



# Programação em Java

## Sessão 03



# **Tecnologia JAVA**

---

Algoritmos  
Celina Lopes



## Sumário

- algoritmos
- Pseudo-código



## Algoritmos

### Componentes de uma linguagem

**Léxico:** conjunto de palavras duma linguagem, i.e. o *dicionário*.

**Sintaxe:** conjunto de regras de construção das frases dum texto, i.e. a *gramática*. Estas frases são designadas por instruções e o texto é designado por programa.

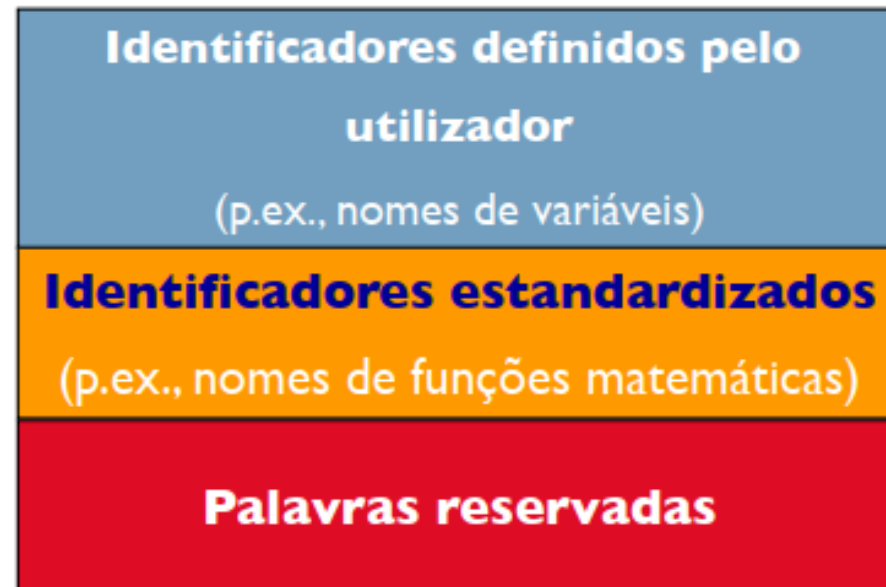
**Semântica:** é o significado de cada instrução de um programa ou mesmo de um do programa completo.



## Algoritmos

### Classificação de nomes/identificadores

- Há 3 classes de nomes ou identificadores:





## Algoritmos

### Entidades e expressões

#### ***Entidades:***

identificadores ou nomes (de variáveis, constantes e funções)  
operadores (adição, subtração, atribuição, ...)

#### ***Expressões:***

É uma construção sintática que tem um valor.

É uma combinação de entidades que representa o cálculo de um valor.

#### ***Tipos de expressões:***

expressão numérica (resultado numérico)

expressão lógica (resultado lógico)

expressão simbólica (resultado simbólico)



## Algoritmos

### Operadores

Escrita de dados em memória (**operador de ATRIBUIÇÃO**)

=

Cálculo de inteiros e/ou reais (**operadores ARITMÉTICOS**)

+, -, \*, /

Cálculo de valores lógicos (falso/verdadeiro) (**operadores LÓGICOS**)

&&, ||, !

Cálculo de valores lógicos (falso/verdadeiro) (**operadores RELACIONAIS**)

<, <=, >, >=, !=, ==

Incrementa valor de variável inteira de 1 unidade (**operador de INCREMENTAÇÃO**)

++

Decrementa valor de variável inteira de 1 unidade (**operador de DECREMENTAÇÃO**)

--



## Algoritmos

### Literais

**Definição:** Uma entidade sem nome que representa um valor constante.

#### Exemplos:

- “XYZ123” é um literal string
- ‘g’ é um literal carácter
- 1234 é um literal inteiro
- 3.78 é um literal real
- 6/10/57 é um literal data





