

# Lógica de Programação

## Aula 03

Professor: Henrique Augusto Maltauro

# Pseudocódigo

# Pseudocódigo

O **pseudocódigo** consiste em escrever os passos que devem ser seguidos para a resolução do problema, utilizando **regras e comandos pré-definidos**, beirando a programação de sistemas.

# Pseudocódigo

Essas **regras** são o que nós chamamos de **lógica de programação**.

E os **comandos** são o que nós chamamos de **sintaxe**.

Cada **linguagem de programação** vai possuir uma **sintaxe diferente**, mas a **lógica** permanece a mesma para todas.

Basicamente **tudo** o que a gente vai daqui pra frente, vai ter algo equivalente nas **linguagens de programação**.

# Pseudocódigo

**Vantagem:** a passagem do pseudocódigo para qualquer linguagem de programação é muito simples, bastando conhecer a sintaxe da linguagem de programação escolhida.

**Desvantagem:** é necessário aprender as regras e comandos do pseudocódigo.

# Pseudocódigo

[dgadelha.github.io/Portugol-Webstudio/](https://github.com/dgadelha/Portugol-Webstudio/)

# Pseudocódigo

- Palavras Reservadas

O pseudocódigo possui algumas **palavras reservadas**, ou também chamadas de **palavras-chaves**.

Essas palavras são os **comandos** que o pseudocódigo vai utilizar para executar alguma coisa em específico.

# Pseudocódigo

- Palavras Reservadas

É muito importante ter em mente que, no **pseudocódigo**, assim como na maior parte das **linguagens de programação**, essas palavras são **case-sensitive**.

Ou seja, a palavra reservada **programa** não pode ser escrita como **Programa**, ou **PROGRAMA**.



# Pseudocódigo

- Palavras Reservadas: **programa**

O **Portugol Webstudio**, utiliza a **palavra reservada programa**, para definir o **bloco de código** que vai fazer parte daquele **pseudocódigo**.

Tudo o que estiver dentro desse **bloco de código**, vai ser considerado na execução do **pseudocódigo**.

# Pseudocódigo

- Delimitadores

O pseudocódigo também possui os chamados **delimitadores**, que serão caracteres como:

- chaves { }
- colchetes [ ]
- parênteses ( )
- aspas simples ' '
- aspas duplas " "

# Pseudocódigo

- Delimitadores

Esses **delimitadores** são utilizados para definir o **início e o fim de alguma coisa**.

# Pseudocódigo

- Delimitadores: **chaves { }**

Os **delimitadores** de **chaves**, são utilizados em geral para definir em qual linha um **bloco de código** começa, e em qual linha um **bloco de código** termina.

# Pseudocódigo

- Palavras Reservadas: **programa**

Uma vez que a **palavra reservada** `programa` define um **bloco de código**, ela precisa ser acompanhada dos **delimitadores de chaves**.

# Pseudocódigo

- Palavras Reservadas: **programa**

```
programa {  
    // Bloco de código  
}
```

# Pseudocódigo

- Comentários

O pseudocódigo também possui uma estrutura de **comentários**.

Esses **comentários** nos permitem fazer **anotações no código**, que podem ser úteis para compreender alguma questão a respeito da lógica implementada, e que serão **ignorados na execução do código**.

# Pseudocódigo

- Comentários

Temos duas maneira de fazer **comentários**:

- **Comentários de uma única linha, utilizando // antes do comentário**
- **E comentários de várias linhas utilizando /\* \*/ como um delimitador de onde começa e onde termina o comentário**



# Pseudocódigo

- Comentários

*// Comentário de uma única linha*

# Pseudocódigo

- Comentários

```
/*  
    Comentário de  
    várias linhas  
*/
```

# Pseudocódigo

- Funções

O pseudocódigo também possui uma estrutura de **funções**.

Essas **funções**, são pequenos **blocos de código**, que ficam obrigatoriamente dentro do **bloco de código** do **programa**.

Para definir essas **funções** é necessário a **palavra reservada** **funcao**, seguida do **nome da função** e de dois **delimitadores**.

# Pseudocódigo

- Funções

1. Delimitador de parênteses ( ), o qual é utilizado para definir informações externas que a função pode ou não receber
2. Delimitador de chaves { }, para definir o bloco de código da função.

