

# Fundamentos de Programação

Prof. Romário Nzenguele da Silva



**Licenciatura em Informática**

# Sobre o Documento



Material de apoio **gratuito**, desenvolvido exclusivamente para os estudantes do **1º Ano da Licenciatura em Informática**;



O conteúdo, sempre que necessário, será atualizado para refletir as mudanças e melhorias no decorrer do curso. Cada atualização será devidamente versionada, permitindo o fácil acompanhamento das revisões;



Este recurso é um **resumo das aulas**, essencial para apoiar no aprendizado da disciplina.  
O **repositório da disciplina** (contendo exercícios, conteúdos detalhados, ...) será reencaminhado por email;



Sugestões de melhoria podem ser apresentadas pelo email que se encontra no rodapé.

# Conteúdo

## 3. Estruturação de Programas e Sintaxe

3.1. Introdução ao Portugol Studio

3.2. Estrutura de um Programa

3.3. Introdução à Sintaxe da Linguagem Algorítmica

3.4. Sintaxe e Semântica

3.5. Palavras Reservadas, Símbolos, Identificadores e Valores

### Objectivos

- ✓ Familiarizar-se com o Portugol Studio e com a estrutura básica de um programa.
- ✓ Entender e diferenciar a sintaxe e semântica na linguagem algorítmica.
- ✓ Entender o conceito e/ou uso de palavras reservadas, símbolos, identificadores e valores.



# 3. Estruturação de Programas e Sintaxe

**Suporte:** Portugol (Portugol Studio)

# 3.1. Introdução ao Portugol Studio

**Ferramenta educacional** desenvolvida para **auxiliar estudantes iniciantes em programação.**

- ❖ É um ambiente de desenvolvimento que utiliza uma **linguagem em português estruturado (portugol)**, facilitando o aprendizado dos conceitos fundamentais de programação.
- ❖ **Desenvolvido e mantido pelo [LITE/UNIVALI](#)**
- ❖ **Características principais:**
  - ✓ Interface gráfica intuitiva
  - ✓ Língua portuguesa
  - ✓ Depurador (debugger) integrado
  - ✓ Console para entrada/saída de dados
  - ✓ Recursos visuais para auxiliar no aprendizado



## 3.2. Estrutura de um Programa

### ❑ Elementos da estrutura:

- **programa**: palavra reservada que indica o início do código;
- **inicio()**: esta função é o ponto de partida da execução
- **{ }**: delimitam blocos de código
- **//**: comentários. Para múltiplas linhas utiliza-se **/\* \*/**.

```
programa {  
  
    funcao inicio() {  
        // Aqui vai o código do programa  
    }  
}
```

```
//O comando programa é obrigatório  
programa  
{  
    //Inclusões de bibliotecas  
    // - Quando houver a necessidade de utilizar  
    //  uma ou mais bibliotecas, as inclusões  
    //  devem aparecer antes de qualquer declaração  
  
    /*  
     * Dentro do programa é permitido declarar  
     * variáveis globais, constantes globais e  
     * funções em qualquer ordem.  
     */  
  
    //Declarações de funções somente  
    //são permitidas dentro do programa.  
    funcao inicio()  
    {  
        /*  
         * Declarações de variáveis locais,  
         * constantes locais, estruturas de  
         * controle e expressões.  
         */  
    }  
}
```

Fonte: Portugol Studio

## 3.3. Introdução à Sintaxe da Linguagem Algorítmica

### ❑ Elementos sintáticos básicos:

- ✓ Declaração de variáveis;
- ✓ Comandos de entrada/saída;
- ✓ Estruturas de controle (condicionais/cíclicas);
- ✓ Operadores;
- ✓ Expressões.

**Nota:** terá um espaço para abordar sobre esses elementos.

```
programa {  
    funcao inicio() {  
  
        inteiro numero  
  
        escreva("Digite um numero: ")  
        leia(numero)  
  
        se (numero > 0) {  
            escreva("Numero positivo")  
        } senao {  
            escreva("Numero negativo ou zero")  
        }  
    }  
}
```

