



**FATEC SENAI**  
Faculdade de Tecnologia SENAI Mato Grosso

**VOCÊ**  
**#SAINAFRENTE**

[WWW.FATECSENAI.COM.BR](http://WWW.FATECSENAI.COM.BR)

## Conteúdo

- Conceitos Iniciais de Algoritmos
  - Definição de algoritmos
  - Formas de representação de algoritmos
  - Tipos de dados, variáveis e constantes
  - Declaração de variáveis
  - Expressões literais, lógicas e aritméticas

## Algoritmo – Definição

“Descrição de um conjunto padronizado de ações primitivas bem definidas e executáveis, que sequênciam a realização de uma tarefa.”

“Seqüência finita de ações que descrevem como um problema pode ser resolvido.”

“Sequência finita de instruções bem definidas e não ambíguas, cada uma das quais pode ser executada mecanicamente num período de tempo finito e com uma quantidade de esforço finita.”

## Algoritmo

- Um algoritmo não representa, necessariamente, um programa de computador, e sim os passos necessários para realizar uma tarefa.
- Sua implementação pode ser feita por um computador, por outro tipo de autômato ou mesmo por um ser humano.

## Algoritmo

- São exemplos de algoritmos instruções de montagem, receitas, manuais de uso, etc.
- Um algoritmo não é a solução do problema, pois, se assim fosse, cada problema teria um único algoritmo; um algoritmo é um caminho para a solução de um problema.
- Em geral, existem muitos (senão infinitos) caminhos que levam a uma solução satisfatória.

## Algoritmo: Fritar um ovo

1. Retirar um ovo da geladeira
2. Colocar a frigideira no fogo
3. Colocar óleo
4. Esperar até o óleo ficar quente
5. Quebrar o ovo separando a casca
6. Colocar o conteúdo do ovo na frigideira
7. Esperar um minuto
8. Retirar o ovo da frigideira
9. Apagar o fogo



## Algoritmo

Existem qualidades que devem ser buscadas em qualquer algoritmo. São elas:

- 1) Perfeitamente definido
- 2) Não ambíguo
- 3) Eficaz (Consegue resolver o problema em qualquer situação. Toda situação que possa alterar o comportamento do algoritmo devem ser especificadas e tratadas.
- 4) Eficiente (Resolve o problema com o mínimo de recursos, (memória principal e auxiliar), tempo de processamento, etc.

## Formas de representação de algoritmos

As formas mais comuns de representação de algoritmos são:

- **Linguagem Natural:** Os algoritmos são expressos diretamente em linguagem natural, como no exemplo “Fritar um ovo”, instruções em manuais e uma receita culinária.
- **Fluxograma Convencional:** 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.



## Algoritmo

- Para que o algoritmo possa ser útil, é necessário ainda que quem faz uso dele conheça os termos utilizados nas instruções
- O algoritmo do exemplo “fritar um ovo” só será útil para alguém que seja fluente na língua portuguesa e conheça o significado dos verbos Retirar, Colocar, Esperar assim como dos substantivos utilizados no contexto de uma receita culinária.
- Em outras palavras, é preciso que a linguagem utilizada no algoritmo seja conhecida tanto por quem o escreveu quanto por quem vai executá-lo.

