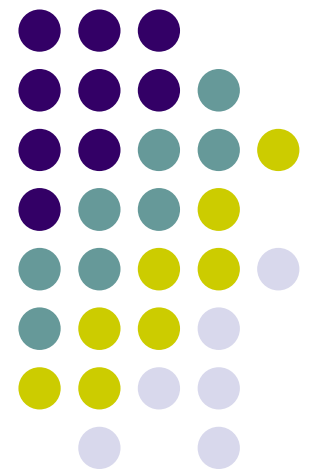


# Paradigmas de Linguagens de Programação

---

## 02: Tipos de Dados

Prof. Márcio Puntel  
[marcio.puntel@ulbra.edu.br](mailto:marcio.puntel@ulbra.edu.br)





# Conceituação

- Valor

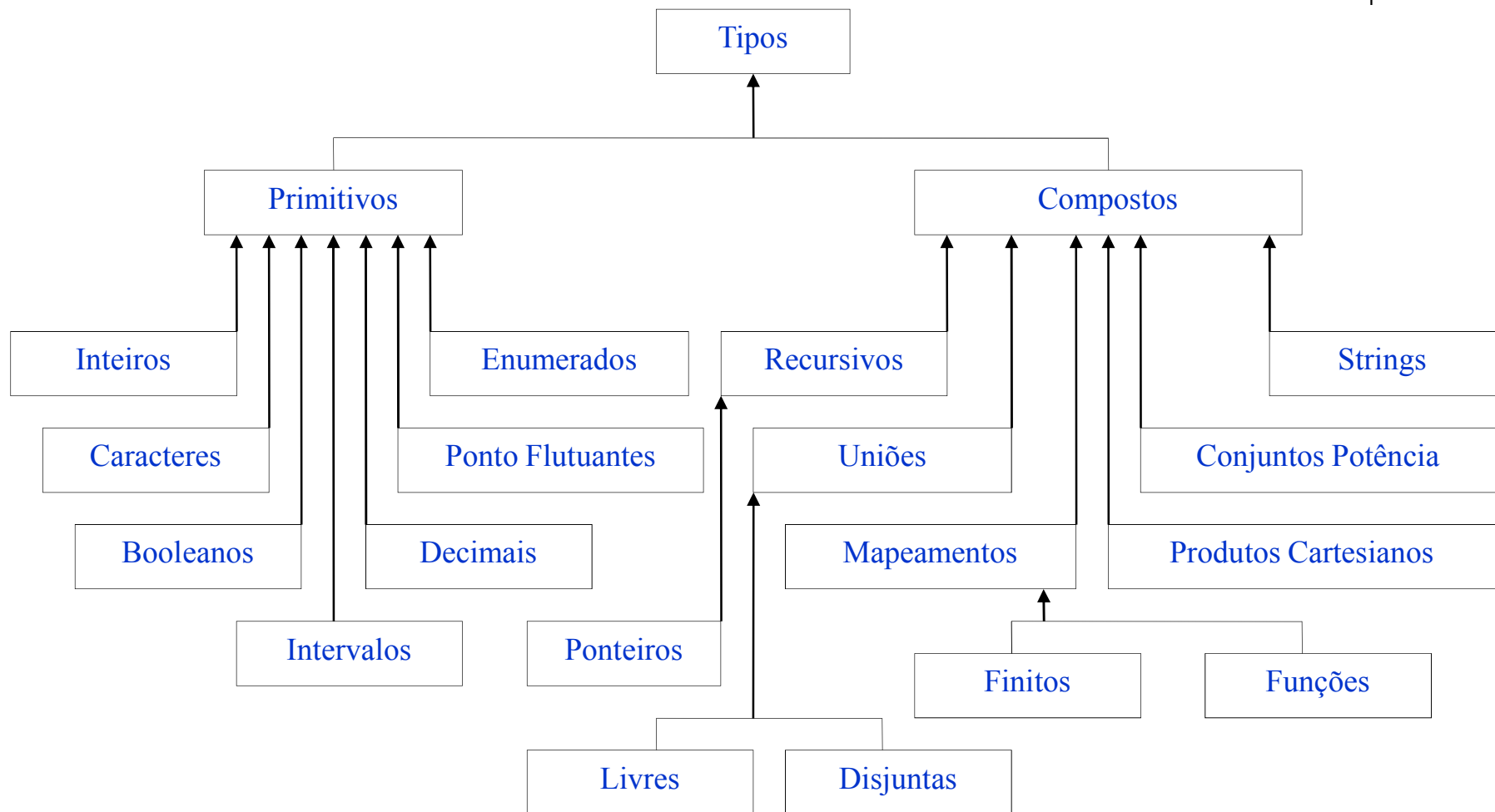
3    2.5    'a'    “Paulo”    0x1F    026

