

■ ■ série de livros didáticos informática ufrgs



bookman[®]
EMPRESA DO GRUPO ARTMED
www.bookman.com.br

.inf
INSTITUTO
DE INFORMÁTICA
UFRGS

18



Estruturas de Dados

Nina Edelweiss
Renata Galante

Conceitos Básicos

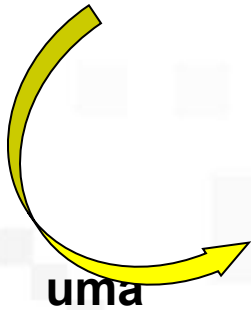
Introduz conceitos necessários ao entendimento das diferentes estruturas a serem vistas ao longo deste curso

Conceitos Básicos

**Tipos de dados e
Estruturas de dados**

Tipos de Dados

Tipo de dado



uma

**definição do conjunto de valores (domínio) que
variável pode assumir**

Ex: inteiro

**< ... -2, -1, 0, +1, +2, ...
>**

lógico

< verdadeiro, falso >

Tipos de Dados

- **Tipos básicos (primitivos)**
 - inteiro, real, lógico e caractere
- **Tipos de estruturados (construídos)**
 - arranjos (vetores e matrizes)
 - registros
 - sequências (conjuntos)
 - referências (ponteiros)
- **Tipos definidos pelo usuário**

Tipos e Estruturas de Dados

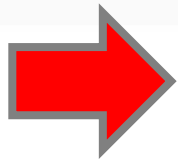
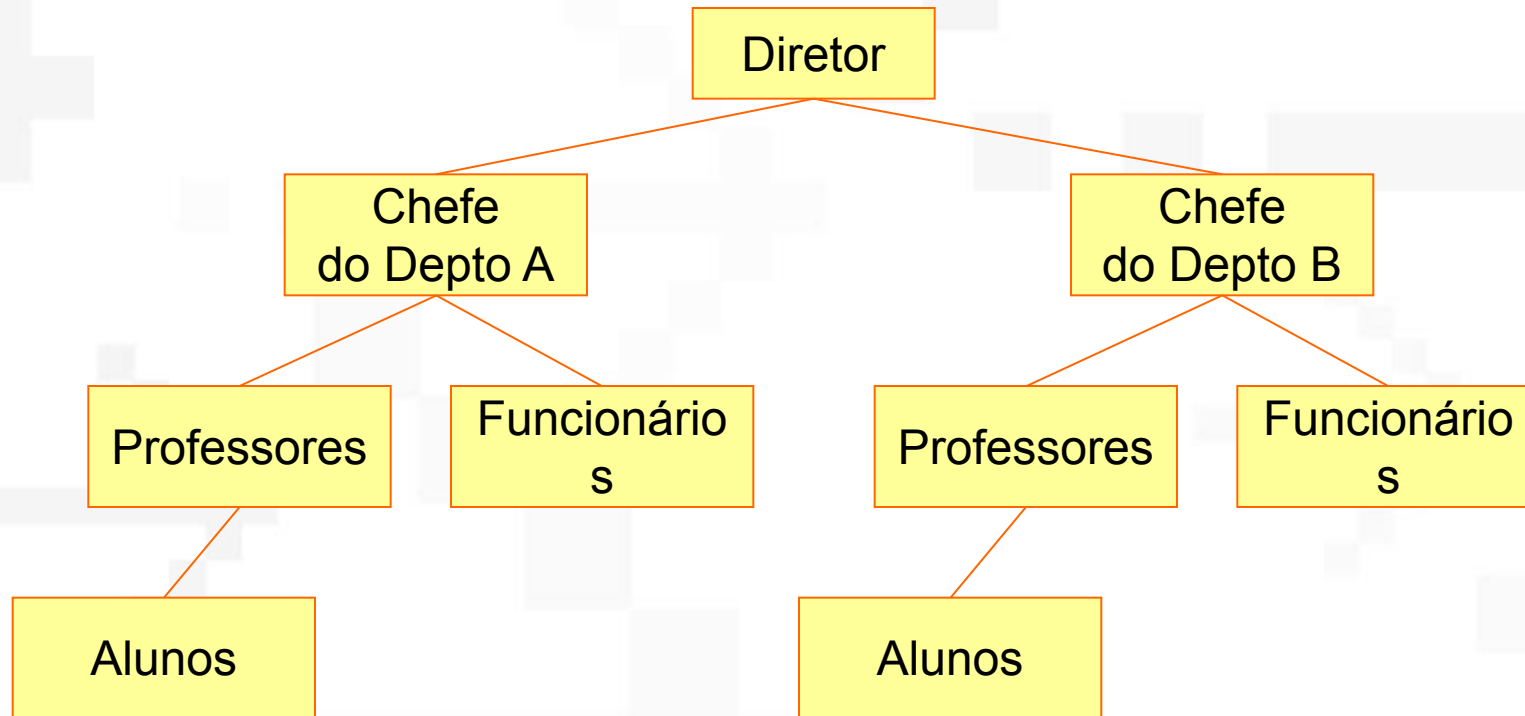
- **Aplicação:**