# Linguagem

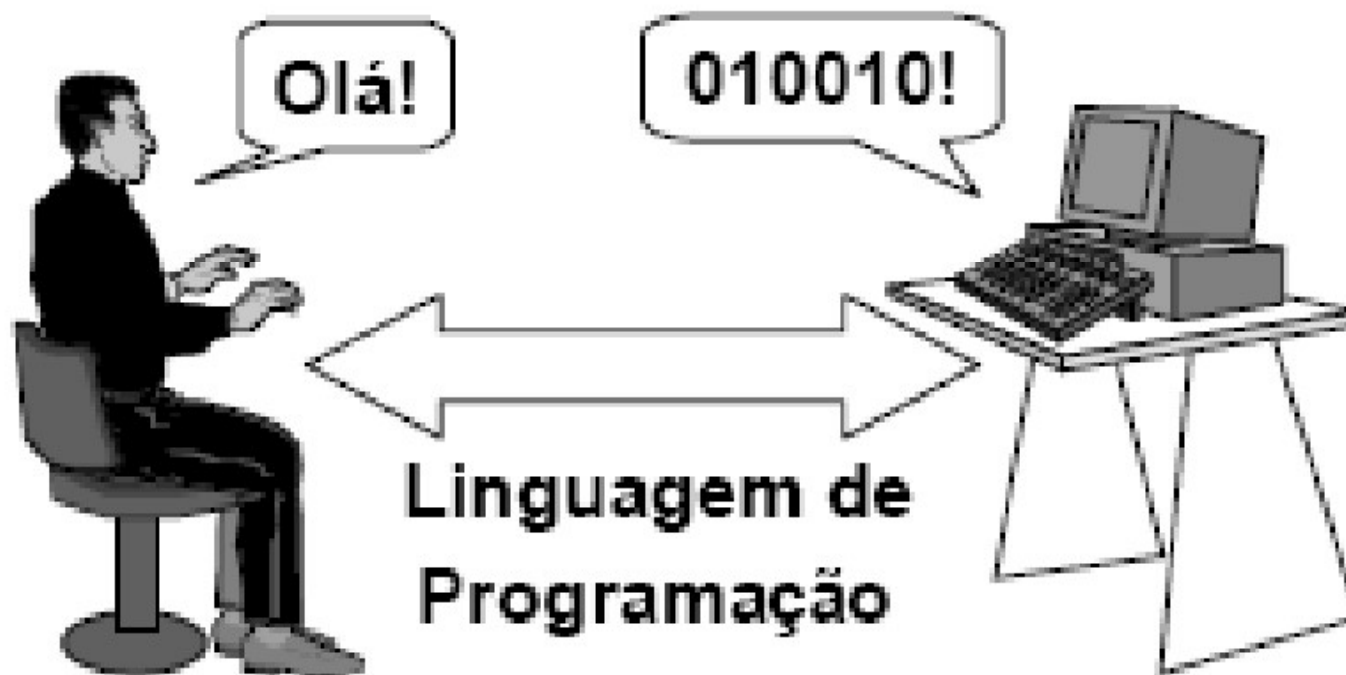
## O que é Linguagem?

“Conjunto de regras que estabelecem normas de comunicação”

- Para haver um entendimento, ambas as partes devem falar a mesma língua.

Caso as partes envolvidas na comunicação falem línguas diferentes, surge a necessidade de um tradutor(intermediário).

# Linguagem de Programação



## Linguagem de Programação

- Uma linguagem de programação deve ser extremamente formal e exata.
- Um determinado comando tem que ter o mesmo significado para todos os programadores que utilizam.

## Linguagem de Programação

- A forma especial de linguagem que utilizaremos é bem mais restrita que o Português e com significados bem definidos para todos os termos utilizados nas instruções.
- Essa linguagem é conhecida como Português Estruturado (às vezes também chamada de Portugol).

## Linguagem de Programação

- O português estruturado é, na verdade, uma simplificação extrema do Português, limitada a umas poucas palavras e estruturas que têm um significado muito bem definido.
- Ao conjunto de palavras e regras que definem o formato das sentenças válidas chamamos sintaxe da linguagem.



## Linguagem de Programação

- Embora o Português Estruturado seja uma linguagem bastante simplificada, ela possui todos os elementos básicos e uma estrutura semelhante à de uma linguagem típica para programação de computadores.
- Além disso, resolver problemas com português estruturado pode ser uma tarefa tão complexa quanto a de escrever um programa em uma linguagem de programação qualquer.

## Algoritmos Computacionais

- O computador, a princípio, não executa nada.
- Para que ele faça uma determinada tarefa – calcular uma folha de pagamento, por exemplo –, é necessário que ele execute um programa.
- Um programa é um conjunto de milhares de instruções que indicam ao computador, passo a passo, o que ele tem que fazer.

## Algoritmos Computacionais

- Um programa nada mais é do que um algoritmo computacional descrito em uma linguagem de programação
- Uma linguagem de programação contém os comandos que fazem o computador escrever algo na tela, realizar cálculos aritméticos, receber uma entrada de dados via teclado, e milhares de outras coisas, mas estes comandos precisam estar em uma ordem lógica.