## Algoritmos

### Sumário até agora:

**PROGRAMA** = sequência de INSTRUÇÕES

**INSTRUÇÃO** = EXPRESSÃO seguida de ponto e vírgula

**EXPRESSÃO** = sequência de TERMOS separados por operadores



## Algoritmos

### Tipos de instruções

Uma instrução é uma das seguintes entidades:

uma *instrução-bloco*

uma *instrução-de-atribuição*

uma *instrução-if*

uma *instrução-while*

uma *instrução-for*

uma *instrução-switch*

uma *instrução-break*

uma *instrução-continue*

uma *instrução-return*

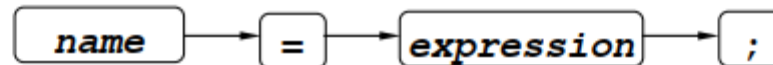


## Algoritmos

### Instrução de atribuição

**Definição:** É uma instrução que atribui um valor a uma variável em memória.

**Diagrama de sintaxe:**





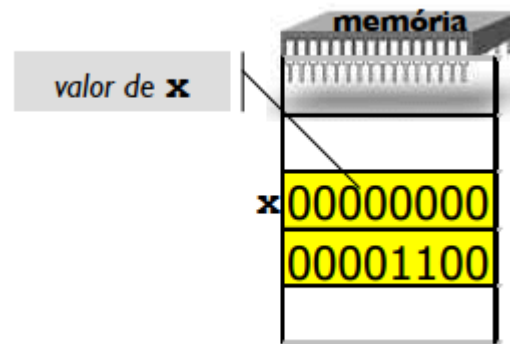
## Algoritmos

### Variável

**Definição:** É um local de memória onde se guarda um dado específico.

Têm nome e valor.

À semelhança das variáveis em matemática, uma variável não tem valor constante. Pode mudar de valor através duma instrução de atribuição.

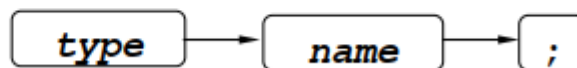




## Algoritmos

### Declaração de variáveis

**Diagrama de sintaxe:**



```
int x;  
x=12;
```



## Algoritmos

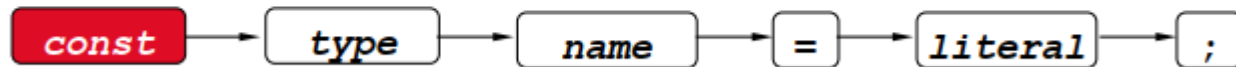
### Constante

**Definição:** É também um local de memória onde se guarda um dado específico.

Também tem nome e valor.

- À semelhança das constantes em matemática, uma constante tem valor fixo e imutável

#### Diagrama de sintaxe:





## Algoritmos

### Problemas e Algoritmos

Para resolver um **problema** através dum computador é necessário encontrar em primeiro lugar uma maneira de descrevê-lo de uma forma clara e precisa.

- É também preciso que encontremos uma sequência de passos que conduzam à sua resolução.

Esta sequência de passos é designada por **algoritmo**.

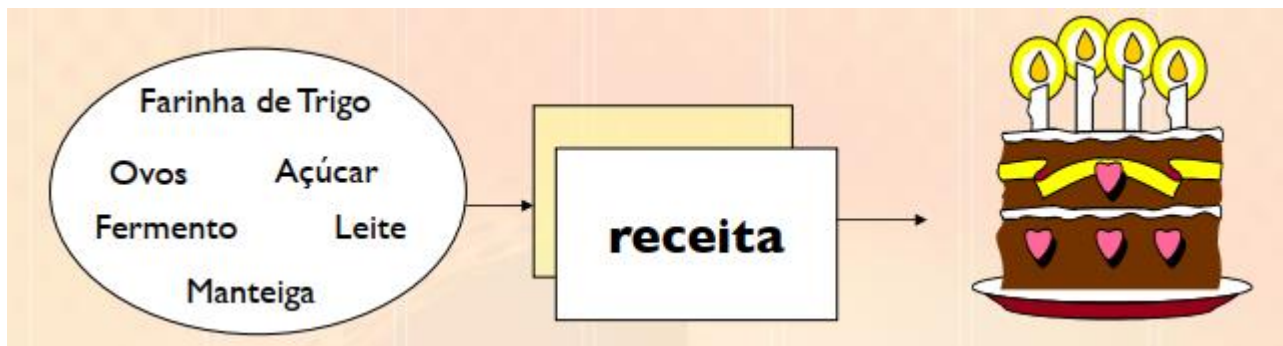
- A noção de algoritmo é central a toda a informática.
- A construção de algoritmos para resolver os problemas é uma das maiores dificuldades, mas também um dos maiores desafios para quem programa computadores.