**representação dos funcionários e alunos
de uma Faculdade**

– Tipo de dado ? ? ? ? ?

Estruturas de Dados

Relacionamento de hierarquia



Como representar com os **tipos** de uma Linguagem de Programação?

Tipos e Estruturas de Dados

- **Tipos de dados básicos**
 - Fornecidos pela Linguagem de Programação
- **Estruturas de Dados**
 - Estruturação conceitual dos dados
 - Reflete um **relacionamento lógico** entre dados, de acordo com o problema considerado

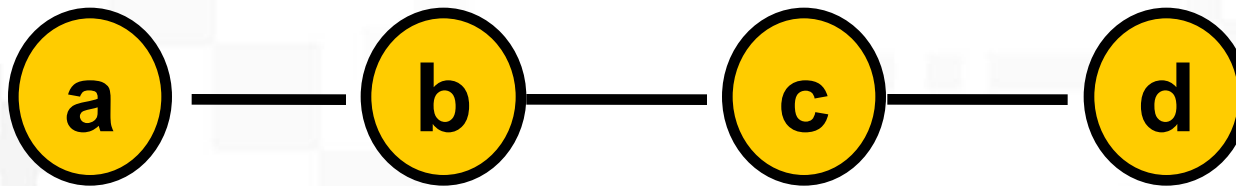
Estruturas de Dados

Estruturas de Dados a serem vistas:

- Listas lineares
- Árvores

Lista linear

- Relação de ordem entre os dados
- Linear - sequencial



Ex:

aplicação: empresa

problema: dados dos funcionários – cada nó um funcionário

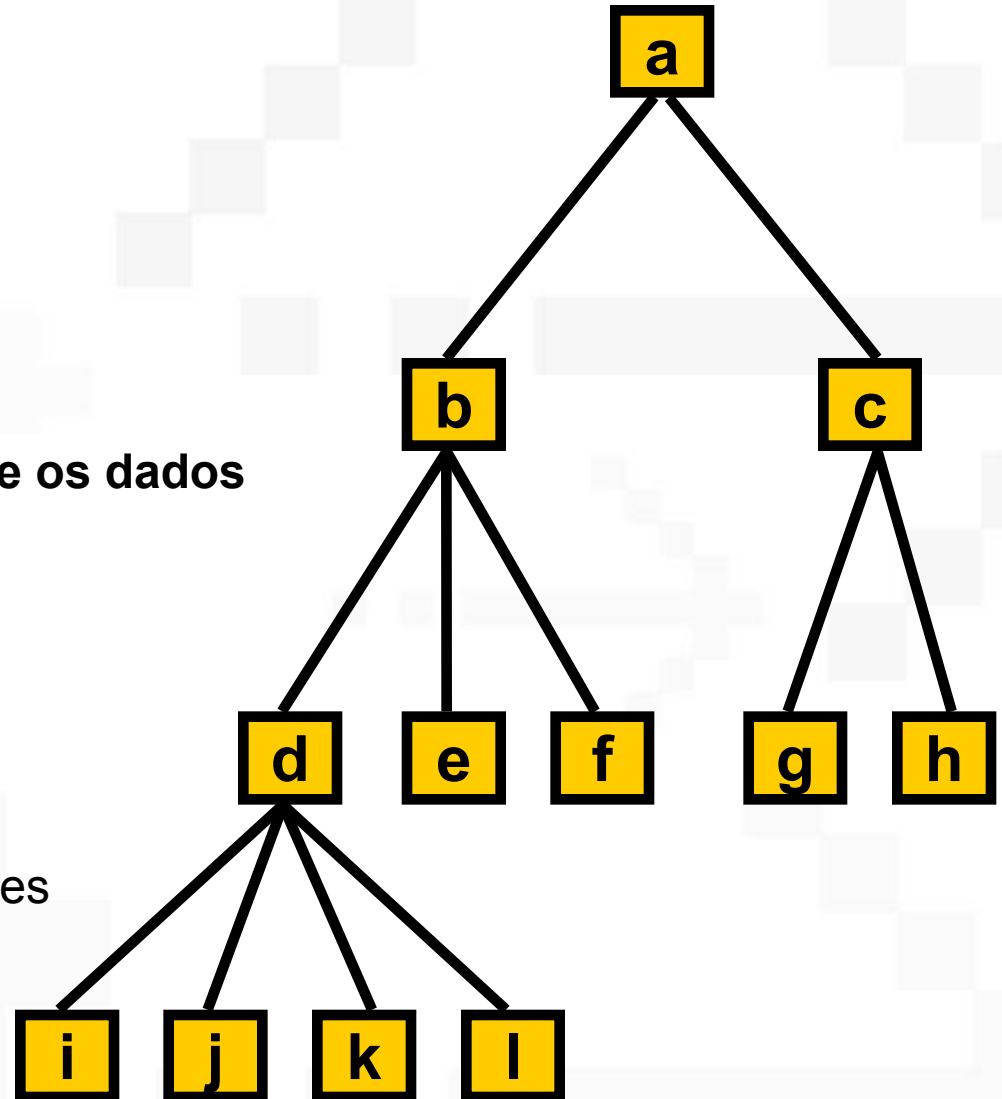
Árvore

- Relação de subordinação entre os dados

Ex:

aplicação: empresa

problema: organograma de funções



Operações sobre estruturas de dados

Estruturas de Dados incluem as **operações** para a manipulação de seus dados

Operações básicas:

- criação da estrutura de dados
- inclusão de um novo elemento
- remoção de um elemento
- acesso a um elemento
- destruição da estrutura de dados

Operações

Solução para uma Aplicação:

1. Identificar as *relações lógicas*

existentes entre os dados,

relevantes ao problema



**Ordem linear
pai / filhos**

...

2. Identificar as

operações

sobre estes dados



**Criação
Manutenção
Inserção de componente
Remoção de componente
Alteração de componente
Consulta
Destruição
Outras**

Exemplo: Folha de frequência

Disciplina: Estruturas de Dados Semestre: 2009-2 Turma: A Professor: Maria Silva				
matrícula	nome
XXXX	Ana			
ZZZZ	Maria			
YYYY	Pedro			

