

# Computação Distribuída

**Odorico Machado Mendizabal**



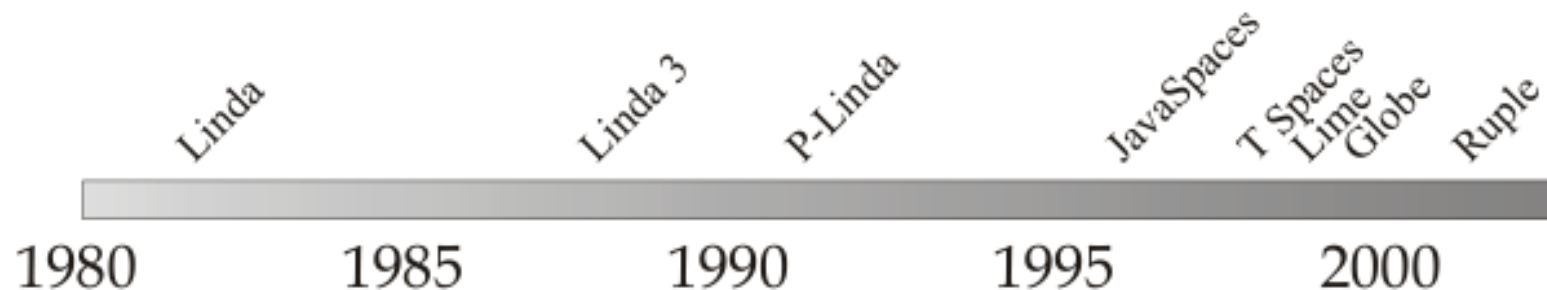
Universidade Federal de Santa Catarina – UFSC  
Departamento de Informática e Estatística – INE



# **Espaço de Tuplas**

# Contextualização

- Paradigma para programação paralela, inicialmente como um modelo teórico (linguagem Linda, meados dos anos 80)
- A partir dos anos 1990, os espaços de tuplas foram efetivamente explorados como ferramenta de programação distribuída
  - JavaSpaces
  - TSpaces (IBM)
  - C-Linda, ...



# Tuplas

$$t = (x_1, x_2, \dots, x_n)$$

$t$  é uma **tupla** composta por itens  $x_1$  até  $x_n$

Um **molde** de  $t$ ,  $\bar{t}$  é uma tupla composta por lacunas e um subconjunto dos itens de  $t$ .

$$\text{Ex. } \bar{t} = (\_, x_2, \dots, x_n)$$

# Tuplas

- Uma tupla é uma sequência ordenada de  $n$  elementos (n-upla)
- Por exemplo, uma quádrupla:  
("Tanenbaum", "Redes de Computadores", "Pearson", 2011)
- A tupla de exemplo representa informações de um livro, de acordo com as atributos:

(Autor, Título, Editora, Ano)

# Espaço de Tuplas

- Um espaço de tuplas pode ser visto como um espaço de armazenamento compartilhado, que fornece operações para armazenar e recuperar conjuntos de dados organizados na forma de tuplas
- Uma abstração de memória compartilhada sobre um sistema distribuído

# Operações básicas sobre o Espaço de Tuplas

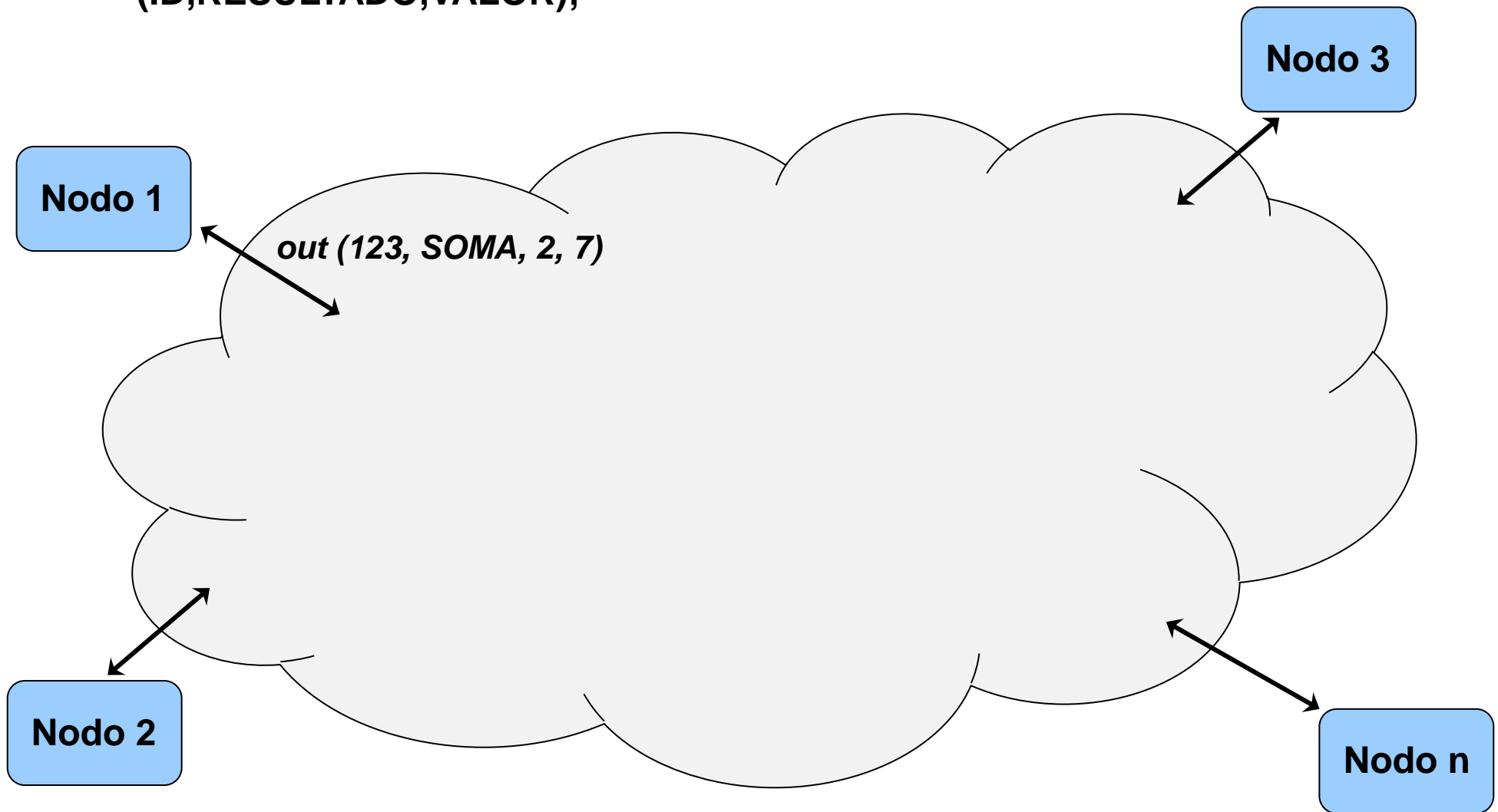
$out ( t )$ : adiciona uma tupla  $t$  no espaço de tuplas