# Pseudocódigo

- Funções

Nós só vamos nos aprofundar melhor no uso das **funções** mais pra frente.

Porém eu já estou explicando agora porque o **Portugol Webstudio** obrigatoriamente precisa de uma **função** chamada **inicio** para poder realizar a execução do código.

# Pseudocódigo

- Funções

```
programa {  
    funcao inicio() {  
        // Bloco de código  
    }  
}
```

# Pseudocódigo

Aqui vem um ponto muito importante que se aplica, tanto no pseudocódigo, como nas linguagens de programação em geral.

Você deve ter percebido que a palavra reservada funcao não se utiliza de caracteres especiais.

# Pseudocódigo

**Não se deve utilizar caracteres especiais** nos códigos de programação, para evitar problemas de conversão de arquivos.

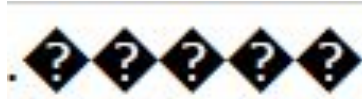
Com **exceção** de **comentários** e informações que estejam dentro dos **delimitadores** de **aspas simples** ou de **aspas duplas**.



# Pseudocódigo

Юуһьє ЁрчЁрсюёушьш яЁюЁрьёюю юсхёяхүхэш т/сшЁр■Ё Java  
тхїэюьюуш Java с/ыр яЁюЁхёЁшЁютрэр, фюЁрсюЁрэр, фюяюэхэр ш яЁютхЁхэр ёююс·хёЁтую ёяхүшрьшёт

эряшёрэшх яЁюЁрьёюю юсхёяхүхэш эр юфёюц яырЁіюЁьх ш хую чряеёь яЁрьЁшүхёьш эр ь■сюц фЁеу»  
ёючфрэшх яЁюЁрьё, ЁрсюЁр■шї т тхс-сЁречхЁх ш шьх■шї фюёёя ь тхс-ёыецсрь  
ЁрчЁрсюёьр яЁшыюцхэшц эр ёёюЁюэх ёхЁтхЁр фы ІюЁеют т ЁэхЁэхЁх, Ёрурчшэют, юяЁюют, юсЁрсюё  
юс·хфшэхэшх яЁшыюцхэшц шыш ёыецс ё шёяюьМчютрэшхь ч/ьр Java фы ёючфрэш т/ёююёяхүшрьшчшЁют  
эряшёрэшх ьёююІеэьүшюэрьМэвї ш шїІхёштэвї яЁшыюцхэшц фы ьюсшыМэвї ЁхыхІюэют, ефрыхээвї яЁш



000000 000000000000 000000000000 000000000000 00000000 Java  
0000000000 Java 0000 00000000000000, 0000000000, 000000000 0 000000000 000000000000 000000000

000000000 000000000000 00000000000 00 00000 000000000 0 000 000000 000000000000 00 00000 00000  
000000000 00000000, 0000000000 0 000-00000000 0 0000000 000000 0 000-0000000  
0000000000 0000000000 00 0000000 00000000 000 0000000 0 0000000000, 000000000, 0000000, 0000000  
00000000000 0000000000 000 00000 0 00000000000000 00000 Java 000 00000000 000000000000000000  
000000000 000000000000000000 0 00000000000 0000000000 000 000000000 000000000, 000000000 000

# Pseudocódigo

- Funções Internas

O pseudocódigo também possui uma estrutura de **funções internas**, que irão realizar algum processo já pronto dentro do pseudocódigo.

Essas **funções internas** são executadas através do **nome dessas funções**.

# Pseudocódigo

- Funções Internas: **escreva**

A **função interna** **escreva** vai escrever alguma coisa em tela.

Depois de digitado o nome da função **escreva**, é utilizado o **delimitador** de **parênteses ( )**.

Dentro desses **parênteses**, é informado **O QUE** vai ser exibido em tela, podendo ser a informação de um **número** ou de um **texto**.

# Pseudocódigo

- Funções Internas: **escreva**

Para exibir **números**, basta informar o número dentro dos **parênteses**.

Para exibir um **texto**, esse texto precisa estar dentro de um **delimitador** de **aspas duplas** “ ”.

