

Grupo 1 - Relatório - Conceitos Básicos de Dados

**Júlio Silveira Ortiz Rocha¹, Rafael A. Lezama¹,
Tadeu Brasil de Souza¹, William Christopher Ramos Oliveira¹**

¹Engenharia de Software – Universidade Federal do Pampa (UNIPAMPA)
Alegrete – RS – Brasil

O relatório foi dividido e desenvolvido da seguinte forma:

- **Introdução:** Rafael Lezama
- **Tipos de dados:** Rafael Lezama
- **Dados primitivos:** Júlio Silveira
- **Dados abstratos:** William Oliveira
- **Problema real:** Júlio Silveira
- **Revisão final:** Júlio Silveira

1. Introdução à Estrutura de Dados

Por trás de toda aplicação, há múltiplas abstrações compostas por conjuntos de ações e características que gerenciam e/ou manipulam dados, por exemplo, uma agenda que disponibiliza ações e nos ajuda a organizar múltiplos dados, do tipo criar, editar e persistir dados de um contato. Se a agenda for uma agenda de contatos, no permite executar estas operações.

Esta coleção de dados pode também ser chamada de estrutura de dados, porém, além de uma agenda genérica, existem outras estruturas de dados mais complexas que representam um arranjo de dados que nos auxiliam na organização de dados para melhor eficiência durante a execução, para um melhor acesso, tornando-o mais rápido, além de deixar o código mais amigável e simples.

Alguns arranjos existentes que serão vistos mais à frente são: vetores, arrays, listas, pilhas e filas, como os diferentes cenários de uso para cada um [Caelum 2019].

2. Tipos de Dados

Assim como no mundo real, dividimos de forma abstrata os dados, por exemplo, os números 1 e 1.0. Ambos representam o número um, mas possuem diferentes tipos. Um deles representa o valor inteiro, enquanto o outro seria um valor real. A mesma lógica se aplica aos dados em um computador. Tipos de dados não são nada mais do que uma abstração de valores que podem ser tomados, assim como as operações que podem ser feitas com esses dados/valores.

Os tipos de dados possuem as classificações (básico e estruturado). O exemplo dado de valores numéricos do tipo inteiro e real são chamados de tipo básico (ou de tipo de dado primitivo), por serem dados mais simples sem uma estrutura, diferentemente dos tipos de dados estruturados. Estes nos permitem a manipulação de diferentes valores em uma variável [Edelweiss and Galante 2009].

Uma estrutura pré-definida e planejada que não nos dá a mesma liberdade que temos com os tipos de dados definidos pelo usuário que possuem um conjunto de diversos

componentes, inclusive, podendo eles serem do tipo primitivo, que, quando juntos, tomam um único nome para representá-los. Essa estrutura permite manipular os dados de forma mais organizada [Edelweiss and Galante 2009].

3. Tipos de Dados Primitivos

Os tipos de dados primitivos, também conhecidos como "básicos", são dados indivisíveis. Eles não possuem estruturas definidas sobre os seus valores, impossibilitando a sua separação em valores menores. Variáveis do tipo lógico, por exemplo, assumem apenas os valores '*verdadeiro*' e '*falso*' [Edelweiss and Galante 2009].

Os dados primitivos são divididos em diversos tipos. De modo geral, são os seguintes:

- **Inteiro:** valores inteiros.
- **Lógico:** valores booleanos.
- **Real:** valores fracionários.
- **Caractere:** representa caracteres.

Dentro da programação, as diferentes linguagens dividem os dados primitivos, alocando-os o espaço necessário para a sua variável. Tal divisão é feita com o objetivo de otimizar ao máximo a memória e a sua utilização dentro do programa a ser desenvolvido [de Oliveira Alves].

4. Tipos Abstratos de Dados

Tipos abstratos de dados (TAD) são estruturas de dados cujo comportamento é definido por um conjunto de valores e operações [Chakraborty]. Esses valores e operações não foram previstos no núcleo das linguagens de programação e, portanto, são necessários para aplicações específicas [Edelweiss and Galante 2009]. Alguns exemplos de TAD são listas, filas e pilhas:

- A **lista** é uma coleção de itens em que cada valor pode aparecer mais de uma vez [Goodrich et al. 2014]. A depender da ordenação, ela pode ser classificada como fila ou pilha.
- Uma **fila** é uma coleção ordenada (lista com ordenação FIFO - *first in, first out*) de itens que possuem início (de onde são removidos primeiro) e fim (onde os itens são adicionados) [Miller and Ranum];
- Uma **pilha** é uma coleção ordenada de itens que são inseridos e removidos do seu "topo" (lista com ordenação LIFO - *last in, first out*) [Miller and Ranum].

5. Problema Real

No exemplo de problema real para a Avaliação em Grupo 1, decidimos mostrar como as **pilhas** são usadas para a navegação na internet.

Qualquer navegador básico possui um botão "voltar", cuja funcionalidade é retornar o usuário aos sites visitados anteriormente. Conforme o usuário navega através de páginas e mais páginas, estas são armazenadas dentro de uma pilha (no caso, o que é armazenado exatamente é o URL da página).

A página atual que o navegador exibe representa o topo da pilha, e a primeira página aberta ao iniciar a sessão de navegação seria a base. Ao clicar na função para

”voltar”, o usuário percorre a pilha de cima para baixo, até chegar no ponto inicial (a base).

As pilhas também são usadas nas operações ’desfazer’ presente em editores de textos e programas como Microsoft Office.

Referências

Caelum (2019). Cs-14 algoritmos e estruturas de dados em java. pages 1–4.

Chakraborty, A. Abstract data type in data structures. <https://www.tutorialspoint.com/abstract-data-type-in-data-structures>. Acesso: 24-06-2021.

de Oliveira Alves, G. F. (-). O que são tipos de dados primitivos? [Online; accessed 24-junho-2021].

Edelweiss, N. and Galante, R. (2009). *Estruturas de Dados: Volume 18*. Série Livros didáticos informática UFRGS. Grupo A - Bookman.

Goodrich, M., Tamassia, R., and Goldwasser, M. (2014). *Data Structures and Algorithms in Java*. Wiley.

Miller, B. and Ranum, D. Problem solving with algorithms and data structures using python. <https://runestone.academy/runestone/books/published/pythonds/index.html>. Acesso: 24-06-2021.