

Universidade Federal do Pampa - Unipampa
Conceitos básicos de Dados
Estruturas Dados AeG1

Rafael C. Ribeiro,
Bhruno R. Leifheit,
Thiago C. de M. Araújo,
Gustavo D. da Silva.
Docente: Williamson A. Freitas
Junho, 2021.

- **1. Introdução a Estrutura de Dados:**

Ao trabalharmos com computadores surge a necessidade de transcrever os dados de forma minuciosa e objetiva para que a máquina possa compreender e assim processá-los para que o sistema em questão funcione. A estrutura de dados é uma forma de armazenamento e organização que busca eficientizar o uso de dados.

Essa necessidade de transcrição pode ser realizada com diferentes tipos de operações como: inserir um novo item, excluir um item, localizar um elemento, percorrer todos os itens e classificar com diferentes tipos de ordem os dados.

Na estrutura de dados podemos dividi-la em dois segmentos, o Dado contendo os inteiros, textos, caracteres, booleanos, reais, e também temos o segmento da Estrutura contendo vetores, matrizes, pilhas, filas, listas (ALVES, 2013).

- **2. Tipos de dados e Tipos de dados primitivos:**

Os tipos de dados seriam uma representação de informações do mundo real para que possam ser interpretadas computacionalmente, assim diferentes linguagens podem criar e manipular esses diferentes dados para o desenvolvimento de softwares.

Para a criação e posterior manipulação de um determinado dado, primeiramente devemos levar em conta o hardware pelo qual estamos utilizando, pois ele determinará quais os tipos de dados primitivos que poderão ser processados na máquina. Parte dos dados são considerados primitivos, que seriam dados básicos fornecidos por uma linguagem de programação (GATTO, 2016).

Existem quatro tipos de dados primitivos, porém algumas linguagens podem acabar dividindo em mais, estes são o inteiro representando os valores negativos e positivos sem casa decimal, o real em que aborda números negativos e positivos com a casa decimal, o lógico/booleano que assume valores "TRUE" (verdadeiro) ou "FALSE" (falso) ou até mesmo representado com apenas bits sendo 1 ou 0 e por fim o carácter que configura um carácter, como exemplo na linguagem C, não são considerados primitivos o dado lógico/booleano e nem Strings.

Além disso, temos linguagens que alocam diferentes tipos de espaço necessário na memória, como o byte, short, int e long em que cada um conta com um armazenamento diferente (ALVES, 2013).

Figura 1: Diferentes Tipos primitivos de inteiros em java.

Tipos Inteiros:

Tipo	Memória consumida	Valor Mínimo	Valor Máximo
byte	1 byte	-128	127
short	2 byte	-32.768	32.767
int	4 bytes	-2.147.483.648	2.147.483.647
long	8 bytes	-9.223.372.036.854.775.808	9.223.372.036.854.775.807

Fonte: <<https://tableless.com.br/java-tipos-de-dados/>>, 2015.

Figura 2: Tipos primitivos de números reais em java.

Números que podem conter partes fracionárias podem ser representados por dois tipos:

Tipo	Memória Consumida	Valor mínimo	Valor máximo	Precisão
float	4 bytes	-3,4028E + 38	3,4028E + 38	6 – 7 dígitos
double	8 bytes	-1,7976E + 308	1,7976E + 308	15 dígitos

Fonte: <<https://tableless.com.br/java-tipos-de-dados/>>, 2015.

- **3. Tipos abstratos de dados:**

Os TAD (Tipos Abstratos de Dados) consistem na especificação de um conjunto de dados e nas operações que podem ser executadas nestes dados. Visto isso, existem duas características fundamentais proeminentes deste tipo de dados (Pessoa, 2014).

1. Os operadores ditam regras bem definidas para manipulação dos valores armazenados;
2. Valores armazenados devem ser manipulados apenas pelos operadores.

De acordo com Fortes(2014), quando fazemos o uso de um tipo abstrato de dados procuramos nos concentrar nas operações com determinado tipo de dado e nos abdicar da maneira como foi concebido.

Além disso contamos com a especificação sintática de um dado abstrato, porém pode não ser o suficiente para descrevermos o seu comportamento, para isso temos também a especificação da sua semântica que conta com uma definição de um conjunto de equações que têm relação com as operações com Dados abstratos, assim como as condições aplicadas utilizando expressões lógicas.

Uma boa prática de programação de TAD é a separação/modularização de arquivos em que geralmente são gerados um 'nome_tad.h' representando somente as declarações e um 'nome_tad.c' contendo as implementações dessas declarações e quem for utilizar seja outro TAD ou um programa basta usar um '#include nome_arquivo.h' (este exemplo seria na linguagem C) (FORTES, 2014).

Existem os mais variados tipos de estrutura utilizados na programação, sendo que os principais e mais utilizados são:

- Vetores: Estrutura linear e estática, o acesso aos dados é realizado de forma rápida por meio do índice do vetor. Esta estrutura tende a trazer complicações caso haja a necessidade de remover elementos sem deixar espaços vazios no vetor. Em vista disto ela é recomendada em casos de dados que não serão alterados (MOTA, 2009).
 - Lista: É uma estrutura de dados abstrata linear também chamada de encadeada que 'aponta' para nós que indicam o próximo elemento da lista, com exceção do último que não 'aponta' para ninguém (MOTA, 2009).
 - Fila: A Fila utiliza do princípio do FIFO (first in, first out / primeiro a entrar, primeiro a sair) sendo um algoritmo de escalonamento em que o primeiro dado que é removido, é o primeiro dado que foi inserido. Possui as funções ENQUEUE que adiciona um dado ao final de sua fila e o DEQUEUE que remove o dado no início de sua fila (MOTA, 2009).
 - Pilha: De acordo com Mota(2009) pilha são estruturas que fazem o uso do princípio LIFO (last in, first out/ último a entrar, primeiro a sair), na qual os dados que foram inseridos por último serão os primeiros a serem removidos. As funções PUSH que bota um dado no topo da linha e o POP que remove o dado do topo da linha.
 - Árvore: A árvore tem como principal característica a hierarquização dos dados, onde ela é dividida por raiz e nós. Há também a divisão de nós, na qual se diferem entre nós internos e externos (que também são conhecidos como nós folhas), os internos são aqueles que possuem 1 ou mais filhos/ramos, e os externos são aqueles que não possuem 0 ramos (UFES, 2011).
 - Grafo: Um grafo $G = (V, A)$ é constituído de um conjunto de vértices e um conjunto de arestas conectando pares de vértices e podemos implementar o tipo abstrato de dados "Grafo" basicamente de duas formas, Matriz de Adjacências e Lista de Adjacências (ROCHA, 2014).
 - Tabelas Hash: Também conhecidas como tabelas de dispersão, elas associam chaves e pesquisa de valores. Seu objetivo é realizar uma rápida busca para chegar ao valor desejado (TAUMATURGO, 2014).
- **4. Problema real (Como os tipos abstratos de dados são empregados)**

A concretização dos tipos abstratos de dados (TAD) pode ocorrer em duas fases bem definidas do sistema: uma responsável por determinar a estrutura dos dados e outra para indicar o conjunto de algoritmos.

Tratando-se da escolha de qual tipo abstrato de dados deve ser no caso de pilha ou fila, deve ser levado em consideração a definição, onde na fila o primeiro a entrar é o primeiro elemento a sair e na pilha o primeiro a entrar é o último elemento

a sair. Então, imaginando um contexto em que um software deve realizar o registro de espera, a implementação do tipo abstrato deve ser uma fila, pois a primeira pessoa na espera deve ser a primeira a ser atendida. Já pensando em uma situação que esteja relacionada com classificação por pontos com eliminatória, de ordem decrescente, e ocorresse as eliminações dos últimos colocados. A aplicação de árvores é realizada em casos em que os dados possuem naturalmente um certo tipo de hierarquia (GeeksforGeeks, 2019).

- **Referências:**

GATTO, Elaine. Tipos de dados para uso em algoritmos. **Embarcados**, 29 de junho de 2016. Disponível em: <<https://www.embarcados.com.br/tipos-de-dados/#:~:text=Os%20tipos%20de%20dados%20primitivos%20são%20os%20tipos%20básicos%20que,%2C%20booleanos%2C%20caracteres%20e%20strings>>.

ALVES, Gustavo. O que são tipos de dados primitivos?. **Dicas de Programação**, 2013. Disponível em: <https://dicasdeprogramacao.com.br/tipos-de-dados-primitivos/#disqus_thread>.

FORTES, Reinaldo. TAD: Tipo Abstrato de Dado. **Decom ufop**, 2014. Disponível em: <http://www.decom.ufop.br/anascimento/site_media/uploads/bcc202/aula_03_-_tipo_abstrato_de_dados.pdf>

MOTA, Kleber. Estrutura de dados - conceitos básicos. **Kleber's Blog**, 20 de agosto de 2009. Disponível em: <<https://klebermota.eti.br/2009/08/20/estrutura-de-dados-conceitos-basicos/#:~:text=Em%20Ciência%20da%20computação%2C%20uma,ser%20usados%20de%20modo%20eficiente.&text=A%20organização%20e%20os%20métodos,é%20que%20lhe%20conferem%20singularidade>>.

ESTRUTURA DE DADOS AULA 15: ÁRVORES. **UFES**, Espírito Santo, 17 de julho de 2011. Disponível em: <<http://www.inf.ufes.br/~pdcosta/ensino/2011-1-estruturas-de-dados-ec/slides/Aula15%20%28arvores%29.pdf>>.

ROCHA, Andreson. MC 202 – Estruturas de Dados - Grafos. **UNICAMP**, São Paulo, 2014. Disponível em <<https://www.ic.unicamp.br/~rocha/teaching/2014s1/mc202/aulas/aula-grafos.pdf>>

TAUMATURGO, Camila. Tabela Hash. **IFRN**, Rio Grande do Norte, 2014. Disponível: <<https://docente.ifrn.edu.br/camilataumaturgo/disciplinas/2014.2/estruturas-de-dados/tabela-hash/view>>.

PESSOA, Julie. Tipos de Dados Abstratos. **UFCG**, Paraíba, 2014. Disponível: <<http://www.dsc.ufcg.edu.br/~pet/jornal/maio2014/materias/recapitulando.html>>.

APPLICATIONS OF TREE DATA STRUCTURE. **GeeksforGeeks**, 05 de dezembro de 2019. Disponível em: <[https://www.geeksforgeeks.org/applications-of-tree-data-structure/#:~:text=Why%20Tree%3F,non-linear\)%20data%20structure.&text=If%20we%20organize%20keys%20in,List%20and%20slower%20than%20arrays\).](https://www.geeksforgeeks.org/applications-of-tree-data-structure/#:~:text=Why%20Tree%3F,non-linear)%20data%20structure.&text=If%20we%20organize%20keys%20in,List%20and%20slower%20than%20arrays).>)>.