# Pseudocódigo

- Funções Internas: **escreva**

```
programa {  
    funcao inicio() {  
        escreva(5)  
    }  
}
```

# Pseudocódigo

- Funções Internas: escreva

```
programa {  
    funcao inicio() {  
        escreva("Texto")  
    }  
}
```

# Pseudocódigo

- Funções Internas: **escreva**

Caso nós quisermos escrever em tela, mais de uma informação diferente, nós informamos todos eles dentro dos **parênteses**, separados por **vírgulas**.

# Pseudocódigo

- Funções Internas: escreva

```
programa {  
    funcao inicio() {  
        escreva("Ola, eu tenho ", 29, " anos!")  
    }  
}
```



# Pseudocódigo

- Funções Internas: escreva

```
programa {  
    funcao inicio() {  
        escreva("Ola, eu tenho 29 anos!")  
    }  
}
```

# Pseudocódigo

- **Funções Internas:** escreva

Se nós utilizarmos mais de uma **função escreva**, as informações delas vão ser apresentadas na mesma linha.

Para conseguirmos quebrar linhas, nós utilizamos dentro do texto um **comando ASCII** para quebra de linha, que é o **\n**.

# Pseudocódigo

- Funções Internas: escreva

```
programa {  
    funcao inicio() {  
        escreva("Ola, meu nome é...\n")  
        escreva("Henrique")  
    }  
}
```

# Pseudocódigo

- Variáveis

As **variáveis** são **informações nomeadas**, que nós conseguimos armazenar na memória de execução do nosso código.

Para utilizarmos as **variáveis**, precisamos primeiro entender duas coisas:

- Tipos de Dados
- Identificadores

# Pseudocódigo

- Tipos de Dados

Os **tipos de dados** são utilizados para definir qual é o tipo de informação que uma **variável** vai armazenar.

No pseudocódigo temos 4 **tipos de dados** diferentes:

- **caracter**
- **cadeia**
- **inteiro**
- **real**

# Pseudocódigo

- Tipos de Dados: **caracter**

O **tipo de dado** de **caracter** vai armazenar apenas um **único caractere**, seja uma letra ou seja um número.

A informação do **caracter** vai ser representada utilizando o **delimitador** de **aspas simples ' '**.

# Pseudocódigo

- Tipos de Dados: **cadeia**

O **tipo de dado** de **cadeia** vai armazenar uma **sequência**, ou uma cadeia, **de caracteres**, sejam letras ou sejam números.

A informação da **cadeia** vai ser representada utilizando o **delimitador** de **aspas duplas “ ”**.

# Pseudocódigo

- Tipos de Dados: inteiro

O tipo de dado de inteiro vai armazenar um número inteiro.



# Pseudocódigo

- Tipos de Dados: **real**

O **tipo de dado** de **real** vai armazenar um **número com casas decimais**.

As casas decimais, ou seja, os números depois da vírgula, devem estar **separados não por uma vírgula, mas por um ponto**.

# Pseudocódigo

- Identificadores

Os **identificadores** são os nomes das variáveis.

Nomes esses que não podem repetir no mesmo **bloco de código**.

# Pseudocódigo

- Variáveis

Uma vez entendido isso, para definirmos uma **variável** é necessário informar primeiro o **tipo de dado**, e logo depois o **identificador** da **variável**.

E para atribuir um valor para essa **variável**, nós utilizamos um  **sinal de igual =**.

# Pseudocódigo

- Variáveis

```
caracter letra = 'A'  
cadeia texto = "Verdade"  
inteiro numero = 5  
real valor = 9.69
```

# Pseudocódigo

- Variáveis

```
cadeia texto = "A Verdade"  
escreva(texto)
```

# Pseudocódigo

- Funções Internas: **leia**

A **função interna leia** é utilizada para receber de fora do código uma informação que será atribuída a uma **variável**.

Depois de digitado o nome da **função leia**, é utilizado o **delimitador de parênteses ( )**.

Dentro desses **parênteses**, é informado qual é a **variável** que irá receber o valor de fora.

# Pseudocódigo

- Funções Internas: **leia**

```
cadeia nome  
leia(nome)  
escreva("Ola, meu nome é ", nome)
```

# Exercício



# Exercício

[gg.gg/LogicaSenac03](https://gg.gg/LogicaSenac03)

[github.com/hmaltaurodev/Slides](https://github.com/hmaltaurodev/Slides)