$in ( \overline{t} )$ : lê e remove do espaço de tuplas, uma tupla  $t$  que combine com o molde  $\overline{t}$

$rd ( \overline{t} )$ : somente faz a leitura de uma tupla combinando com  $\overline{t}$ , sem removê-la do espaço

# Espaço de Tuplas

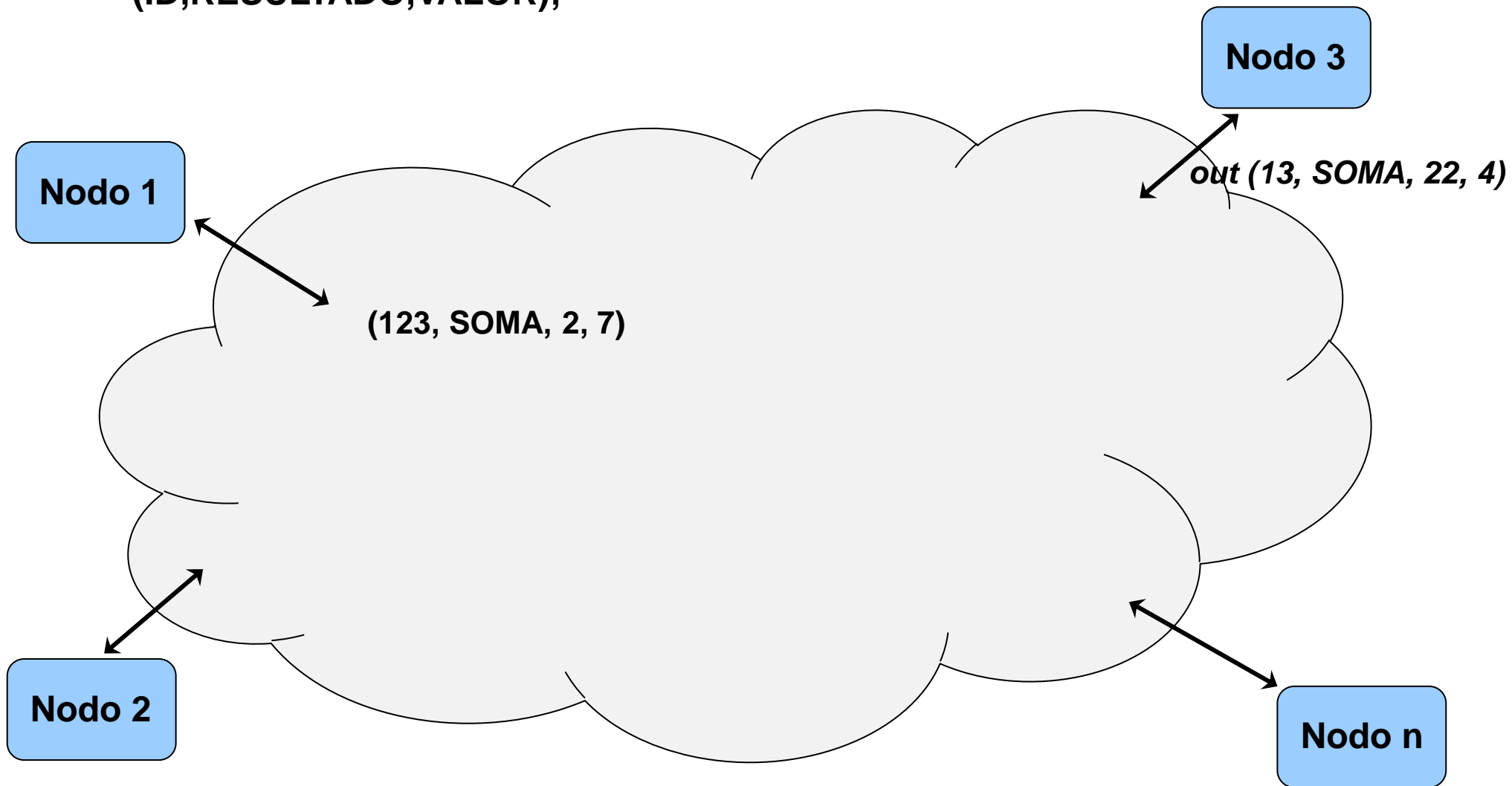
Tuplas: (ID,OPERAÇÃO,ARGUMENTO1,ARGUMENTO2);  
(ID,RESULTADO,VALOR);