**Obs:** Ver o equivalente nos slides 13, 14 e 15 do **M02 v1.0**

## 3.4. Sintaxe e Semântica

❑ **Sintaxe:** regras gramaticais da linguagem -> **fórmula**

- Como escrever comandos
- Uso correto de pontuação
- Estrutura das instruções

❑ **Semântica:** significado das instruções

- O que cada comando faz
- Como as instruções são interpretadas
- Resultado esperado da execução

```
programa {  
    funcao inicio() {  
        inteiro a = 5      // Sintaxe: declaração e atribuição  
                           // Semântica: criar variável e guardar valor 5  
  
        escreva(a + 3)     // Sintaxe: comando de saída com expressão  
                           // Semântica: mostrar resultado da soma (8)  
    }  
}
```



## 3.5. Palavras Reservadas, Símbolos, Identificadores e Valores

### ❑ Palavras Reservadas

- ✓ São palavras que têm significado especial para o compilador/interpretador;
- ✓ Não podem ser usadas como **identificadores** (nomes de variáveis, funções, etc.);
- ✓ São escritas **geralmente** em minúsculas;
- ✓ Possuem função específica na linguagem.

### ❖ Características principais:

- ✓ São predefinidas pela linguagem;
- ✓ Não podem ser redefinidas;
- ✓ Têm sintaxe específica de uso;
- ✓ São case-sensitive em muitas linguagens.

```
programa {  
    // Correto - usando palavra reservada  
    inteiro numero = 10  
  
    // Incorreto - tentando usar como variável  
    // inteiro se = 20 // Erro! 'se' é palavra reservada  
  
    funcao inicio() {  
        // Correto - usando em estrutura condicional  
        se (numero > 0) {  
            escreva("Positivo")  
        }  
    }  
}
```

## 3.5. Palavras Reservadas, Símbolos, Identificadores e Valores

### ❑ Símbolos

- ✓ São caracteres especiais usados para operações e organização do código

#### 1. Operadores Aritméticos:

- + (soma)
- - (subtração)
- \* (multiplicação)
- / (divisão)
- % (resto da divisão)

#### 2. Operadores Relacionais:

- > (maior que)
- < (menor que)
- >= (maior ou igual)
- <= (menor ou igual)
- == (igual)
- != (diferente)

#### 3. Operadores Lógicos:

- e (AND lógico)
- ou (OR lógico)
- nao (NOT lógico)

#### 4. Símbolos de Organização:

- { } (delimita blocos de código)
- ( ) (agrupa expressões)
- , (separa elementos)

## 3.5. Palavras Reservadas, Símbolos, Identificadores e Valores

### ❑ Identificadores

- ✓ **Nomes definidos pelo programador para identificar:** variáveis, procedimentos (funções), constantes, ...

#### ❖ Regras importantes:

- ✓ Devem começar com letra
- ✓ Podem conter letras, números e \_
- ✓ Não podem conter espaços
- ✓ Não podem ser palavras reservadas
- ✓ São sensíveis a maiúsculas/minúsculas

#### Exemplos válidos:

idade  
nome\_completo  
salarioBase  
contador1

#### Exemplos inválidos:

1contador // começa com número  
nome completo // contém espaço  
se // palavra reservada

### ❑ Valores

- ✓ **Dados atribuídos (armazenados) nas variáveis.**
- ✓ **A atribuição depende do tipo de dados definido.**
  - Ex: nota de um aluno deve ser declarada com um tipo de dados numérico.



# ANEXOS

# Ferramentas [Portugol Studio]

- ✓ Site Oficial: <https://lite.acad.univali.br/portugol-studio/>
- ✓ Download: <https://github.com/UNIVALI-LITE/Portugol-Studio>
- ✓ Ambiente Online: <https://portugol.dev/>

# Ferramentas [semelhantes]

- ✓ **Visualg:** <https://visualg30.yolasite.com/>
- ✓ **Portugol IDE:** <http://orion.ipt.pt/~manso/Portugol/download.html>
- ✓ **[ES] Pse Int:** <https://pseint.sourceforge.net/index.php?page=portada.php>
  - Sobre a ferramenta: <https://pseint.sourceforge.net/slide/pseint.html>
- ✓ **[Para Fluxogramas – EN] Flowgorithm :** <https://www.flowgorithm.org/index.html>
- ✓ **[FR] Algobox:** <https://www.xm1math.net/algobox/>

# Recomendações

- ✓ [Cadastro] **GitHub**: <https://github.com/signup?source=login>
- ✓ [Download] **Git**: <https://git-scm.com/downloads>
- ✓ [Tutorial] **GitHub/Git**: <https://docs.github.com/pt/get-started/start-your-journey>

...

Sequência -> **M04**

---