente do fato de que definir um tipo abstrato de dado requer tanto a especificação de um conjunto de valores como a de um conjunto de operações para tais valores.

## 2 Desenvolvimento

### 2.1 Introdução à estrutura de dados

"De uma forma simples, uma estrutura de dados é uma forma sistemática de organizar e acessar dados, e um algoritmo é um procedimento passo a passo para executar alguma tarefa em tempo finito." (GOODRICH; TAMASSIA, 2007, p. 156)

As principais estruturas de dados são:

- Vetores (arrays);
- Pilha;
- Listas;
- Filas;
- Outras estruturas.

Computadores são máquinas que manipulam dados e informações. A computação abrange o estudo da forma como as informações são organizadas, manipuladas e utilizadas em um computador.

A estrutura de dados é o ramo da computação que estuda os diversos mecanismos de organização de dados para atender aos diferentes requisitos de processamento. As estruturas de dados definem a organização, métodos de acesso e opções de processamento para a informação manipulada pelo programa.

Uma estrutura de dados pode ser dividida em dois pilares fundamentais:

- **Dado:** elemento que possui valor agregado e que pode ser utilizado para solucionar problemas computacionais. Os dados possuem tipos específicos.
- **Elemento:** Elemento estrutural que é responsável por carregar as informações dentro de uma estrutura de software.

### 2.2 Tipo de dados

"The data type of a variable is the set of values that the variable may assume." (AHO et al., 1983, p. 13)

Dentro da programação existem 4 tipos de dados primitivos, sendo eles inteiros que recebem números inteiros sem casa decimais, reais podendo receber números

com pontos flutuantes, texto que recebe caracteres de texto entre aspas dupla ou simples e lógico que recebe uma variável booleana podendo ser verdadeiro ou falso.

De modo geral os 4 tipos de dados primitivos são:

- **Tipos Inteiro** - Capaz de armazenar valores inteiros como por exemplo 1, 2;
- **Tipos Real** - Pode se armazenar valores negativos ou positivos contendo pontos flutuantes;
- **Tipo Texto** - Usado para se armazenar caracteres de texto, geralmente em aspas duplas;
- **Tipo Lógico** - Usado para armazenar valor lógico podendo ser verdadeiro ou falso.

Algumas linguagens podem subdividir esses tipos de dados em mais categorias de acordo da necessidade de alocar mais capacidade de memória para a variável.

“Num programa, áreas de memória para armazenamento de dados são representadas por variáveis. A forma como os bits numa variável são agrupados, interpretados e manipulados pelo computador é definida pelo seu tipo de dados.” (PEREIRA, 2008, p. 17)

## 2.3 Tipo de dados primitivos

Algumas linguagens de programação dividem esses tipos de dados em mais variações, como por exemplo no java o tipo de dado inteiro possui quatro tipos de variação, sendo dividido em 4 tipos primitivos como byte, short, int e long. Cada um deles possui uma capacidade de armazenamento diferente.

A linguagem Java oferece os seguintes tipos básicos (também chamados de tipos primitivos) que não são objetos (GOODRICH e TAMASSIA, 2007, p. 26):

- **boolean**: Dado primitivo que armazena valor booleano como verdadeiro ou falso;
- **char**: Armazena caracteres unicodes de 16 bits;
- **byte**: Armazena números inteiros com sinal em complemento de dois de 8 bits, com uma capacidade de 0 a 255;
- **short**: Armazena números inteiros com sinal em complemento de dois em 16 bits, com uma capacidade de – 32768 até 32767;
- **int**: Armazena valores numéricos inteiros sem casas decimais. Inteiro com sinal em

complemento de dois em 32 bits;

- **long**: Armazena valores numéricos inteiros mas possuem mais espaço capacidade de armazenamento que o int, são inteiros com sinal em complemento de dois em 64 bits.
- **float**: Armazena valores com casas decimais, e suporta até 32 bits. Número de ponto flutuante de 32 bits (IEE 754-1985);
- **double**: Armazena valores com casas decimais mas tem maior capacidade que o float, suportando até 64 bits. Número de ponto flutuante de 64 bits (IEE 754-1985).

Os tipos de dados básicos variam de uma linguagem para outra.

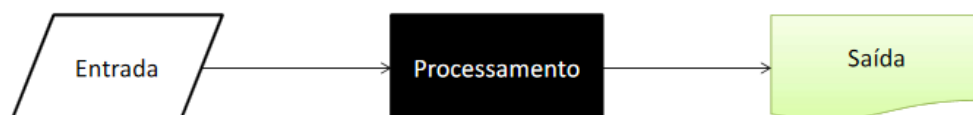
As regras para construir tipos de dados compostos a partir dos tipos básicos também variam.

## 2.4 Abstração de dados

Diz respeito à representação de características selecionadas de entidades do mundo real, necessárias à resolução de um determinado problema.

