

# Introdução a Programação

Profs: Saulo e Suelen

# Algoritmo

- Conjunto finito de regras (que provê uma sequência de operações) para resolver um tipo de problema específico.
  - Exemplos:
    - Maneira como tomamos banho
    - Instruções para manusear um aparelho
    - Receita de bolo

# Programa

- Formalização de um algoritmo em uma determinada **linguagem de programação**, segundo suas regras de sintaxe e semântica, de **forma a permitir que o computador possa entender a sequência de ações**;

# Linguagem de Programação

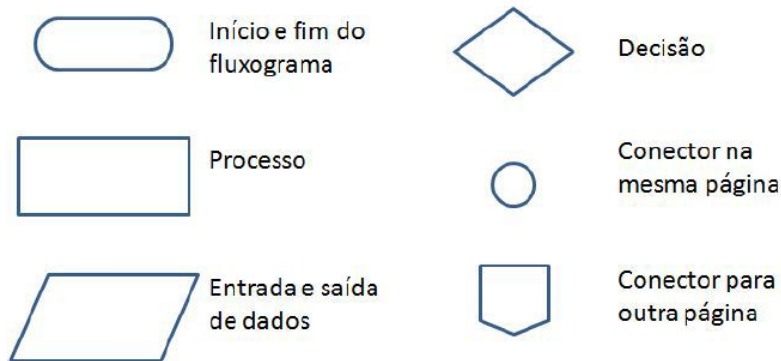
- Método padronizado para expressar instruções para um computador.
- É um conjunto de regras sintáticas e semânticas usadas para definir um programa de computador.
  - Exemplo: C, C++, C#, Java, Python

**Para que o algoritmo possa se tornar um programa executável é necessário escrevê-lo em uma linguagem de programação real, seguindo uma determinada estrutura. Assim, o computador será capaz de “entender” o algoritmo e executá-lo.**

# Algoritmos Não Estruturados

- São aqueles que buscam resolver problemas através de conjunto de instruções/regras que não seguem uma organização lógica.
- Podem ser descritos:
  - Narrativa
    - manual de instruções

- Fluxograma:



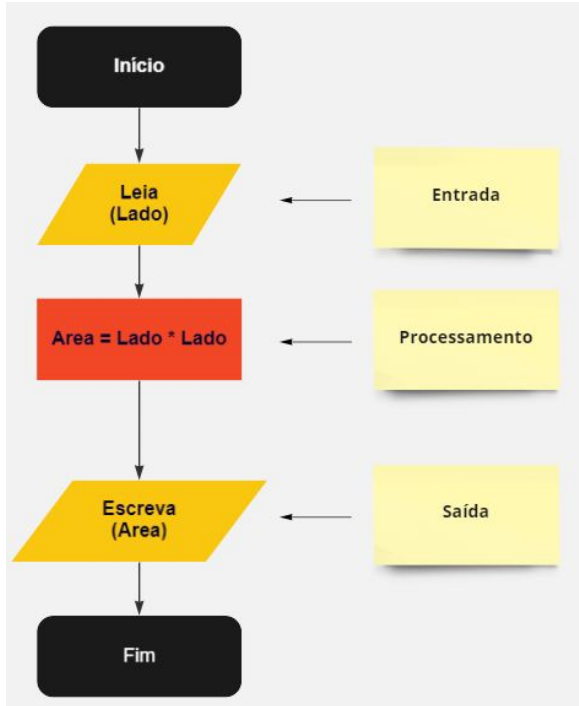
# Algoritmos Estruturados

- São aqueles que buscam resolver problemas através do uso de um computador, portanto são descritos através de uma linguagem de programação e seguindo uma organização lógica.
- Exemplos:
  - Algoritmos representados em linguagem java;
  - Algoritmos representados em linguagem C;
  - **Algoritmos representados em Português Estruturado (Portugol/Pseudocódigo).**

# Por que usar Portugol primeiro?

- Simplicidade da linguagem (sintaxe/semântica) apropriada para iniciantes;
- Palavras-chave e comandos (vocabulário) em Português;
- Foco no desenvolvimento da habilidade de resolver problemas usando lógica de programação;
- Semelhança com as linguagens de programação profissionais.