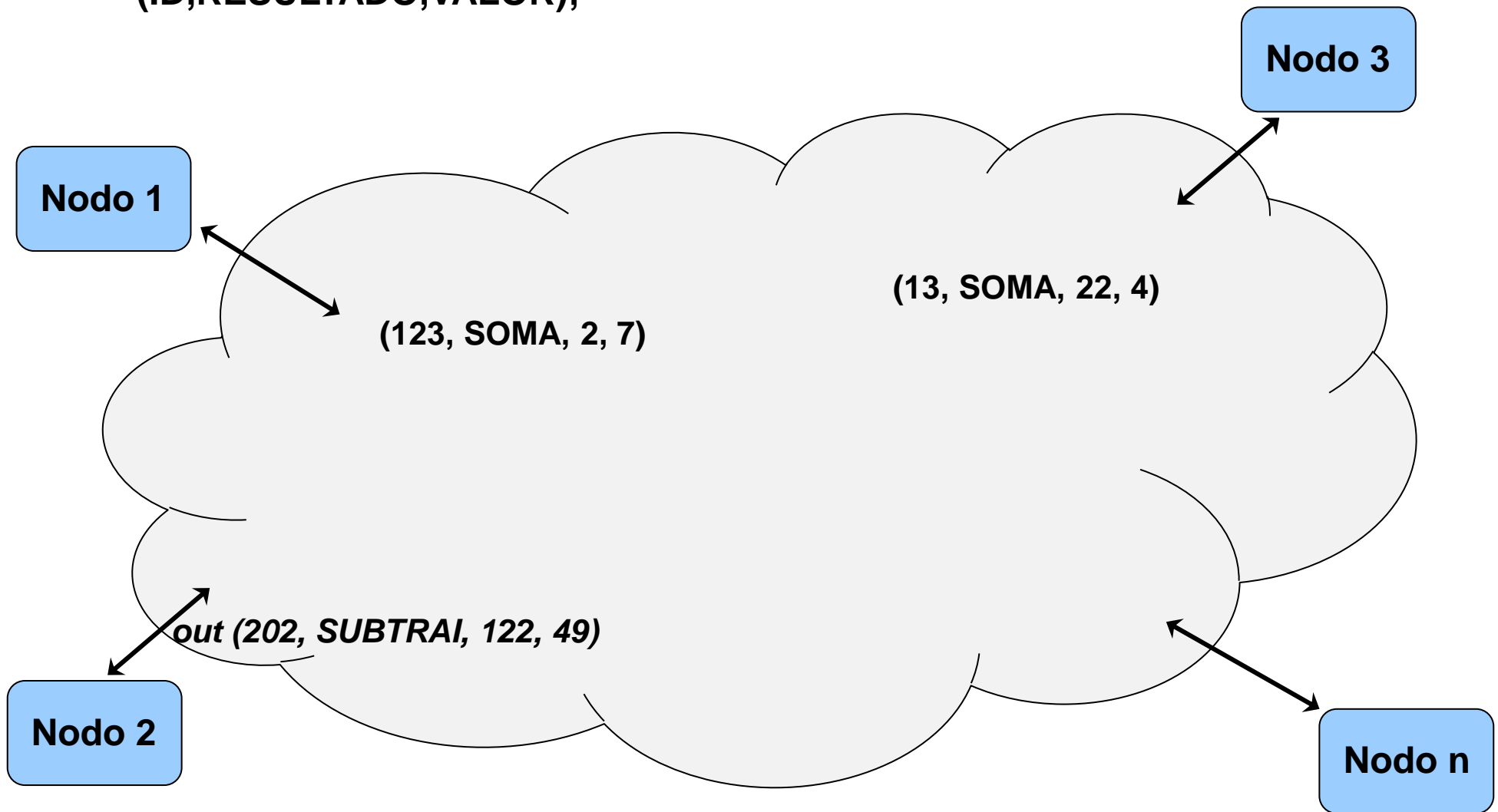
# Espaço de Tuplas

Tuplas: (ID,OPERAÇÃO,ARGUMENTO1,ARGUMENTO2);  
(ID,RESULTADO,VALOR);