## Algoritmos Computacionais

- O termo processamento de dados é muitas vezes utilizado em conjunto com computadores, pois, em geral, é isto o que eles fazem: processar dados.
- Daí podem extrair os dois componentes básicos de um algoritmo computacional: dados e código.
  - Dados são os valores (números, nomes, etc.) de que precisamos para resolver o problema.
  - Código é o conjunto de comandos ou instruções que usamos para manipular e "processar" os dados.

## Dados

- Os dados são representados pelas informações a serem tratadas (processadas) por um computador.
- Essas informações estão caracterizadas por três tipos de dados:
  - dados numéricos (inteiros e reais),
  - dados caracteres
  - dados lógicos.

## Tipos de dados

- **Inteiro:** Números pertencentes ao conjunto dos números inteiros, números positivos ou negativos, não fracionados.

**Ex: 1, 0, -1, 230**

- **Real:** Números pertencentes ao conjunto dos números Reais, números positivos ou negativos, inclusive os fracionados.

**Ex: 1, 0, 1.2, -0.32**



## Tipos de dados

- **Caractere:** Também chamados alfanuméricos; contém sequências de caracteres como frases, endereços e outros, representados sempre entre aspas("").

**Ex: “Bom Dia!”, “Rua 45, quadra 07”, “35”**

- **Lógicos ou booleanos:** são valores verdadeiros ou falsos, sendo que esse tipo só aceita como entrada um dos dois valores.

**Verdadeiro, Falso**

## Variáveis

- São os dados que serão alterados pelo programa.
- Devem ser previamente declaradas e identificadas.
- Devem ser especificadas por tipos para que o computador possa fazer o uso correto dos valores delas nos momentos oportunos.
- É recomendável que os nomes das variáveis sejam os mais **significativos possíveis, isto é, que reflitam, da melhor** maneira possível, a natureza dos valores que nelas estão sendo armazenados

## Variáveis

Exemplos:

Variável	Conteúdo
valor	23
nome	Danilo
salário	2.500
fornecedor	skol comércio de bebidas

## **Variáveis: Regras de nomenclatura**

- Os nomes podem ser atribuídos com um ou mais caracteres
- O primeiro caractere não poderá ser um número.
- Não poderá possuir espaços em branco.
- Não poderá ter por nome uma instrução do programa.
- Não poderão ser utilizados outros caracteres que não sejam letras e números

## Variáveis: Regras de nomenclatura

Permitidos	Não permitidos
A	5b
nota	e)13
matricula	a;b
x5	x~y
a32b	nota.1
f1g3h5	b*d

## Constantes

- Uma constante é um determinado valor fixo que não se modifica ao longo do tempo.
- Uma constante pode ser um número, um valor lógico ou uma sequência de caracteres quaisquer com algum significado para o problema em estudo.

Exemplo:

```
Resultado <- entrada * 1.23
```



## Declaração de variáveis

- As variáveis só podem armazenar valores de um mesmo tipo de maneira que também são classificadas como sendo **real, inteiro, lógico ou caractere**.

*Como saber, então, qual o tipo de variável, ou seja, que conjunto de valores ela pode armazenar?*

- Para indicar o tipo de uma ou mais variáveis, é usa a **declaração de variáveis**

## Declaração de variáveis

- Uma vez declarada a variável, qualquer referência que se faça ao seu identificador implica a referência ao conteúdo do local da memória representado pelo mesmo.
- Toda declaração de variáveis tem a seguinte forma:

**var lista-de-identificadores: tipo-de-dados**

## Declaração de variáveis

Exemplo:

**var**

*a, b, c: inteiro*

*nome, endereco, cidade: caractere*

*salario, altura, peso: real*

*matricula\_regular: logico*

## Expressões Aritméticas

- Denomina-se expressão aritmética aquela cujos operadores são aritméticos e cujos operandos são constantes e/ou variáveis do tipo numérico.
- O conjunto de operações básicas adotado é o que se conhece da Matemática, a saber:

**Adição**

**Divisão**

**Subtração**

**Potenciação**

**Multiplicação**

**Radiciação**

## Operadores Aritméticos

**+** adição

Exemplos:  $5 + 2 = 7$ ,  $17 + 5 = 22$ ,  $48 + 9 = 57$

**-** subtração

Exemplos:  $5 - 2 = 3$ ,  $17 - 5 = 12$ ,  $48 - 9 = 39$

**\*** multiplicação

Exemplos:  $5 * 2 = 10$ ,  $17 * 5 = 85$ ,  $48 * 9 = 432$

**/** divisão de números

Exemplos:  $5 / 2 = 2.5$ ,  $17 / 5 = 3.4$ ,  $48 / 9 = 5.33$

## Operadores Aritméticos

**Div** divisão inteira

Exemplos:  $5 \text{ div } 2 = 2$ ,  $17 \text{ div } 5 = 3$ ,  $48 \text{ div } 9 = 5$