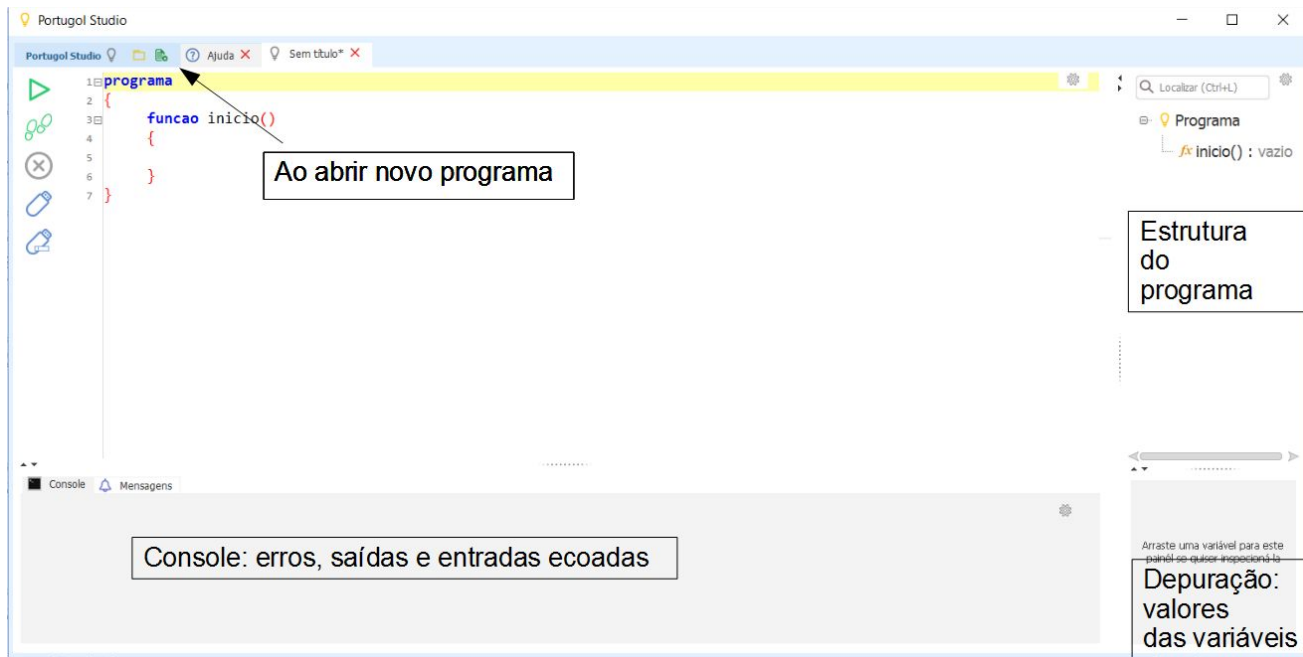
# Algoritmo para calcular a área de um quadrado



- Como esse algoritmo é escrito em português?
- Onde pode ser escrito?



# Portugol Studio - IDE (Integrated Development Environment)



Disponível gratuitamente para download de: <http://lite.acad.univali.br/portugol/>.

# Outros exemplos

- Crie um programa em portugol para:
  - Converter o valor de uma temperatura em graus Celsius para graus Fahrenheit (  $F = (9 \cdot C + 160) / 5$  );
  - Calcular o perímetro e a área de um quadrado cujo lado é informado pelo usuário;
  - Solicitar ao usuário 4 números. Calcular e escrever a soma e o produto;
  - Solicitar ao usuário a altura e base de um retângulo e escrever seu perímetro e sua área.

# Conceito de Variável

- Variável é um local na memória do computador que armazena um conteúdo de um tipo específico;
- Contém um valor que se modifica durante a execução de um programa;
- É identificada por um nome (identificador), que pode ser representado:

**tipo** `identificador`

# Tipos primitivos

- **Inteiro:** qualquer informação numérica que pertença ao conjunto dos números inteiros (negativa, nula ou positiva).

Exemplo: 100, 0, 1, 2, 1250

- **Real:** qualquer informação numérica que pertença ao conjunto dos números reais (negativo, nula ou positiva).

Exemplo: -10, -1.5, 11.2, 0, 1, 2, 50

- **Caracter:** qualquer informação composta por um único caractere alfanumérico (0..9, a .. z, A..Z) e/ou símbolos especiais (por exemplo: \*, \$, %, &, #, ?, >, <, !, @).

Exemplo: A, &,

# Tipos primitivos

- **Cadeia:** qualquer informação composta por um conjunto de caracteres alfanuméricos (0..9, a .. z, A..Z) e/ou símbolos especiais (por exemplo: \*, \$, %, &, #, ?, >, <, !, @).

Exemplo: casa, Win31, 123

- **Lógico:** qualquer informação que possa apenas assumir duas situações possíveis (biestável). Verdadeiro ou Falso.