Programa: manipula dados dos alunos matriculados

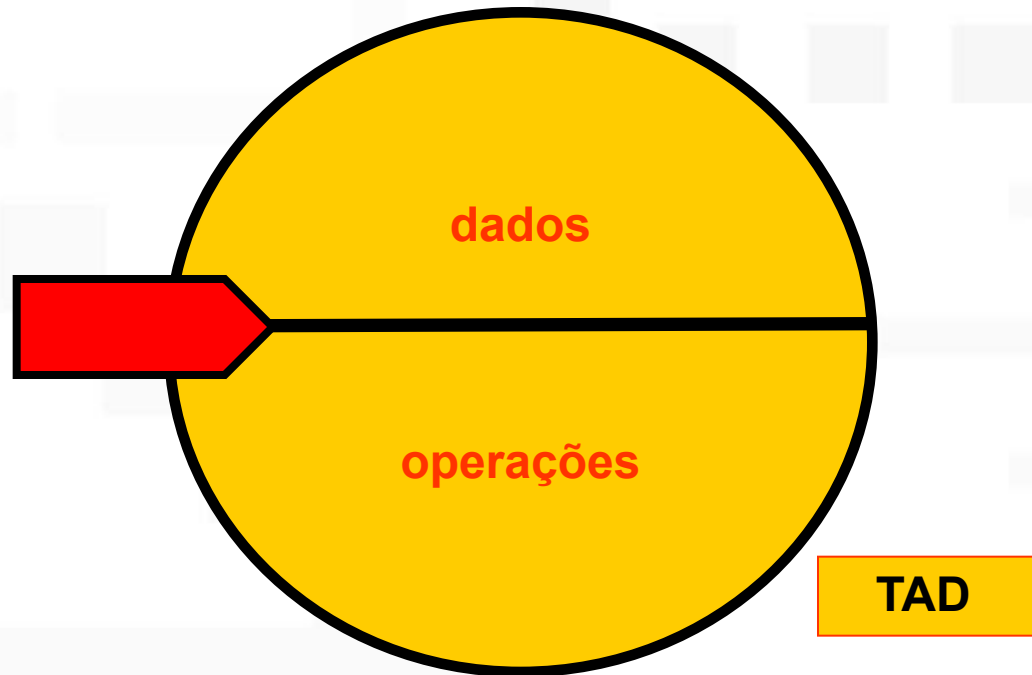
Operações sobre os dados dos alunos:

- inserir os nomes
- buscar os nomes dos alunos e armazenar na memória
- pesquisar os nomes ordenados para folha de frequência
- alterar os nomes
- excluir os nomes

Conceitos Básicos

TADs - Tipos Abstratos de Dados

Tipos Abstratos de Dados



TADs

Um **TAD** é uma forma de definir um **novo tipo** de dado
juntamente com as **operações** que manipulam esse
novo tipo de dado

TADs

- Separação entre conceito (definição do tipo) e implementação das operações
- Visibilidade da estrutura interna do tipo fica limitada às operações
- Aplicações que usam o TAD são denominadas *clientes* do tipo de dado
- Cliente tem acesso somente à forma abstrata do TAD

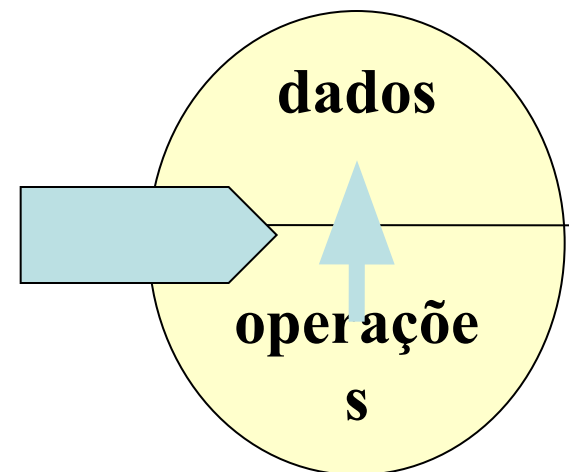
TADs

Um **TAD** (em LP) é um tipo de dado que satisfaz as condições:

- A representação ou a definição do tipo e as operações sobre variáveis desse tipo **estão contidas numa única unidade sintática**
 - MÓDULO
- A representação interna do tipo (a implementação) **não é visível de outras unidades** sintáticas, de modo que só as operações oferecidas na definição do tipo podem ser usadas com as variáveis desse tipo

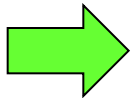
Propriedades dos TADs

- Satisfazem as propriedades de
 - **encapsulamento**: definição isolada de outras unidades do programa
 - **invisibilidade** e **proteção**: representação do tipo deve ser acessada somente no ambiente encapsulado
- A LP deve possibilitar
 - ambiente encapsulado
 - proteção de dados
 - interface para acesso
 - operações básicas