## Algoritmos

### Noção de Problema

#### Exemplo: como fazer um bolo?



Uma receita é uma descrição dum conjunto de **passos** ou **ações** que fazem a combinação de um conjunto de ingredientes com vista a obter um produto gastronómico particular





## Algoritmos

### Noção de Algoritmo



Um algoritmo opera sobre um conjunto de entradas (farinha ovos, fermento, etc. no caso do bolo) de modo a gerar uma saída que seja útil (ou agradável) para o utilizador (o bolo pronto).

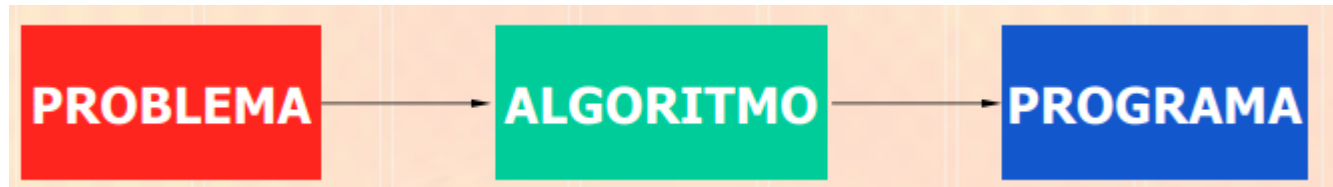
Assim, os passos ou ações de um algoritmo para confeccionar um bolo são os seguintes:

1. Bater duas claras em castelo;
2. Adicionar duas gemas;
3. Adicionar um xícara de açúcar;
4. Adicionar duas colheres de manteiga;
5. Adicionar uma xícara de leite de coco;
6. Adicionar farinha e fermento;
7. Colocar numa forma e levar ao forno em lume brando.



## Algoritmos

### Desenho ou Conceção de Algoritmos para a Resolução de Problemas



- Um algoritmo é uma descrição, passo-a-passo, de uma metodologia que conduz à resolução de um problema ou à execução de uma tarefa.
- A programação consiste na codificação precisa desse algoritmo, segundo uma linguagem de programação específica.
- Há, pois, que ter em consideração que existem três fases distintas na elaboração de programas:
  - a análise do problema (especificação do problema, análise de requisitos, pressupostos, etc.)
  - a concepção do algoritmo
  - a tradução desse algoritmo na linguagem de programação



## Algoritmos

### Passos na Conceção e Construção de Algoritmos

- Compreender o problema
- Identificar os dados de entrada
- Identificar os dados de saída
- Determinar o que é preciso para transformar dados de entrada em dados de saída:
  - usar a estratégia do **dividir-para-conquistar**
  - observar regras e limitações
  - identificar todas as ações a realizar
  - eliminar ambiguidades
- Construir o algoritmo
- Testar (manualmente) o algoritmo
- Executar o algoritmo



## Algoritmos

### Dividir-para-Conquistar

- Este método consiste em dividir um problema em partes menores (ou sub-problemas) de modo a que seja mais fácil a sua resolução.
  - **Exemplo:** Fazer sumo de laranja?
    - Lavar laranja;
    - Partir laranja ao meio;
    - Espremer laranja;
    - Filtrar o sumo;
    - Servir o sumo.
  - Passo-a-passo, significa que cada passo é completado antes que o próximo comece.
  - **Exemplo:** é impossível “ver telejornal” antes de executar por inteiro o passo anterior de “ligar a TV”

Pode haver mais do que um algoritmo para resolver um problema.  
Por exemplo, para ir de casa até o trabalho, podemos escolher diversos meios de transportes em função do preço, conforto, rapidez, etc..



## Algoritmos

### Representações de Algoritmos

- **Linguagem Natural**

- Os algoritmos são expressos diretamente em linguagem natural (e.g. o português como no exemplo do bolo).