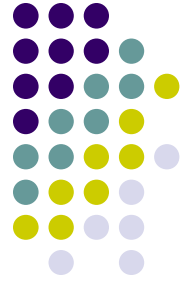
- Tipo

{true, 25, 'b', “azul” } não corresponde a um tipo

{ true, false } corresponde a um tipo

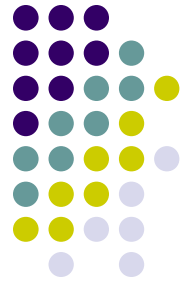
# Hierarquia de Tipos





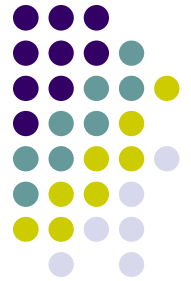
# Tipos Primitivos

- Não podem ser decompostos em valores mais simples
- Costumam ser definidos na implementação da LP
  - Sofrem influência direta do hardware



# Tipo Inteiro

- Corresponde a um intervalo do conjunto dos números inteiros
- Vários tipos inteiros numa mesma LP
  - Normalmente, intervalos são definidos na implementação do compilador
- Em JAVA, o intervalo de cada tipo inteiro é estabelecido na definição da própria LP



# Tipos Inteiros em JAVA

Tipo	Tamanho (bits)	Intervalo	
		Início	Fim
<i>byte</i>	8	-128	127
<i>short</i>	16	-32768	32767
<i>int</i>	32	-2.147.483.648	2.147.483.647
<i>long</i>	64	-9223372036854775808	9223372036854775807



# Tipo Caractere

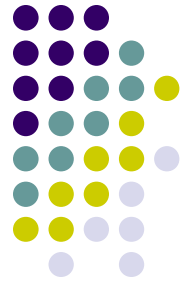
- Armazenados como códigos numéricos
  - Tabelas EBCDIC, ASCII e UNICODE
- PASCAL oferece o tipo *char*
- Em C, o tipo primitivo *char* é classificado como um tipo inteiro

```
char d;
```

```
d = 'a' + 3;
```

```
...
```

```
while (*p) *q++ = *p++;
```



# Tipo Booleano

- Tipo mais simples
  - Possui apenas dois valores
- C não possui o tipo de dado booleano, mas qualquer expressão numérica pode ser usada como condicional

Valores  $\neq$  zero  $\Rightarrow$  verdadeiro

Valores = zero  $\Rightarrow$  falso

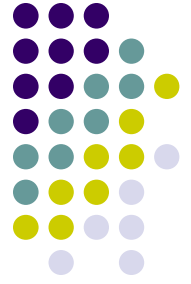
- Abordagem de C pode provocar erros

if (c += 1) x = 10;