Vantagens de TADs

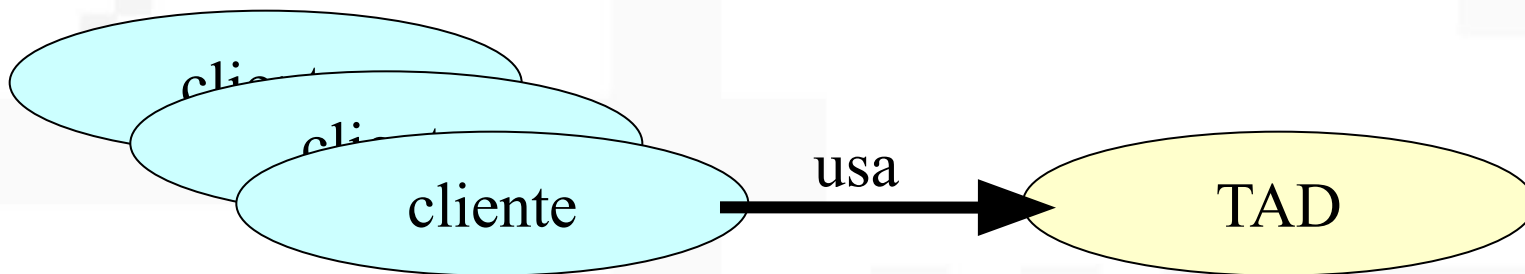
- **Possibilidade de utilização do mesmo TAD em diversas aplicações diferentes**
- **Possibilidade de alterar o TAD sem alterar as aplicações que o utilizam**



REUTILIZAÇÃO

Vantagens de TADs

- Código do cliente do TAD não depende da implementação
- **Segurança:**
 - clientes não podem alterar a representação
 - clientes não podem tornar os dados inconsistentes



Projeto de um TAD

Envolve a escolha de operações adequadas para uma determinada estrutura de dados, definindo seu comportamento

- **Dicas para definir um TAD:**
 - definir pequeno número de operações
 - conjunto de operações deve ser suficiente para realizar as computações necessárias às aplicações que utilizarem o TAD
 - cada operação deve ter um propósito bem definido, com comportamento constante e coerente

Exemplo de TAD: DATA

- **Par (v,o)**
 - **v – tripla formada por dia-mês-ano**
 - **o – operações aplicáveis sobre o tipo DATA**
 - **verificar se a data é válida**
 - **calcular o dia da semana de uma determinada data**
 - **calcular a data do Carnaval de um determinado ano**
 - **...**

Exemplo de TAD: DATA

- **Estrutura de representação**

Data = registro

Dia: inteiro

Mês: inteiro

Ano: inteiro

fim registro

Exemplo de TAD: DATA

- **Operações**

- Procedimento InicializaData

Entradas: Dia, Mês, Ano (inteiro)

Saída: D(Data)

- Função AcrescentaDias

Entradas: D(Data), Dias (inteiro)

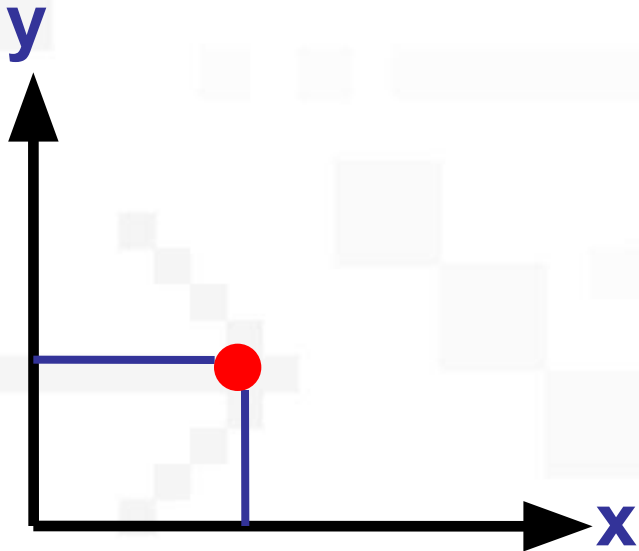
Retorno: (Data)

- Função EscreveExtenso

Entradas: D (Data)

Retorno: (lógico)

Exemplo de TAD: representação de um ponto



- **Modelo**

Par ordenado (x,y)

