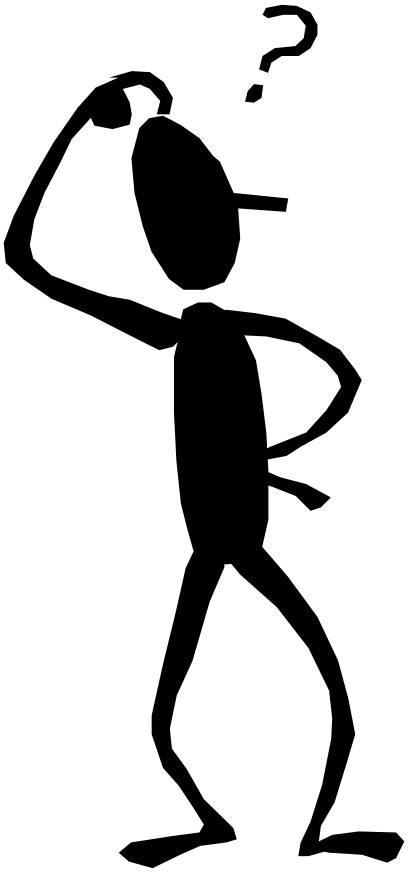
FAI - Centro de Ensino Superior

Curso: Sistemas de Informação
Disciplina: Estruturas de Dados
Unidade/assunto: Tipos de Dados

TIPOS DE DADOS



INTRODUÇÃO

- ✓ **O tipo de dado delimita o escopo da informação.**

- ✓ **Ao declararmos o tipo de uma variável, delimitamos:**
  - O Conjunto de valores que ela poderá armazenar;
  - As operações que poderemos realizar com ela;
  - A forma de representação destes valores;

TIPOS DE DADOS PRIMITIVOS

- São tipos básicos pré-definidos.
- São também chamados de atômicos ou simples.
- Ex: Integer,
  Real,
  Boolean,
  char,
  ponteiro.

INTEGER (INTEIRO)

- **Valores numéricos inteiros.**
- **Escopo:** -2 ^ (n-1) a 2 ^ (n-1) - 1

  Onde: n = número de bits ocupados na memória

  Ex: n = 16 $\Longrightarrow$ -32768 a 32767
- **Operações:** Adição(+), Subtração(-), Multiplicação (*), Divisão Inteira (DIV),

  Resto da Divisão (MOD).
- **Comparações:** Igualdade (=), Diferença ($\neq$),

  De Ordem (<, $\leq$, >, $\geq$)

REAL (REAL)

- **Valores numéricos com parte fracionária.**
- **Escopo:** negativos, zero ou positivos

  (pelo menos 4 bytes)
- **Operações:** Adição(+), Subtração(-), Multiplicação (*), Divisão (/).
- **Comparações:** Igualdade (=), Diferença (≠),

  De Ordem (<, ≤, >, ≥) .

BOOLEAN (LÓGICO)

- **Valores lógicos (verdadeiro ou falso).**
- **Escopo:** True ou False (ocupa 1 byte)
- **Operações:** Conjunção (E),
  Disjunção (OU),
  Negação (NÃO).
- **Comparações:** Igualdade (=), Diferença (≠)

CHAR (CARACTER)

- ✓ **Valores alfanuméricos representados entre aspas.**
- ✓ **Escopo:** Caracteres de 0 - 9, a - z, A - Z, sinais especiais (Ocupa 1 byte).
- ✓ **Comparações:** Igualdade (=), Diferença (≠).

POINTER (PONTEIRO)

- ✓ **Representa o endereço de um dado na memória.**

TIPOS DE DADOS ESTRUTURADOS

- ✓ **São também chamados de construídos ou não primitivos.**
- ✓ **São construídos a partir da composição de tipos primitivos.**
- ✓ **Esses tipos têm uma estrutura que define:**
  - Como os dados estarão organizados.
  - Como poderão ser acessados.