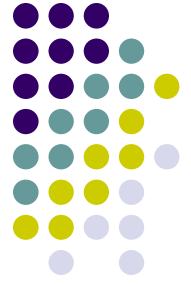
- JAVA inclui o tipo de dado *boolean*



# Pense...

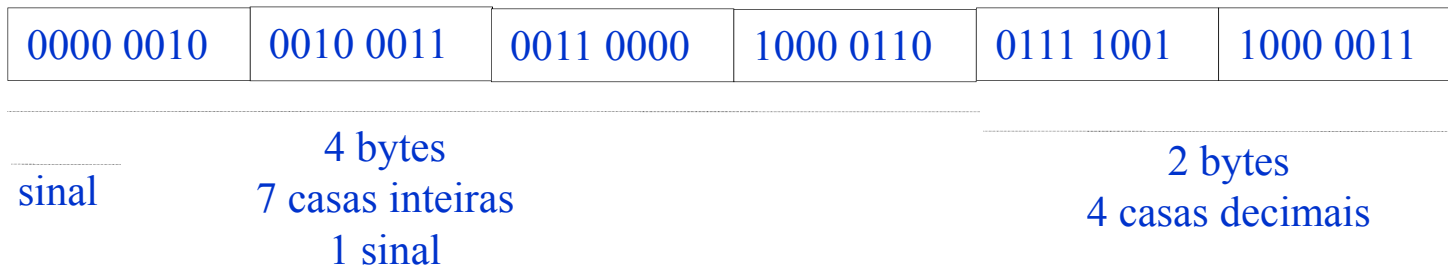


- Por que a linguagem de programação C, umas das mais usadas da história, não implementa o tipo booleano?

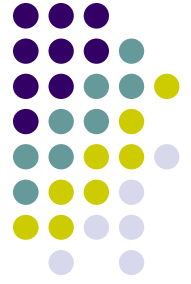


# Tipo Decimal

- Armazena um número fixo de dígitos decimais
  - Existe, por exemplo, no Cobol e no Pascal



# Pense...



- Variáveis do tipo Decimal armazenam os valores numa variável inteira, logo ela não pode ser classificada como um Tipo Primitivo, pois pode ser decomposta numa variável mais simples. Correto?

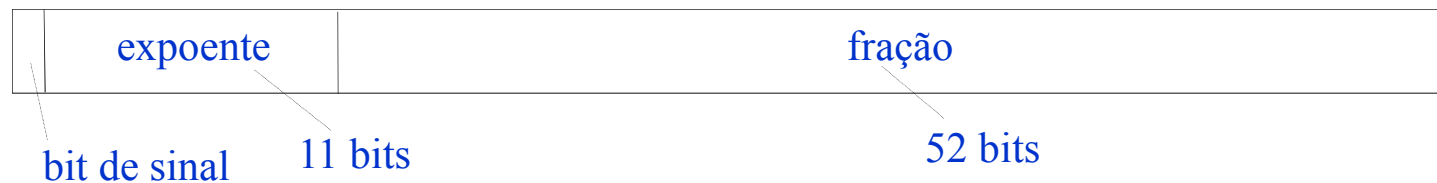


# Tipo Ponto Flutuante

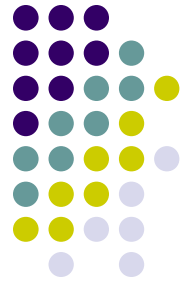
- O tipo primitivo ponto flutuante modela os números reais
- LPs normalmente incluem dois tipos de ponto flutuante: *float* e *double*



Padrão IEEE 754 - Precisão Simples



Padrão IEEE 754 - Precisão Dupla



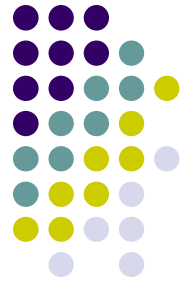
# Tipo Enumerado

- PASCAL, ADA, C e C++ permitem que o programador defina novos tipos primitivos através da enumeração de identificadores dos valores do novo tipo

```
enum mes_letivo { mar, abr, mai, jun, ago, set, out, nov };
```

```
enum mes_letivo m1, m2;
```

- Possuem correspondência direta com intervalos de tipos inteiros e podem ser usados para indexar vetores e para contadores de repetições
- Aumentam a legibilidade e confiabilidade do código
- Java não inclui o tipo enumerado de C e C++



# Tipo Intervalo de Inteiros

- Em PASCAL e ADA, também é possível definir tipos intervalo de inteiros

```
type meses = 1 .. 12;
```