- **Fluxograma (ou Diagrama de Fluxo)**

- Esta é uma representação gráfica que emprega formas geométricas padronizadas para indicar as diversas ações e decisões que devem ser executadas para resolver o problema.

- **Pseudo-linguagem**

- Emprega uma linguagem intermediária entre a linguagem natural e uma linguagem de programação para descrever os algoritmos.

Não existe consenso entre os especialistas sobre qual é a melhor maneira de representar um algoritmo. Actualmente a maneira mais comum de representar algoritmos é através de uma pseudo-linguagem ou pseudo-código. Esta forma de representação tem a vantagem de o algoritmo seja escrito de uma forma que está próxima de uma linguagem de programação de computadores.



## Algoritmos

### Codificação em Linguagem Natural (1)

- **Problema:**
  - Cálculo da média aritmética de dois valores reais?
- **Entradas:**
  - dois valores reais, X e Y
- **Saídas:**
  - a média  $M = (X + Y) / 2$
- **Algoritmo:**

1. Início
2. Ler X,Y
3. Calcular a média M de X e Y
4. Escrever M
5. Fim



## Algoritmos

### Codificação em Fluxograma (2)

- **Problema:**

- Cálculo da média aritmética de dois valores reais?

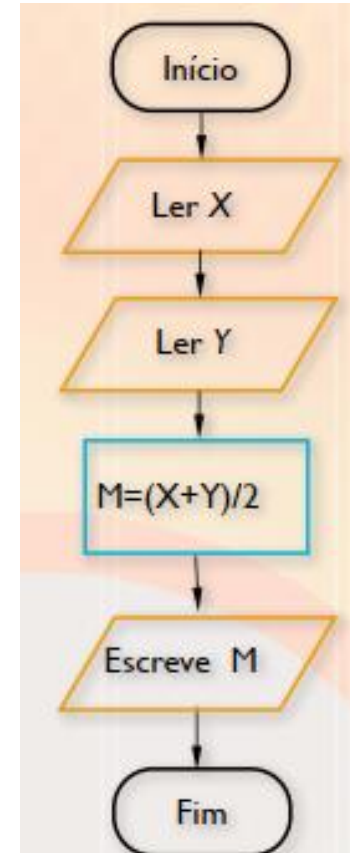
- **Entradas:**

- dois valores reais, X e Y

- **Saídas:**

- a média  $M = (X + Y) / 2$

- **Algoritmo:**





## Algoritmos

### Codificação em Pseudo-código (3)

- **Problema:**
  - Cálculo da média aritmética de dois valores reais?
- **Entradas:**
  - dois valores reais, X e Y
- **Saídas:**
  - a média  $M=(X+Y)/2$
- **Algoritmo:**

1. Início
2. Ler X,Y
3. Calcular a média  $M=(X+Y)/2$
4. Escrever M
5. Fim

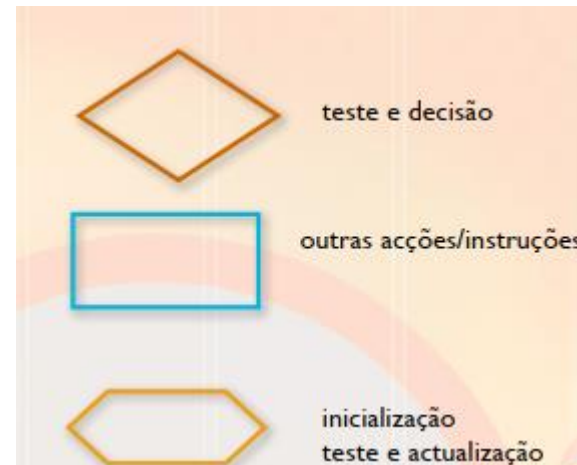




## Algoritmos

### Programação Visual com Fluxogramas

- Um fluxograma é uma representação gráfica de um algoritmo.
- Programação visual: é a utilização de diagramas na programação.
- Descrevem o fluxo de um algoritmo através de um conjunto de figuras geométricas padronizadas ligadas por setas de fluxo.