Exemplo:  $x > y$  ?

# Declaração de Variável

- **inteiro** X, Y, Idade, QtdFuncionarios
- **real** Salario, Peso, Dolar, ValorVenda, NotaMedia
- **character** formapagamento, sexo, estadocivil
- **cadeia** Nome, endResidencial, endProfissional,  
Data\_Nascimento
- **logico** TemDependente, moraOuroBranco

# Dicas para nomear variáveis

- Devem começar por um caractere alfabético (letra);
- Pode ser seguido por mais caracteres alfabéticos e/ou numéricos, ou pelo caractere sublinhado “\_”
- Não é permitido o uso de caracteres especiais (@, #, \$, % “, ^, ~, ...);
- Costuma-se usar maiúsculas e minúsculas para separar palavras:  
Ex: "PesoDoCarro";
- Deve ser diferente dos comandos da linguagem;
- Evitar nomes longos, com mais de 31 caracteres;
- Pode conter números a partir do segundo caractere.
- Use nomes coerentes/significativos ao que vai ser armazenado.

# Constante

- Uma constante é um valor fixo que não se modifica ao longo do tempo;
- Em pseudocódigo representaremos constantes da seguinte maneira:

```
const inteiro MAIORIDADE = 18
```



# Comando de Entrada

- **Comando leia():** lê, do meio externo, a informação disponível para leitura e armazena na variável discriminada após o comando, entre parênteses.
  - leia(variável\_1)
    - **Ex:** **leia (v)** //O valor da variável (v) é dado por um dispositivo de entrada e armazenado em memória.

# Comando de Saída

- **Comando escreva():** Imprime na tela as mensagens e conteúdo da(s) variável(eis) especificada(s) após o comando, entre parênteses.
  - Ex: **escreva (x)** //O valor atual da variável (x) é informado para um dispositivo de saída.
  - Ex: **escreva (" O valor do resultado é: ", x)**

# Operadores Aritméticos

Operador	Descrição	Exemplo
+	soma	$a + b$
-	subtração	$a - b$
*	multiplicação	$a * b$
/	divisão	$a / b$
/	divisão inteira (se os dois operandos são inteiros)	inteiro x, y ... $x = 5$ $y = 2$ $x / 2$ // será igual a 2
%	módulo da divisão (se os dois operandos são inteiros)	inteiro c = 6 ... $c \% 4$ // será igual a 2

# Precedência de Operadores

- Para mudar a precedência usamos parênteses ( e ), como nos exemplos:

```
funcao inicio()
{
    real resultado

    // Neste exemplo, a operação de multiplicação (*) será executada primeiro
    resultado = 5.0 + 4.0 * 2.0
    escreva("Operação: 5 + 4 * 2 = ", resultado)

    // Neste exemplo, a operação de soma (+) será executada primeiro
    resultado = (5.0 + 4.0) * 2.0
    escreva("\nOperação: (5 + 4) * 2 = ", resultado)

    /*
     * Neste exemplo, a operação de divisão (/) será executada primeiro,
     * seguida pela operação de multiplicação (*). Por último, será
     * executada a operação de soma (+)
     */
    resultado = 1.0 + 2.0 / 3.0 * 4.0
    escreva("\nOperação: 1 + 2 / 3 * 4 = ", resultado)

    /*
     * Neste exemplo, a operação de soma (+) será executada primeiro,
     * seguida pela operação de multiplicação (*). Por último, será
     * executada a operação de divisão (/).
     */
    resultado = (1.0 + 2.0) / (3.0 * 4.0)
    escreva("\nOperação: (1 + 2) / (3 * 4) = ", resultado, "\n")
}
```

# Precedência de Operadores

Devemos obedecer a precedência dos operadores da seguinte forma:

- **Sem uso de parênteses:**

- 1º ) / e %
- 2º ) \*
- 3º ) +

# Funções Matemáticas

- As funções de Potência e Raiz fazem parte da biblioteca mat do Portugol Studio.

```
programa
{
    inclui biblioteca Matematica --> mat // Inclui a biblioteca Matemática
```

- E para usar, é necessário colocar o nome da biblioteca mat, seguida de ponto (.) e o nome da função.

```
mat.potencia(valor, 3.0) // Calcula o valor elevado ao cubo
```

```
mat.raiz(4, 2.0) // Calcula a raiz quadrada de 4
```

# Referências

- Programação de Computadores I - 2011.1
  - Algoritmos: Conceito e Representação
  - Profº José Romildo Malaquias
  - Departamento de Computação - UFOP