- Tipos intervalos herdam as operações dos inteiros



# Tipos Compostos

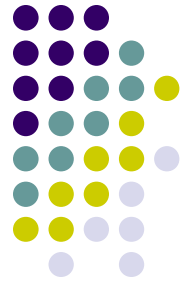
- Tipos compostos são aqueles que podem ser criados a partir de tipos mais simples  
registros, vetores, listas, arquivos

- Permitem Operações

Ex.: Produto cartesiano, união e conjunto de potências

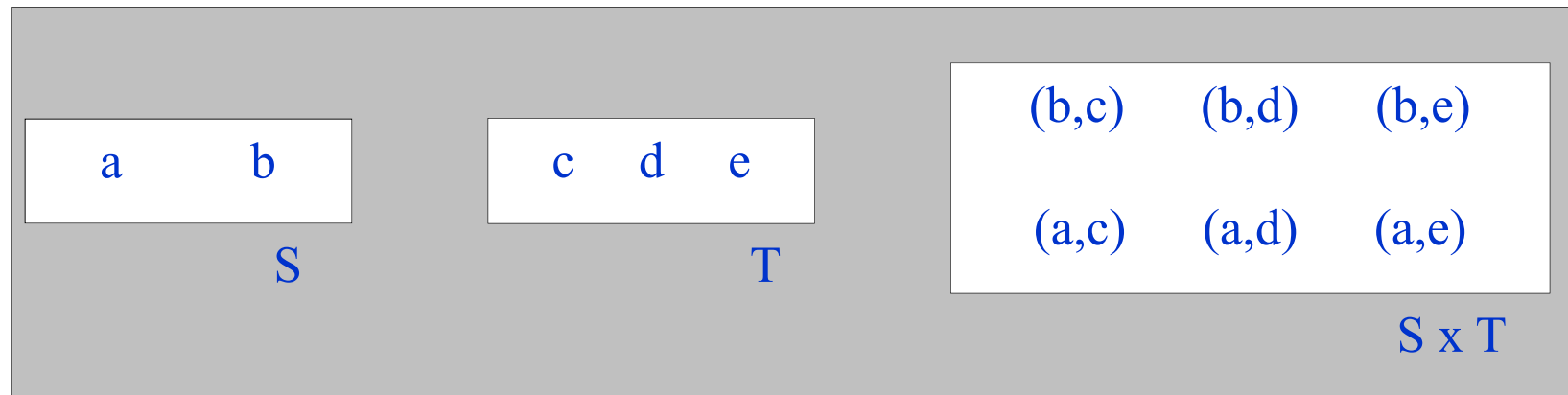
- Possuem Cardinalidade

Número de valores distintos que fazem parte do tipo



# Produto Cartesiano

- Combinação de valores de tipos diferentes em tuplas



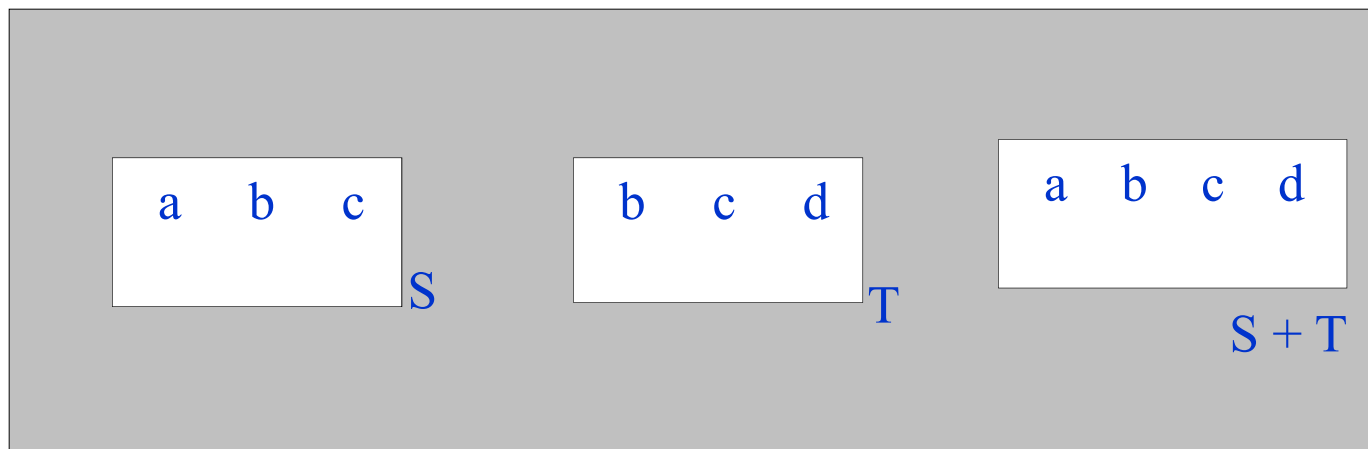
- São produtos cartesianos os registros de PASCAL e COBOL e as estruturas de C
- Em LPs orientadas a objetos, produtos cartesianos são definidos a partir do conceito de classe