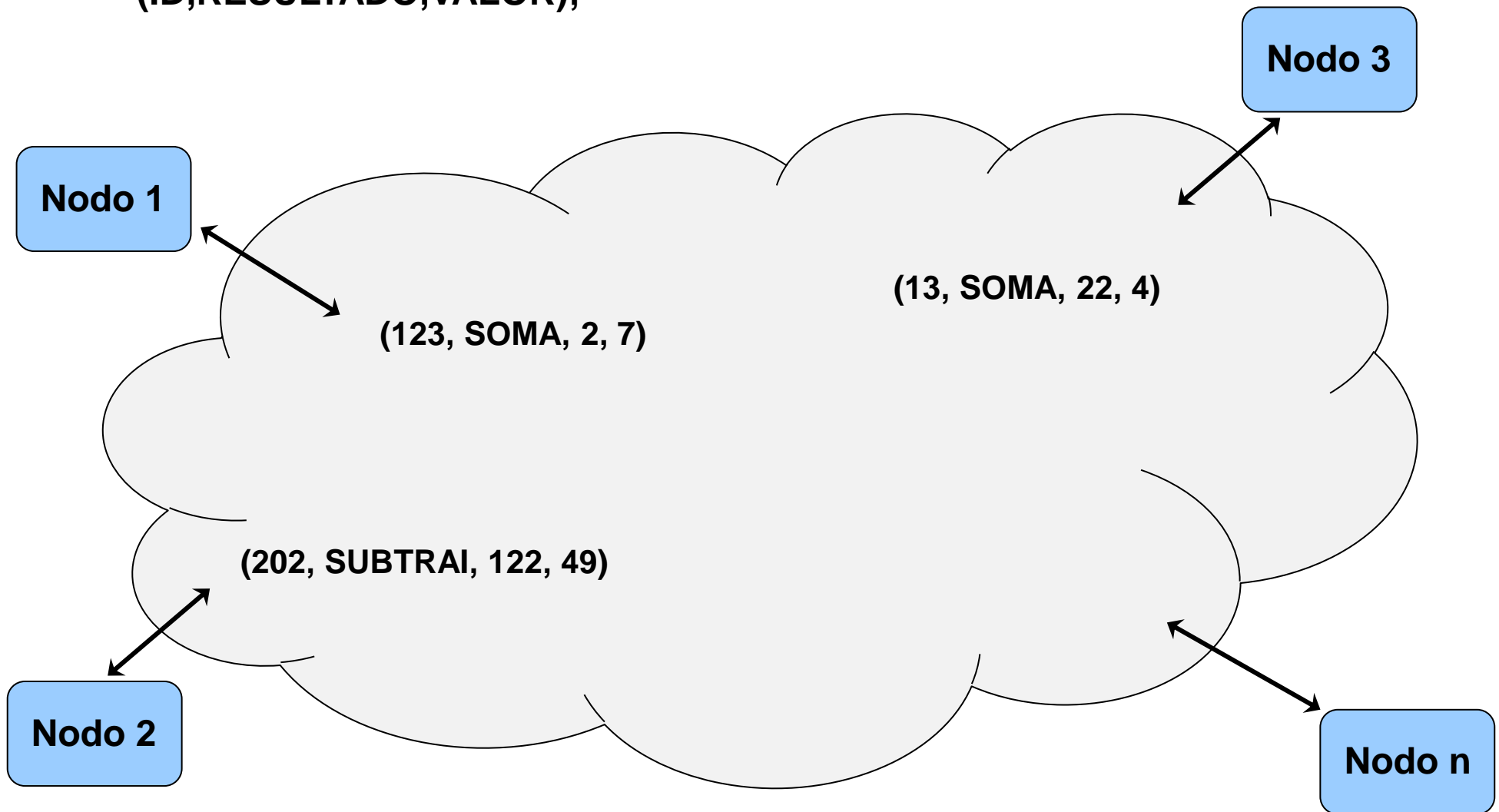
# Espaço de Tuplas

Tuplas: (ID,OPERAÇÃO,ARGUMENTO1,ARGUMENTO2);  
(ID,RESULTADO,VALOR);



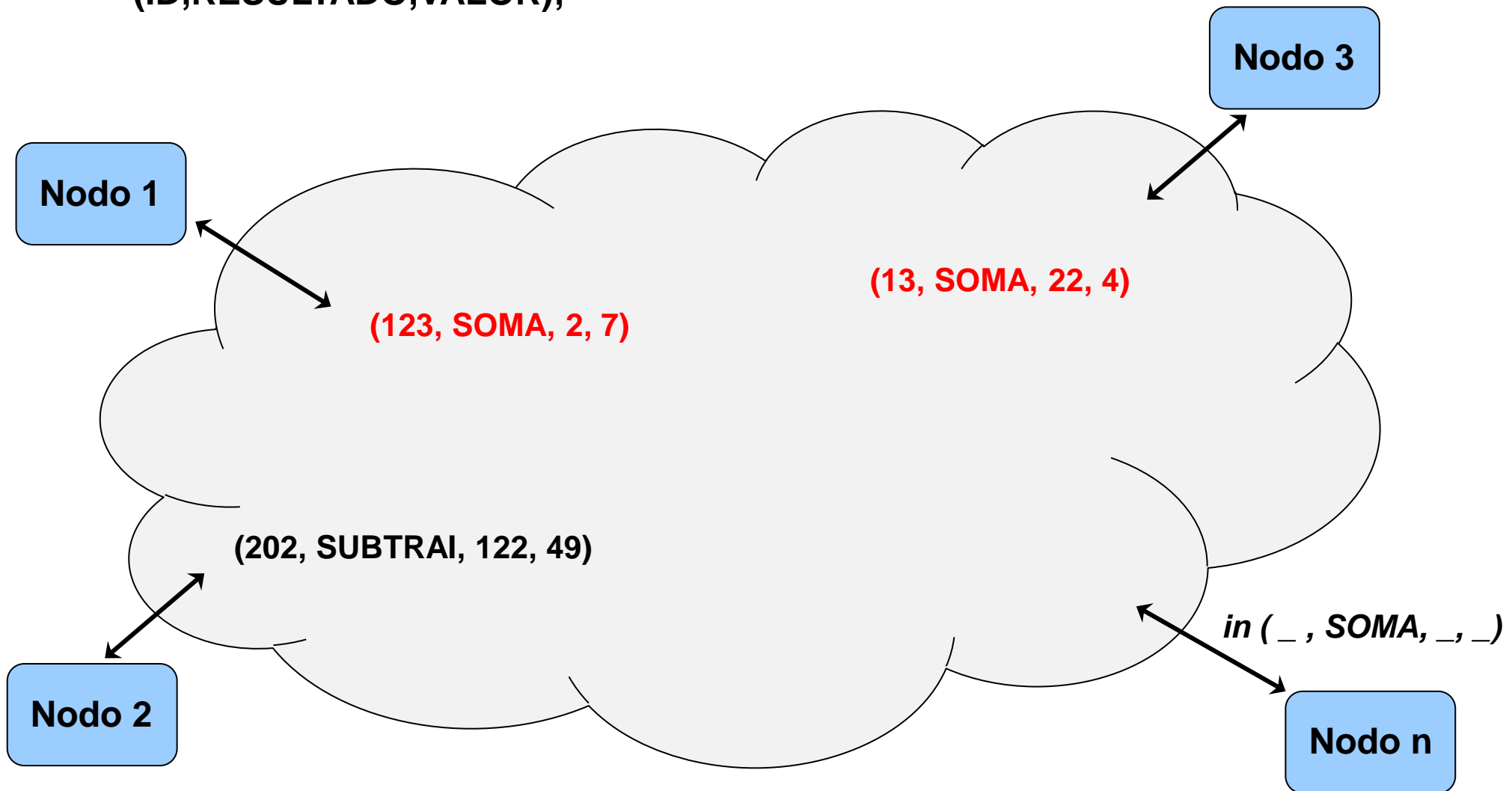
# Espaço de Tuplas

Tuplas: (ID,OPERAÇÃO,ARGUMENTO1,ARGUMENTO2);  
(ID,RESULTADO,VALOR);