TIPOS DE DADOS ESTRUTURADOS

- ✓ **Estruturas mais utilizadas:**

  - **De alocação fixa de memória:**

    - Vetores, Matrizes, Registros, Strings.

  - **De alocação dinâmica de memória:**

    - Listas encadeadas, Pilhas, Filas, Árvores, Grafos, Arquivos.

MECANISMOS PARA CONSTRUÇÃO DE TIPOS

- **Vetor/Matriz:**
  - Valores agregados homogêneos;
  - Tamanho definido;
  - Definição: vet [lim_inf..lim_sup] de tipo.
- **Registro:**
  - Valores agregados heterogêneos;
  - Definição: reg (campo1: tipo1, campo2: tipo2,..., campoN: tipoN).

MECANISMOS PARA CONSTRUÇÃO DE TIPOS

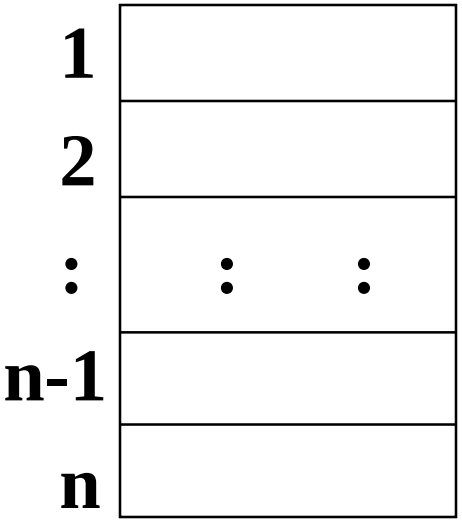
- **Sequência:**
  - Coleções ordenadas de dados;
  - Exemplo mais comum são as cadeias de caracteres (strings);
  - Definição: seq de tipo.
- **Alternativa:**
  - Em momentos diferentes uma variável pode ter valores de tipos diferentes;
  - Definição: alt (tipo1|tipo2|tipo3|....|tipoN).

MECANISMOS PARA CONSTRUÇÃO DE TIPOS

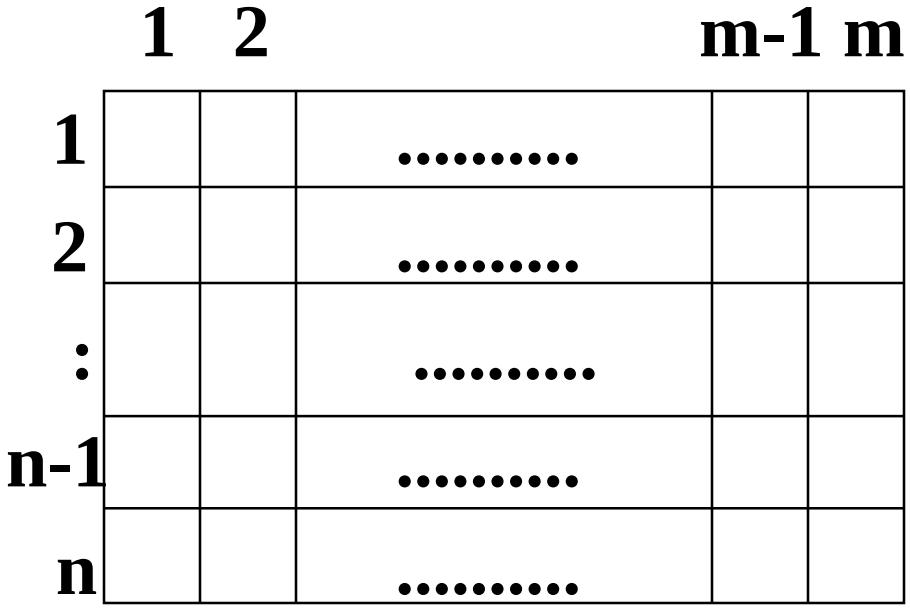
- **Referência:**
  - Permite a alocação dinâmica de memória;
  - Definição: ref a tipo.
- **Enumeração:**
  - Os valores dos dados definem o tipo;
  - Definição: (val1,val2,....,valN).