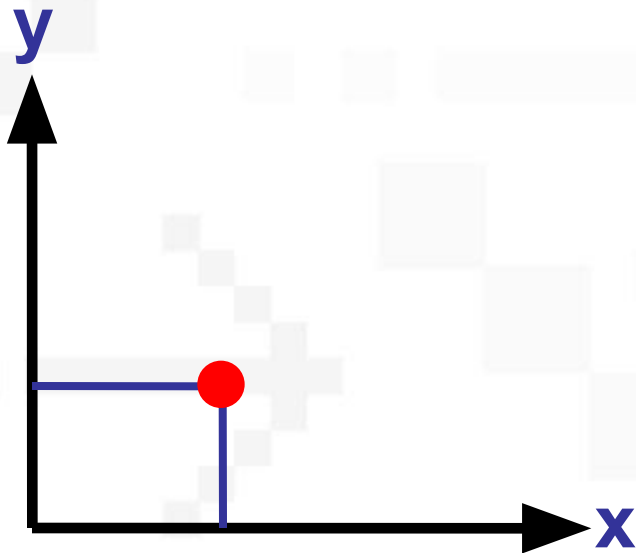
- **Dados representando o modelo**

- Coordenada X
- Coordenada Y

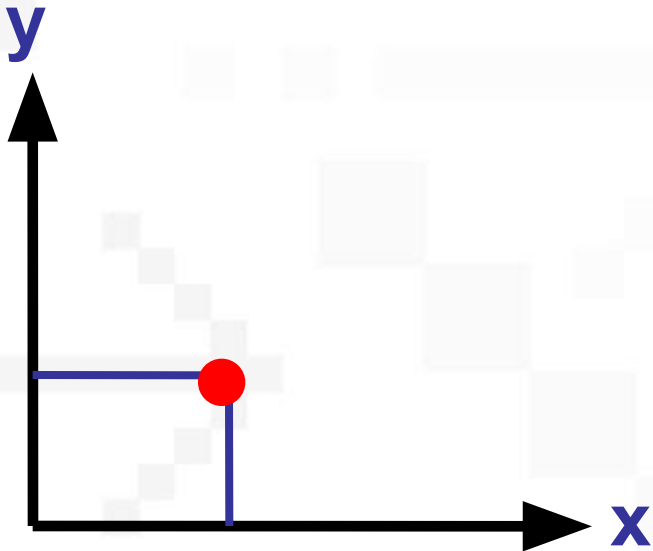
Exemplo de TAD: representação de um ponto

Operações:

- **cria**: operação que cria um ponto, alocando memória para as coordenadas x e y;
- **libera**: operação que libera a memória alocada por um ponto;
- **acessa**: operação que devolve as coordenadas de um ponto;
- **atribui**: operação que atribui novos valores às coordenadas de um ponto;
- **distancia**: operação que calcula a distância entre dois pontos.



Exemplo de TAD: representação de um ponto



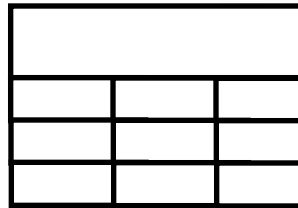
Operações:

- cria (x,y)
- libera (ponto P)
- acessa (ponto P)
- atribui (ponto P, x,y)
- distancia (ponto P1, ponto P2)

Conceitos Básicos

Alternativas de representação física

Modelagem dos dados



Identificação dos Dados
Abstração do mundo real



MODELO LÓGICO
Relacionamentos entre os dados
DADOS + OPERAÇÕES



MODELO FÍSICO
Alternativas de implementação
ALGORITMOS PARA AS OPERAÇÕES

Níveis de abstração

**Mundo
Real**

Objetos reais sem omissão de detalhes

Níveis de abstração

**Mundo
Real**

Objetos reais sem omissão de detalhes

**Informações relevantes
para o sistema**

**Selecionadas aquelas informações relevantes
para as aplicações consideradas
Depto, Emp, Veículo, etc.**

Níveis de abstração

**Mundo
Real**

Objetos reais sem omissão de detalhes

**Informações relevantes
para o sistema**

**Selecionadas aquelas informações relevantes para as
aplicações consideradas**

Depto, Emp, Veículo, etc.

**Tipos Abstratos de
Dados (TAD)**

**Componentes e operações dos diversos Tipos de Dados,
abstraídos detalhes e implementação:**

CADASTRO (insere, remove, ...)