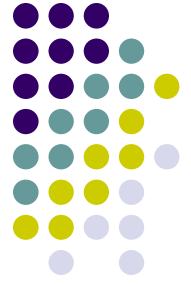


# Unões



- Consiste na união de valores de tipos distintos para formar um novo tipo de dados

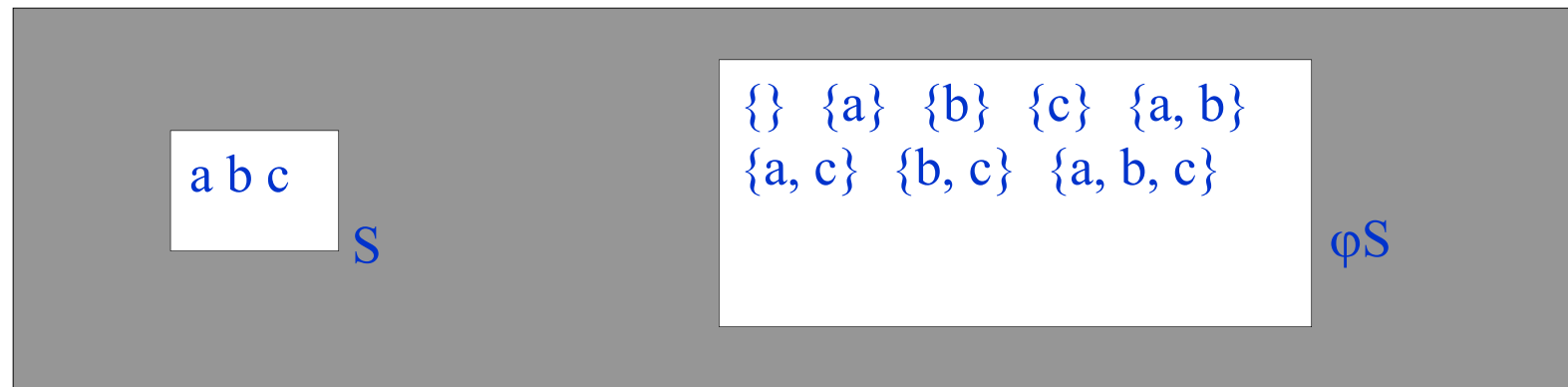


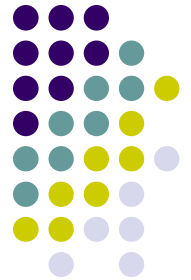


# Conjuntos Potência

- Tipos de dados cujo conjunto de valores corresponde a todos os possíveis subconjuntos que podem ser definidos a partir de um tipo base  $S$

$$\varphi S = \{s \mid s \subseteq S\}$$





# Conjuntos Potência

- Poucas LPs oferecem. Muitas vezes de forma restrita
- PASCAL

TYPE

Carros = (corsa, palio, gol, civic, vectra);

ConjuntoCarros = SET OF Carros;

VAR

Carro: Carros;

CarrosPequenos: ConjuntoCarros;

BEGIN

Carro:= corsa;

CarrosPequenos := [palio, gol]; /\*atribuicao\*/

CarrosPequenos:= CarrosPequenos + [corsa]; /\*uniao\*/

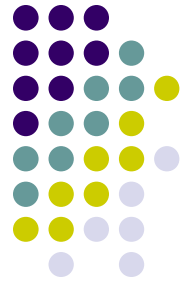
if Carro in CarrosPequenos THEN

.....