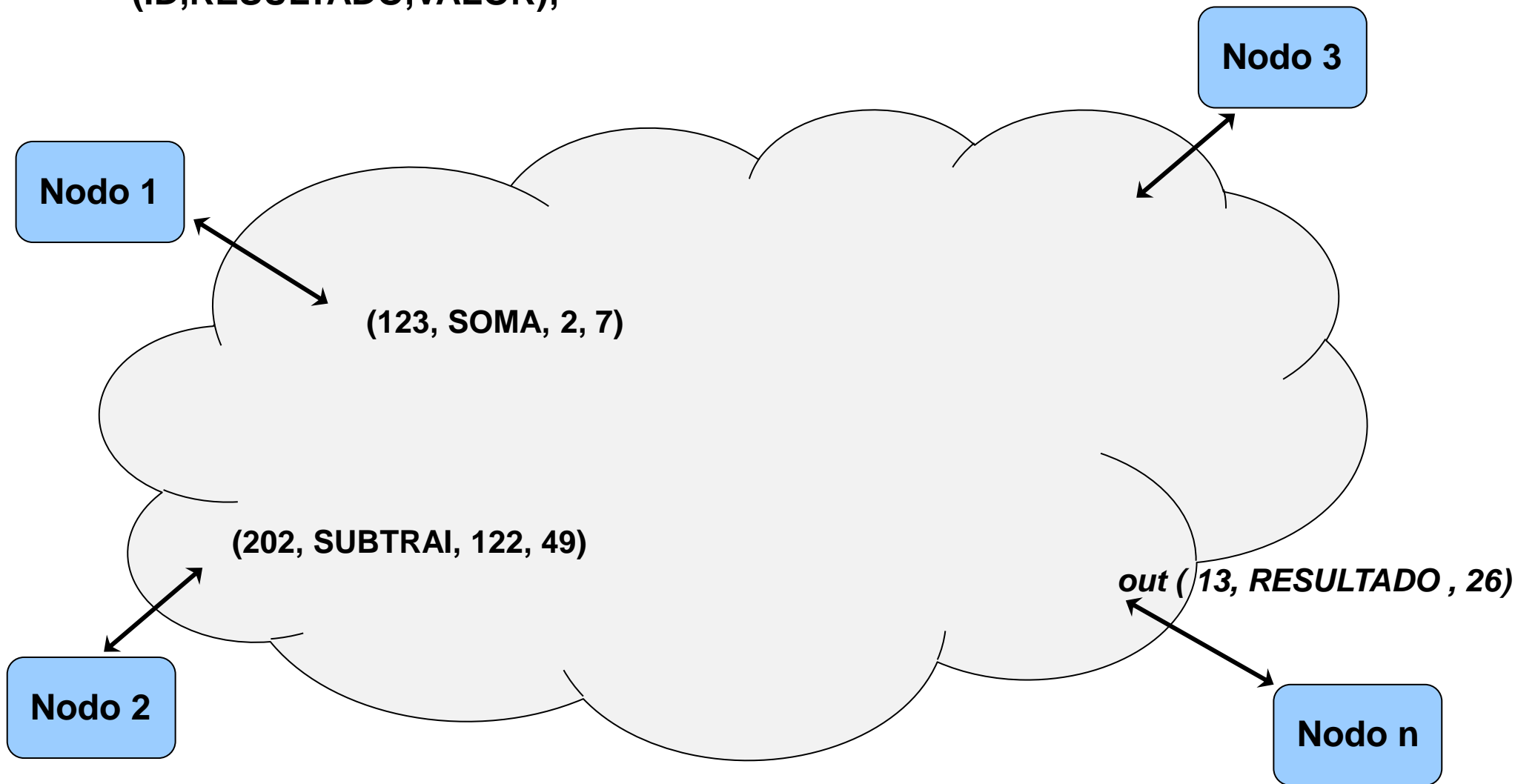
# Espaço de Tuplas

Tuplas: (ID,OPERAÇÃO,ARGUMENTO1,ARGUMENTO2);  
(ID,RESULTADO,VALOR);