“Sempre que resolvemos um problema utilizando computador, a abstração de dados se faz necessária, já que não podemos ‘trabalhar’ dentro do computador diretamente com os objetos envolvidos no problema.” (PEREIRA, 2008, p. 15)

## 2.5 Tipo Abstrato de Dados



- **Entrada**: conjunto de dados colhidos do mundo real;
- **Processamento**: série de operações efetuadas sobre os dados de entrada;
- **Saída**: informação desejada.

“An abstract data type is a mathematical model, together with various operations defined on the model.” (AHO et al., 1983, p. 13)

“Um tipo de dado abstrato (TDA) é um modelo matemático definido por um conjunto de valores e por um conjunto de operadores que atuam sobre esses valores.” (PEREIRA, 2008, p. 16)

Algoritmos devem ser projetados a partir de TAD, que não levam em conta como os valores são armazenados na memória do computador, nem se preocupa com o tempo gasto para efetuar operações com esses valores. Os dados processados representam uma abstração da realidade.

Em Java, um TAD pode ser expresso por uma interface, ou seja, uma lista de declarações de métodos sem corpo.

#### Exemplo 1:

– Conjunto de inteiros, com operações de união, intersecção e diferença entre conjuntos.

#### Exemplo 2:

– Contador, que pode ser acessado, incrementado e decrementado.

#### Exemplo 3:

– Fila, onde o primeiro a entrar é o primeiro a sair.

#### Exemplo 4:

```
public interface IData {  
    /**Recebe três inteiros, dia, mês e ano, e inicializa a data.  
    * @param dia Informa o dia da data a ser inicializada.  
    * @param mes Informa o mês da data a ser inicializada.  
    * @param ano Informa o ano da data a ser inicializada.  
    * @return True caso a operação tenha sido efetuada e False caso  
    contrário.*/  
    public boolean inicializaData(int dia, int mes, int ano);  
    /**Soma um determinado número de dias a data.  
    * @param dias Informa o número de dias a ser somado.  
    * @return Retorna true caso a operação tenha sido efetivada ou false*/  
    public boolean acrescentaDias(int dias);  
    /**Retorna a data escrita por extenso. Por exemplo, 10/02/2121  
    * deve retornar uma String contendo 10 de fevereiro de 2121.  
    * @return A data por extenso ou null caso a operação não possa ser  
    efetuada */  
    public String extenso();  
}
```



## 2.6 Como os TAD são empregados

Em Java, um TAD é implementado por uma estrutura de dados concreta, que é modelada por uma classe. A classe de um TAD deve implementar sua interface e especificar como as operações são executadas.

Exemplo de como os TAD são empregados:

```
public class Data implements IData {  
    private int dia, mes, ano;  
    public boolean inicializaData(int dia, int mes, int ano) {  
        /* definição do método */  
    }  
    public boolean acrescentaDias(int dias) {  
        /* definição do método */  
    }  
    public String extenso() {  
        /* definição do método */  
    }  
}
```

## 2.7 Problema real

Estrutura de dados está em todos os pontos da computação e seu uso, quando usamos o “ctrl+z” estamos utilizando o conceito de pilha. Uma estrutura de dados correta pode fazer um sistema muito mais rápido, onde da mesma forma que uma estrutura implementada errada pode deixar o sistema lento. Dado isto, o conhecimento em estruturas de dados pode ser mais importante do que o conhecimento em uma linguagem de programação, pois ele possibilita o profissional criar códigos com máxima qualidade.

### **3 Considerações Finais**

Podemos concluir que este trabalho foi importante para se ter uma base maior sobre a estrutura de dados e tipos de dados e tudo que envolve eles, pois percebemos o uso das estruturas de dados nos sistemas que utilizamos diariamente. Também conhecer sobre outros tipos de dados primitivos e suas capacidades, também foi importante para conhecer outros tipos de estruturas como pilha, listas e filas. A união do conhecimento de dados primitivos com as estruturas fazem com que, ao escrever códigos em qualquer linguagem, seja possível pensar e codar de forma estruturada, otimizando o funcionamento dos sistemas.

### **4 Referências**

FURTADO DE OLIVEIRA ALVES, Gustavo. O que são tipos de dados primitivos?. Disponível em: <https://dicasdeprogramacao.com.br/tipos-de-dados-primitivos/>. Acesso em: 22 jun. 2021.

GOODRICH, M. T.; TAMASSIA, R. Estruturas de Dados e Algoritmos em Java. 4a ed., Bookman, 2007.

PEREIRA, S. L. Estruturas de Dados Fundamentais: conceitos e aplicações. 12ª ed. Érica, 2008.

AHO, A. V., HOPCROFT, J. E. Data Structures and Algorithms. Addison-Wesley, 1983.

PREISS, B. R. Estruturas de Dados e Algoritmos: padrões de projetos orientados a objetos com Java. Elsevier, 2000.