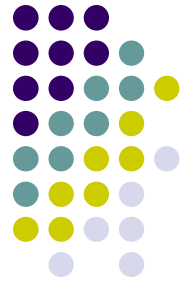
# Categorias de Vetores

<b>Categoria de Vetor</b>	<b>Tamanho</b>	<b>Tempo de Definição</b>	<b>Alocação</b>	<b>Local de Alocação</b>	<b>Exemplos</b>
Estáticos	Fixo	Compilação	Estática	Base	FORTRAN 77
Semi-Estáticos	Fixo	Compilação	Dinâmica	Pilha	PASCAL, C, MODULA 2
Semi-Dinâmicos	Fixo	Execução	Dinâmica	Pilha	ALGOL 68, ADA, C
Dinâmicos	Variável	Execução	Dinâmica	Monte	APL, PERL



# Vetores Dinâmicos

- Podem ser implementados em Pascal, C, C++ e JAVA
- É necessário alocar nova memória e copiar conteúdo quando vetor aumenta de tamanho
- É encargo do programador controlar alocação e cópia. Em C e C++, o programador deve controlar desalocação também. Isso torna a programação mais complexa e suscetível a erros



# Vetores Multidimensionais

- Também são conhecidos como matrizes.
- Em linguagens que não possuem o conceito de matrizes, como JAVA, vetores multidimensionais são obtidos com o uso de vetores unidimensionais cujos elementos são outros vetores

```
int [ ] [ ] a = new int [5] [ ];
```

- O mesmo efeito pode ser obtido em C com o uso de ponteiros para ponteiros



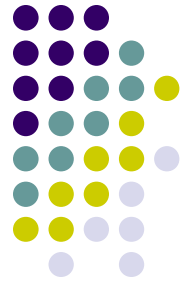
# Tipos Recursivos

- Tipos recursivos são tipos de dados cujos valores são compostos por valores do mesmo tipo

$R ::= \langle \text{parte inicial} \rangle R \langle \text{parte final} \rangle$

Tipo Lista  $::=$  Tipo Lista Vazia  $|$  (Tipo Elemento  $\times$  Tipo Lista)

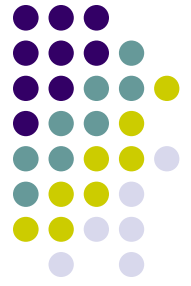
- A cardinalidade de um tipo recursivo é infinita
- Isto é verdade mesmo quando o tipo do elemento da lista é finito
- O conjunto de valores do tipo listas é infinitamente grande (não podendo ser enumerado) embora toda lista individual seja finita



# Tipos Ponteiros

- Não se restringe a implementação de tipos recursivos embora seja seu uso principal
- Ponteiro é um conceito de baixo nível relacionado com a arquitetura dos computadores
- O conjunto de valores de um tipo ponteiro são os endereços de memória e o valor nil
- Considerados o grito das estruturas de dados





# Tipo Ponteiros: Problemas

- Baixa Legibilidade
- Possibilitam violar o sistema de tipos

```
int i, j = 10;  
int* p = &j; // p aponta para a variavel inteira j  
p++;        // p pode nao apontar mais para um inteiro  
i = *p + 5;    // valor imprevisivel atribuido a i
```

- Objetos Pendentes

```
int* p = (int*) malloc (10*sizeof(int));  
int* q = (int*) malloc (5*sizeof(int));  
p = q;           // area apontada por p torna-se inacessivel
```



# Tipo String

- Valores correspondem a uma seqüência de caracteres
- Não existe consenso sobre como devem ser tratadas
- Podem ser consideradas tipos primitivos, mapeamentos finitos ou tipo recursivo lista
- Três formas comuns de implementação
  - Estática
  - Semi-Estática
  - Dinâmica