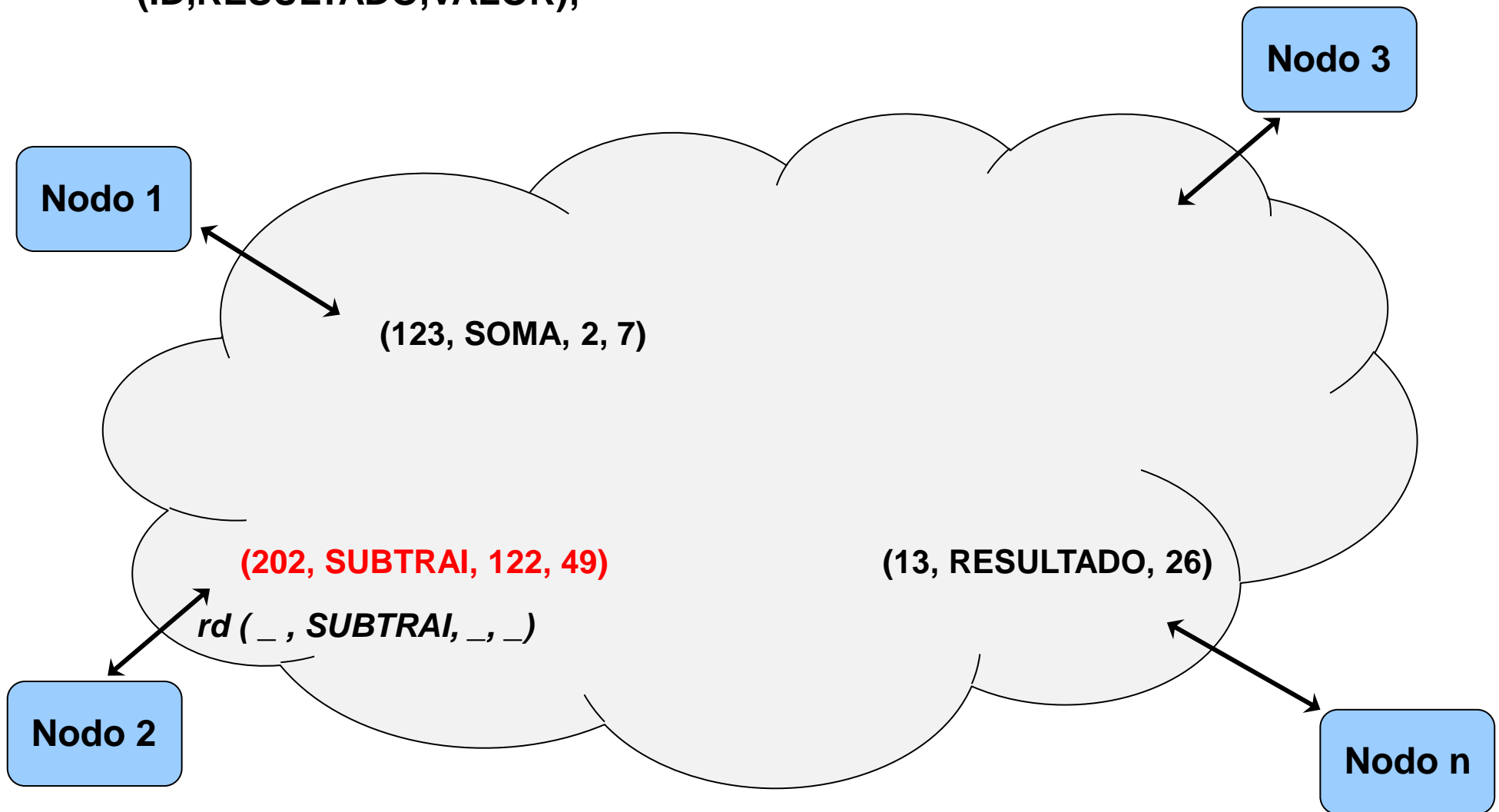
# Espaço de Tuplas

Tuplas: (ID,OPERAÇÃO,ARGUMENTO1,ARGUMENTO2);  
(ID,RESULTADO,VALOR);



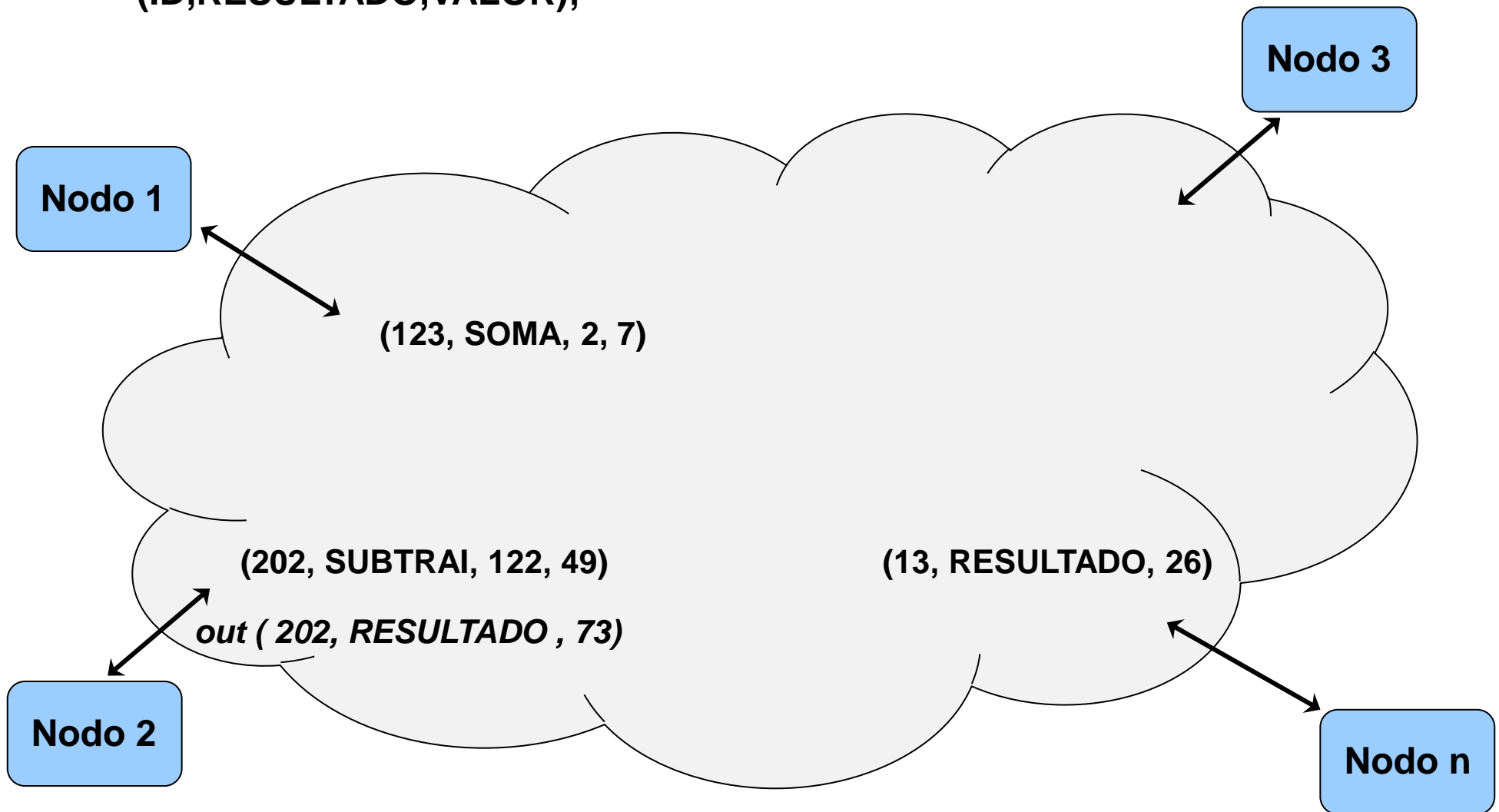
# Espaço de Tuplas

Tuplas: (ID,OPERAÇÃO,ARGUMENTO1,ARGUMENTO2);  
(ID,RESULTADO,VALOR);