**Mod** resto de divisão inteira

Exemplos:  $5 \text{ mod } 2 = 1$ ,  $17 \text{ mod } 5 = 2$ ,  $48 \text{ mod } 9 = 3$

**^** potenciação

Exemplos:  $5 ^ 2 = 25$ ,  $8 ^ 3 = 512$

**Raizq raiz quadrada de um número**

Exemplos:  $\text{raizq}(25) = 5$ ,  $\text{raizq}(9) = 3$ ,  $\text{raiz}(49) = 7$



## Expressões Aritméticas

- Nas expressões aritméticas, as operações guardam entre si uma relação de prioridade, tal como na Matemática.
- A modularização é a divisão de uma expressão em partes, proporcionando maior compreensão e definindo prioridades para a resolução da mesma.
- Em expressões computacionais, utilizamos somente parênteses "(" e ")" para modularização.

## Expressões Aritméticas

- Na sintaxe do Português Estruturado podemos ter parênteses dentro de parênteses, como seriam os colchetes e as chaves na matemática.
- Os parênteses indicam quais sub-expressões, dentro de uma expressão, serão executados primeiro.
- A princípio, a execução é da esquerda para direita, mas além dos parênteses, existem prioridades entre os operadores envolvidos na expressão.

## Expressões Aritméticas – Prioridades

Operador Aritmético	Prioridade
Exponenciação	3 (maior)
Multiplicação	2
Divisão	2
Adição	1
Subtração	1 (menor)

## Expressões Aritméticas – Prioridades

Exemplos:

Expressão	Resultado
$2+2/2$	3
$(2 + 2) / 2$	2
$2*3-1$	5
$2 * (3 - 1)$	4
$2*5+2-6/3$	10
$2 * 5 +( 2 - 6) / 3$	8.67
$2 *( 5 + (2 - 6 ))/ 3$	0.67

## Expressões Lógicas




- É comum nos algoritmos surgirem situações em que a execução de uma ação, ou sequência de subações, está sujeita a uma certa condição.
- Esta condição é representada no texto de algoritmo por meio de uma expressão lógica.
- Denomina-se expressão lógica a expressão cujos operadores são lógicos e cujos operandos são relações, constantes e/ou variáveis do tipo lógico.

## Operadores Relacionais

Os operadores relacionais realizam a comparação entre dois operandos ou duas expressões e resultam em valores lógicos (VERDADEIRO ou FALSO).

Símbolo	Significado
=	Igual a
<>	Diferente de
>	Maior que
<	Menor que
>=	Maior ou igual a
<=	Menor ou igual a

## Operadores Lógicos

- **e**  é o operador usado quando é necessário que uma condicional atenda obrigatoriamente mais de uma exigência.
- **ou**  operador usado caso seja necessário que uma condicional atenda opcionalmente uma ou mais exigências.
- **nao**  operador usado para negar uma exigência, ou seja para executar uma informação quando as exigências sejam descumpridas.



## Operadores Lógicos

Exemplos:

A	B	A ou B	A e B	não A
V	V	V	V	F
V	F	V	F	F
F	V	V	F	V
F	F	F	F	V

## Expressões Lógicas – Prioridades

Operador Lógico	Prioridade
e	3
ou	2
não	1

## Expressões Literais

- Uma expressão literal é aquela formada por operadores de caracteres e operandos que são constantes e/ou variáveis do tipo caractere.

## Operador Literal

- +** Operador de concatenação de caracteres (isto é, cadeias de caracteres), quando usado com dois valores (variáveis ou constantes) do tipo "caractere".

Por exemplo: "Rio " + " de Janeiro" = "Rio de Janeiro".

## Operadores – Prioridade

Entre as categorias de operadores também há prioridades, conforme mostrado na tabela abaixo:

Operador	Prioridade
Operadores Aritméticos	3
Operadores Relacionais	2
Operadores Lógicos	1