VETORES E MATRIZES

Vetor [n]



Matriz [n x m]



VETORES E MATRIZES

- ✓ Todos os elementos são do mesmo tipo.
- ✓ O tipo pode ser primitivo ou estruturado.
- ✓ São estruturas do tipo array.
- ✓ Podem ser unidimensionais, bidimensionais ou multidimensionais.
- ✓ Usa-se índice para acessar um componente individual.

VETORES E MATRIZES

- ✓ Erro comum na utilização de arrays é tentar acessa-la com um índice fora dos limites.
- ✓ A alocação de memória é feita no momento da declaração.
- ✓ A alocação de memória é contígua.
- ✓ Os limites de um array não podem ser modificados durante o processamento.
- ✓ Referência: vetor[i].

REGISTROS

Estudante

Nome	
Nome do Pai	
Nome da Mãe	
Telefone	Classe

REGISTROS

- ✓ Os elementos podem ser de tipos diferentes.
- ✓ Os elementos podem ser referenciados em conjunto (identificador do registro) ou individualmente.
- ✓ Um registro é composto de campos (elementos), cujos nomes são usados para identifica-los.
- ✓ O acesso a um elemento do registro é feito através do seu nome, e não através de índices como nos arrays.

REGISTROS

- Declaração: reg = registro (Declaração de tipo)

```
        campo1, campo2   : tipo1
        campo3           : tipo2
        .............
        campoN           : tipoN
    fim-registro
    var_reg : reg  (Declaração variável)
```

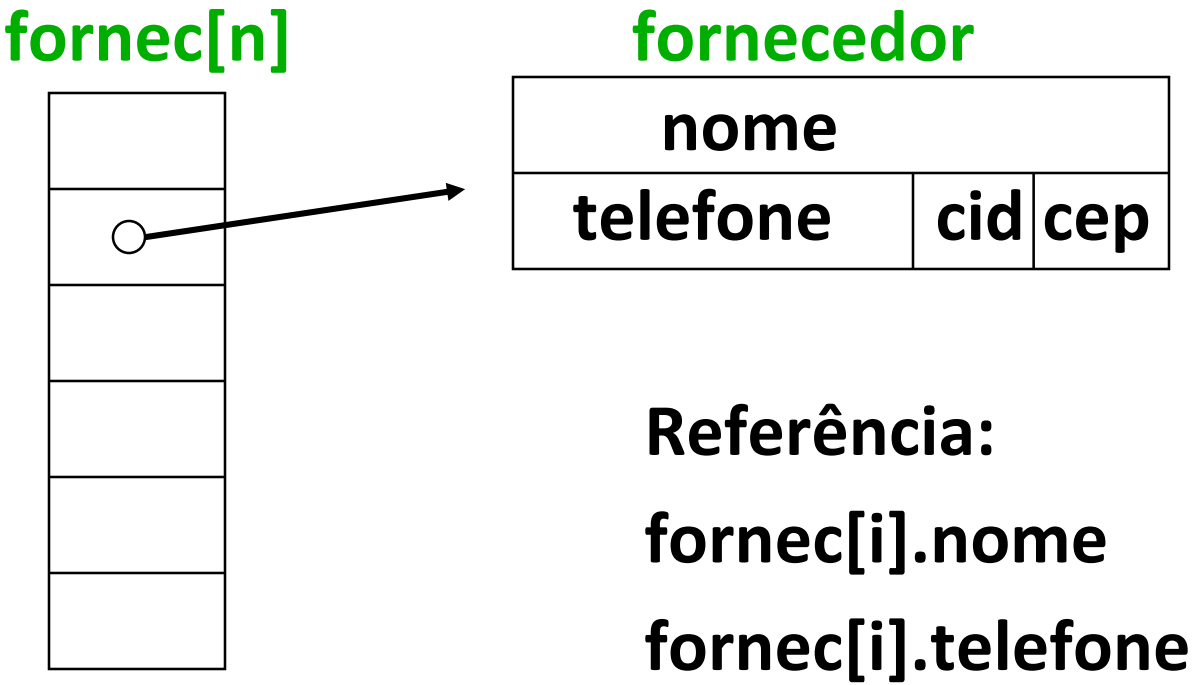
- Referência:

var_reg, var_reg.campo1, var_reg.campo2

ARRAYS DE REGISTROS

```
fornecedor = registro        (Declaração de Tipo)
                nome      : literal[30]
                tel       : literal[20]
                cidade    : literal[02]
                cep       : inteiro
             fim-registro
vet_for = vetor [1 .. N] de fornecedor
fornec : vet_for                (Declaração de Variável)
```

ARRAYS DE REGISTROS



Referência:

fornec[i].nome

fornec[i].telefone

fornec[i].cid

fornec[i].cep

REGISTROS DE ARRAYS

```
curso = registro        (Declaração de Tipo)
            professor  : vetor[1..N]
            disciplina  : vetor[1..M]
            semestres: inteiro
        fim-registro
cur : curso             (Declaração de Variável)
```

- Referência: cur.professor[i], cur.disciplina[i]
  cur.semestres

REGISTROS EMBUTIDOS

```
cursos = registro          (Declaração de Tipo)
                graduação       : literal[30]
                especialização  : literal[30]
                mestrado        : literal[30]
                doutorado       : literal[30]
          fim-registro
professor = registro
                nome            : literal[30]
                formação        : cursos
            fim-registro
ficha : professor          (Declaração de Variável)
```

- **Referência: ficha.nome, ficha.formação.graduação**

CADEIA DE CARACTERES (STRINGS)

- Sequência ordenada de elementos do tipo caracter.
- É representada entre aspas.
- Ex: “Santa Rita do Sapucaí”, “Brasil”.
- O computador registra a cadeia e seu comprimento.
- Cadeia vazia ou nula = ““.
- Assemelha-se aos vetores.

CADEIA DE CARACTERES (STRINGS)

- **Pode ter comprimento fixo ou variável:**

- Fixo:

I	N	F	O	R	M	A	T	I	C	A

- Variável:

B	R	A	S	I	L	\	U	S	A	\	C	U	B	A

6	B	R	A	S	I	L	3	U	S	A	4	C	U	B	A

- Ex: Pascal - Utiliza contador de caracteres.

C - Utiliza um terminador (sentinela) "\0".

CADEIA DE CARACTERES (STRINGS)

- ✓ **Operações:**
  - **Atribuição: A ← "Brasil", B ← "Penta"**
  - **Concatenação (+): C ← A + " é " + B ("Brasil é Penta")**
  - **Comparação:**

    Sendo: A = a1,a2,...,aN e B = b1,b2,...,bM

    A = B é verdadeiro se:

    1) N = M; 2) Ai = Bi p/ todo 1 <= i <= N

    A comparação é feita da esquerda p/ a direita.

CADEIA DE CARACTERES (STRINGS)

- **Ex:**

  A ← "MARIA", B ← "DARIO"

  A = B - Falso

  A <= B - Falso

  A < B - Falso

  A <> B - Verdadeiro

  A >= B - Verdadeiro

  A > B - Verdadeiro

CADEIA DE CARACTERES (STRINGS)

- Operações:

  - Tamanho (comprimento da cadeia) – len :

    A ← “Brasil”

    len (A) - 6

    len (“informática”) - 11

  - Localização de subcadeia – pos :

    Sintaxe: pos (cadeia, subcadeia)

    pos (“informática”, “forma”) - 3

CADEIA DE CARACTERES (STRINGS)

- **Operações:**

  - Subcadeia (Retorna parte da cadeia) - sub :

  Sintaxe: sub (A1, A2, A3), onde:

  A1 = cadeia, A2 = posição do caracter inicial,

  A3 = comprimento da subadeia a ser retornada

  Ex: sub (“informática”, 3, 5) - “formá”

  A ← “comemoração”

  sub (A, 7, 5) - “ração”