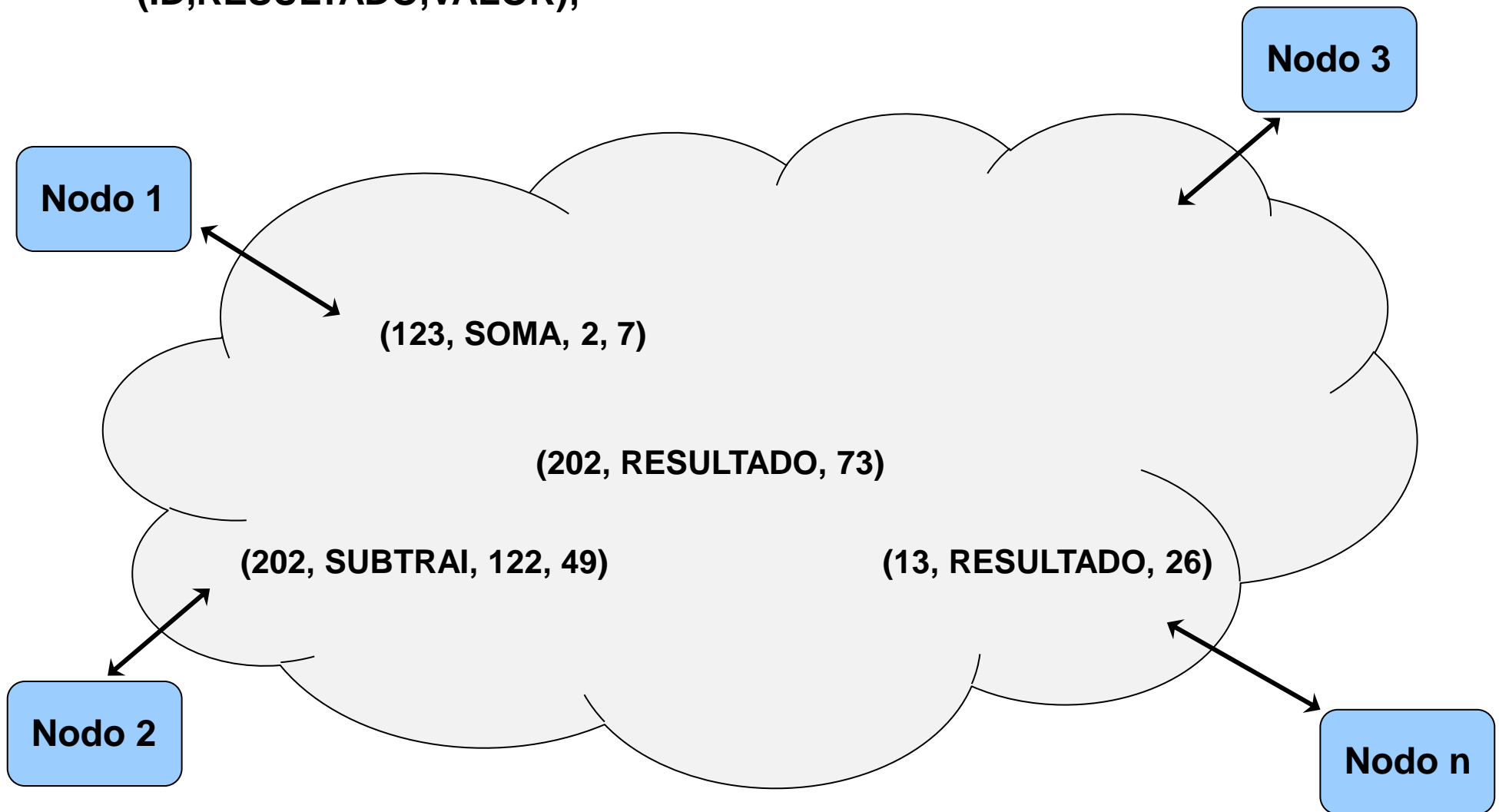
# Espaço de Tuplas

Tuplas: (ID,OPERAÇÃO,ARGUMENTO1,ARGUMENTO2);  
(ID,RESULTADO,VALOR);



# Espaço de Tuplas

Tuplas: (ID,OPERAÇÃO,ARGUMENTO1,ARGUMENTO2);  
(ID,RESULTADO,VALOR);





# Características das operações

- Tanto para a operação de escrita e leitura, o parâmetro retornado é uma tupla
- Operações *in* e *rd* são bloqueantes
- Cria-se as operações não bloqueantes: *inp* e *rdp*

# Busca por tuplas

- Uma tupla  $t$  e um molde  $\bar{t}$  combinam se, e somente se:
  - Ambos têm o mesmo número de campos,
  - Todos os valores dos campos definidos em  $t$  possuem o mesmo valor dos campos correspondentes em  $\bar{t}$

Exemplo:

- Uma tupla (“Copa do Brasil”, “Grêmio”, 2016) combina com o molde (“Copa do Brasil”, \*, 2016)

