

# Tipos Abstratos de Dados

Franciny Medeiros  
franciny@ufj.edu.br

**ICET**  
INSTITUTO DE CIÊNCIAS  
EXATAS E TECNOLÓGICAS



# Tipos de dados

- Uma aplicação em Ciência da Computação é basicamente um programa de computador que manipula dados.
  - A representação dos dados manipulados por uma aplicação pode ser feita por diferentes estruturas de dados.
- É essencial que as linguagens de programação tenham suporte a uma variedade de tipos e estruturas.
  - Os termos “tipos de dados” e “estruturas de dados” têm significados diferentes.

- Um **tipo de dado** consiste da definição do conjunto de valores (domínio) que uma variável pode assumir ao longo da execução de um programa, e de seu conjunto de operações.

**Por exemplo:**

- O tipo **inteiro** pode assumir **valores** inteiros (negativos, zero, positivos):  $\langle \dots, -2, -1, 0, 1, 2, \dots \rangle$ .
- As **operações** possíveis são: soma, subtração, multiplicação, divisão, entre outras.

- As principais linguagens de programação oferecem uma grande variedade de tipos de dados, classificados em **básicos** e **estruturados**.
  - **Básicos** (primitivos): não possuem uma estrutura sobre seus valores, ou seja, não é possível decompor o tipo primitivo em partes menores. São indivisíveis. Por exemplo: inteiro, real, lógico e caractere.
  - **Estruturados**: permitem agregar mais do que um valor em uma variável, existindo uma relação estrutural entre seus elementos. Por exemplo: arranjos (vetores e matrizes), registros, sequências e conjuntos.

- Existem ainda os **tipos de dados definidos pelo usuário**.
- É constituído por um conjunto de componentes, que podem ser de tipos diferentes, agrupados sob um único nome.
- Os elementos desse tipo estruturado têm alguma relação semântica.

### Por exemplo:

- Tipo “Aluno”, que pode conter os componentes:
  - Inteiro: matrícula
  - Caractere: nome
  - Real: nota

- Dentro desse contexto surgem as **estruturas de dados** que especificam conceitualmente os dados, de forma a refletir um relacionamento lógico entre os dados e o domínio de problema considerado.
- Esse nível conceitual de abstração **não** é fornecido diretamente pelas linguagens de programação.
  - As linguagens fornecem os tipos de dados e os operadores que permitem a construção de uma estrutura de dados flexível para o problema que está sendo definido.

# Tipos Abstratos de Dados (TAD)

- Os TADs são estruturas de dados capazes de representar os tipos de dados que não foram previstos pelas linguagens de programação, mas que são necessários para aplicações específicas.
- Essas estruturas são divididas em **duas** partes:
  - Dados.
  - Operações.

- Um TAD é, portanto, uma forma de definir um novo tipo de dado juntamente com as operações que manipulam esse novo tipo de dado.

## Por exemplo:

### → Registro “data”:

- Inteiro: dia.
- Inteiro: mês.
- Inteiro: ano.

### → Funções:

- Inicializa data.
- Acrescenta dias.
- Escreve por extenso.



- A característica essencial de um TAD é separar o conceito da implementação.
- Os TADs são geralmente implementados em linguagens de programação através do conceito de **bibliotecas**.
  - A implementação das operações propriamente dita é feita em um arquivo separado.
  - A forma de implementação das operações não deve alterar o TAD.

- Resumindo:
  - um TAD especifica um **conjunto de operações** e a **semântica** dessas operações (o que elas fazem). Essa especificação é denominada **tipo abstrato**.
  - A implementação propriamente dita das operações é denominada **tipo concreto**.
  - As aplicações que utilizam esse TAD são denominadas **clientes do tipo de dado**.
- Duas importantes **vantagens** são:
  - a possibilidade de sua utilização em diversas aplicações diferentes.
  - A possibilidade de alterar o tipo sem alterar as aplicações que o utilizam.

- O projeto de um TAD envolve a escolha de operações adequadas. Algumas dicas são:
  - **definir um número pequeno de operações**, com soluções simples, que combinadas possam realizar funções mais complexas.
  - **o conjunto de operações deve ser adequado** o suficiente para realizar as computações necessárias às aplicações que utilizarão o TAD.
  - **cada operação deve ter um propósito definido**, com um comportamento constante e coerente, sem muitos casos especiais e exceções.



# Atividades

Enviar as respostas no SIGAA  
até o fim da aula



Faça um programa em linguagem C, que implemente um tipo Conta. Esse tipo terá como valores um nome do titular, um número da conta e um saldo.

Implemente funções para manipular o tipo de dado criado, como uma função para depósito, consultar saldo e realizar uma retirada.

O seu programa deve permitir que o usuário informe os dados da conta e escolha as operações disponíveis.