Níveis de abstração

**Mundo
Real**

Objetos reais sem omissão de detalhes

**Informações relevantes
para o sistema**

**Selecionadas aquelas informações relevantes
para as aplicações consideradas
Depto, Emp, Veículo, etc.**

**Tipos Abstratos de
Dados (TAD)**

**Componentes e operações dos diversos Tipos
de Dados, abstraídos detalhes e
implementação:
CADASTRO (insere, remove, ...)**

**Estrutura Lógica
e Algoritmos**

**Esquemas de representação dos dados e
detalhamento das operações definidas sobre
tais esquemas**

Níveis de abstração

**Mundo
Real**

Objetos reais sem omissão de detalhes

**Informações relevantes
para o sistema**

**Selecionadas aquelas informações relevantes para as aplicações consideradas
Depto, Emp, Veículo, etc.**

**Tipos Abstratos de
Dados (TAD)**

**Componentes e operações dos diversos Tipos de Dados, abstraídos detalhes e implementação:
CADASTRO (insere, remove, ...)**

**Estrutura Lógica
e Algoritmos**

Esquemas de representação dos dados e detalhamento das operações definidas sobre tais esquemas

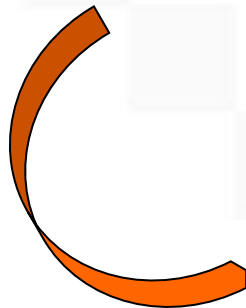
**Estrutura Física
e Programas**

Representação física dos dados e programas que implementam as operações (o nível físico pode corresponder a tipos primitivos de uma linguagem de programação)

Modelo físico

Escolha da representação física

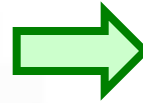
- Preservar as relações lógicas
- Permitir operações através de procedimentos simples e eficientes



As operações definidas sobre os dados influenciam decisivamente na escolha da representação física a ser adotada

Alternativas de Representação Física

Contiguidade física



posicional - implícita

Encadeamento



posições aleatórias - ordem explícita

A posição do componente na
estrutura lógica determina sua
posição na estrutura física

Modelo físico

- Diversas alternativas de estruturas físicas podem implementar uma mesma estrutura lógica
 - Vetores
 - Matrizes
 - Estruturas
 - Ponteiros

Modelo físico

- **A escolha pela melhor alternativa de estrutura física depende de alguns fatores**
 - **volume de dados**
 - **número (quantidade) de dados**
 - **fixo ou variável**
 - **operações realizadas sobre os dados**

Exemplo

- Folha de frequência

Disciplina: Estruturas de Dados Semestre: 2009-2 Turma: A Professor: Maria Silva				
matrícula	nome
XXXX	Ana			
ZZZZ	Maria			
YYYY	Pedro			

Exemplo

- **Folha de frequência**
 - **Modelo Lógico**
 - Estrutura abstrata para manter o relacionamento entre os dados

Exemplo

- Folha de frequência
 - **Modelo Lógico**
 - Estrutura abstrata para manter o relacionamento entre os dados: **LISTA**



Exemplo

- Folha de frequência
 - **Modelo Físico**
 - Implementa a estrutura lógica em uma estrutura de armazenamento (estrutura física) em uma linguagem de programação

Exemplo

- Folha de frequência
 - **Modelo Físico**
 - Implementa a estrutura lógica em uma estrutura de armazenamento (estrutura física) em uma linguagem de programação: **ARRAY**

vetor de alunos

	Ana	Maria	Pedro	
--	-----	-------	-------	--

Exemplo

- Folha de frequência

Disciplina: Estruturas de Dados Semestre: 2009-2 Turma: A Professor: Maria Silva				
matrícula	nome
XXXX	Ana			
ZZZZ	Maria			
YYYY	Pedro			

Operações:

- insere
- consulta
- excluir
- altera
- calculaMedia

Programa: manipula dados a respeito
dos alunos matriculados

Estruturas de Dados

Listas Lineares

Árvores

- **formas de estruturar os dados**
- **opções para armazenamento físico**
- **algoritmos de manipulação**

Final Capítulo 2

Conceitos básicos