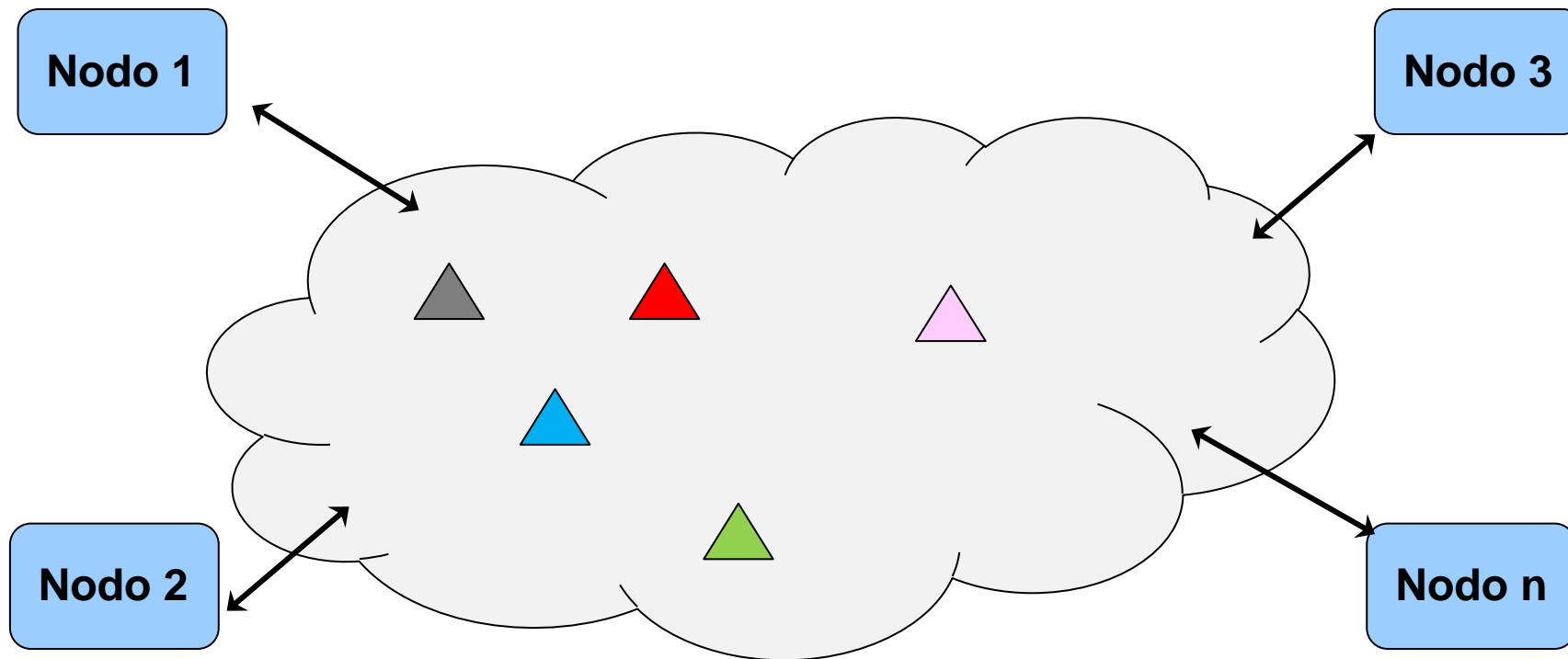
# Busca por tuplas

- Como resultado da operação, o espaço retorna uma tupla que possui os mesmos valores nos campos especificados pelo molde
  - Caso mais de uma tupla atenda a este requisito, apenas uma será retornada como resultado da operação
- Se nenhuma tupla atende ao requisito:
  - Processo fica bloqueado até que uma tupla escrita satisfaça o requisito (operações bloqueantes)
  - Operações retorna uma tupla vazia para o processo chamador

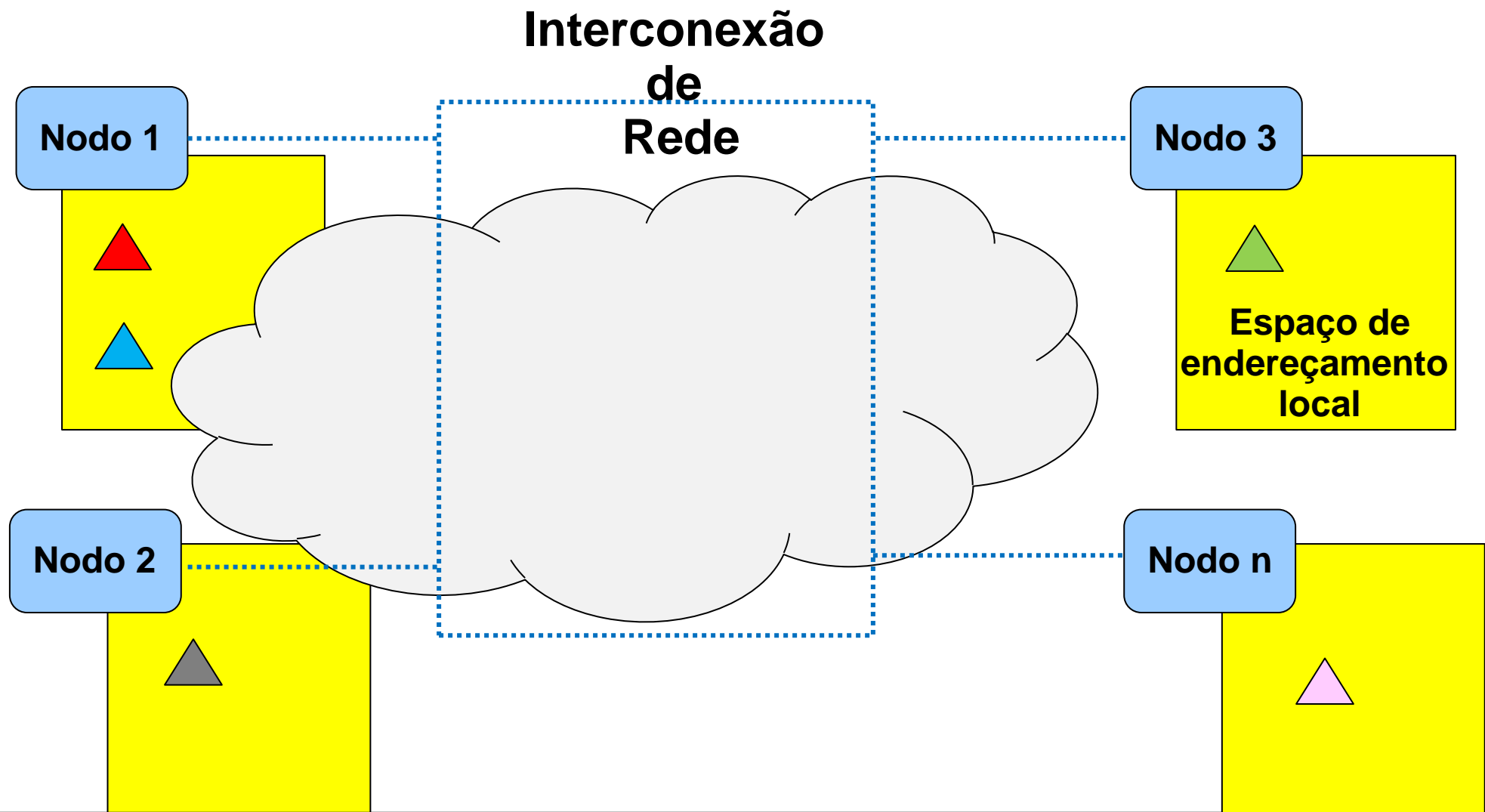
# Visão abstrata de um Espaço de Tuplas

**Processos acessam um espaço de memória compartilhado**

**- Aspectos relacionados à localização, sincronização (,tolerância à falhas) são abstraídos pelo programador**



# Disposição real dos recursos de um Espaço de Tuplas



# Exemplo JavaSpace

# Java Spaces

## Principais operações:

**write:** escreve uma entrada no serviço JavaSpace

**read:** lê uma entrada do JavaSpace que esteja de acordo com o dado *template*

**take:** lê uma entrada, assim como operação read, porém retira a entrada do JavaSpace

**notify:** notifica um objeto especificado quando entradas que correspondem a um dado template são escritas no JavaSpace