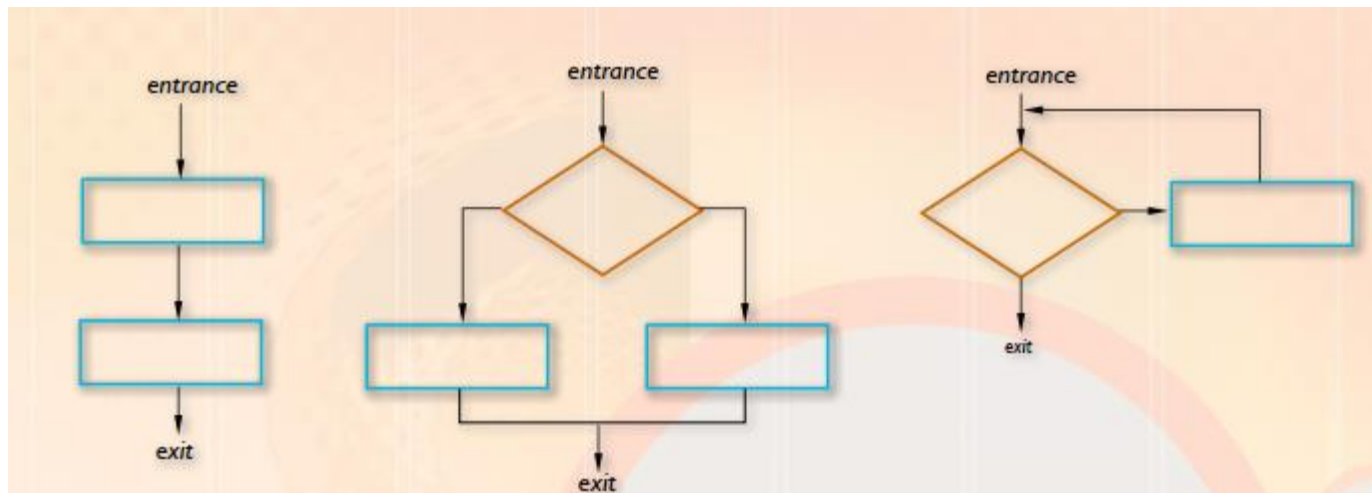




## Algoritmos

### Estruturas Lógicas de Programação

- Uma **estrutura** (de controlo) é a unidade básica da lógica de programação.
- Em meados da década de 60, alguns matemáticos provaram que qualquer programa podia ser construído através da combinação de 3 estruturas básicas: **sequência**, **seleção** e **repetição**.



SEQUÊNCIA

SELEÇÃO

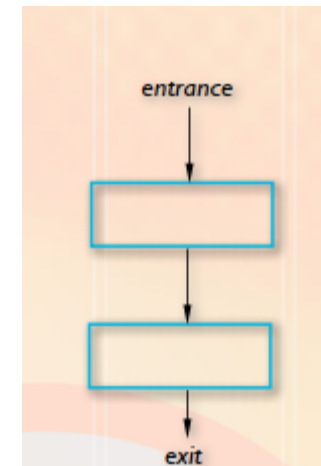
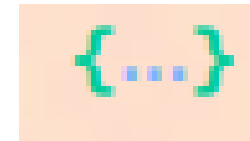
REPETIÇÃO



## Algoritmos

### Sequência

- Numa sequência é processado um conjunto de instruções (ou ações) em série.
- Não há qualquer possibilidade de alterar a ordem de processamento das instruções, i.e. após processar a 1ª instrução processa-se a 2ª, depois da 2ª processa-se a 3ª, e assim por diante até processar a última Ação.
- Em java, uma sequência é um bloco de instruções que começa com { e termina com }





## Algoritmos

### Seleção de 2-vias

- Uma estrutura de **seleção** é também designada por estrutura de **decisão**.
- Neste caso, o fluxo de processamento segue por 1 das 2 vias, dependendo do valor lógico (verdadeiro ou falso) da expressão avaliada no início da estrutura.
- Se o fluxo de processamento só passa por 1 via, então só uma das ações é realizada ou processada.
- Em java, uma estrutura de seleção com 2 vias é a instrução **if-else**.

if-else

