

# Lógica de Programação



Professor : Dr. Xiankleber Cavalcante Benjamim  
E-mail: [prof.xiankleber@gmail.com](mailto:prof.xiankleber@gmail.com)

# Sumário

## Lógica de Programação

- Narrativa
- Fluxograma
- Diagrama de Chapin
- Pseudocódigo

# Lógica de Programação

Descrição Narrativa.

Fluxograma.

Diagrama de Chapin.

Pseudocódigo, também conhecido como Português Estruturado ou Portugol.”

# Lógica de Programação Narrativa

Nessa forma de representação, os algoritmos são expressos diretamente em linguagem natural.

Para escrever um

algoritmo, precisamos descrever a sequência de instruções, de maneira simples e objetiva. Para isso devemos obedecer algumas regras básicas:

# Lógica de Programação

Usar somente um verbo por frase;

- Imaginar que você está desenvolvendo um algoritmo para pessoas que não trabalham com informática;

- Usar frases curtas e simples;

Ser objetivo;

- Procurar usar palavras que não tenham sentido dúbio

# Lógica de Programação

Como exemplo, têm-se os algoritmos seguintes:

Troca de um pneu furado:

1. Afrouxar ligeiramente as porcas.
2. Suspender o carro.
3. Retirar as porcas e o pneu.
4. Colocar o pneu reserva.
5. Apertar as porcas.
6. Abaixar o carro.
7. Dar o aperto final nas porcas..

# Lógica de Programação

Como exemplo, têm-se os algoritmos seguintes:

Cálculo da média de um aluno:

1. Obter as notas da primeira e da segunda prova.
2. Calcular a média aritmética entre as duas notas.
3. Se a média for maior ou igual a 7, o aluno foi aprovado, senão ele foi reprovado.

# Lógica de Programação

Como exemplo, têm-se os algoritmos seguintes:

## Somar três números

Passo 1: Receber três números

Passo 2: Somar os três números

Passo 3: Mostrar o resultado obtido



# Lógica de Programação

- Ponto positivo: Não é necessário aprender novos conceitos, pois a língua natural já é bem conhecida.
- Ponto negativo: A língua natural abre espaço para várias interpretações, dificultando a transcrição desse algoritmo para programa

# Lógica de Programação

Crie um algoritmo para exibir o resultado da multiplicação de dois números.

Descrição narrativa:

- PASSO 1 – Receber os dois números que serão multiplicados
- PASSO 2 – Multiplicar os números
- PASSO 3 – Mostrar o resultado obtido da multiplicação

## Atividade

Crie um algoritmo para exibir o dobro  
( $n*2$ ) de um número

# Atividade

Descrição narrativa:


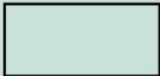
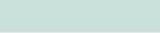




- PASSO 1 – Receber um número
- PASSO 2 – Multiplicar o número \* 2
- PASSO 3 – Mostrar o resultado obtido da multiplicação

FLUXOGRAMA

# Fluxograma

É a forma gráfica de representar um algoritmo mais conhecida e utilizada. O fluxograma nos permite mostrar graficamente a lógica de um algoritmo, enfatizando passos individuais e o fluxo de execução.

# Fluxograma

FIGURA	SIGNIFICADO
	Figura para definir início e fim do algoritmo
	Figura usada no processamento de cálculo, atribuições e processamento de dados em geral
	Figura utilizada na representação de entrada de dados
	Figura utilizada para representação da saída de dados
	Figura que indica o processo seletivo ou condicional, possibilitando o desvio no caminho do processamento
	Símbolo geométrico usado como conector
	Símbolo que identifica o sentido do fluxo de dados, permitindo a conexão entre as outras figuras existentes

**Figura 2.1: Principais símbolos utilizados em um fluxograma**

Fonte: adaptado de Sampaio, 2008

# Fluxograma

Calcular a média do aluno:

Vamos discutir passo a passo a criação de um fluxograma para representar um algoritmo que permita calcular a média final de um aluno considerando que todo aluno realiza três provas no semestre. O aluno é considerado “aprovado” se a sua média for igual ou superior a 7 (sete), senão é considerado “reprovado”.

- Se a média for  $\geq 7$  Aprovado
- Se a média for  $\leq 7$  Reprovado



# Fluxograma

Antes de iniciarmos a construção do fluxograma, devemos formalizar o algoritmo para solucionar o problema.

Para resolver esse problema, fica claro que precisaremos conhecer as três notas do aluno, que deverão ser informadas pelo usuário e, em seguida, calcular a média por meio do cálculo de média aritmética simples para a qual devemos aplicar a fórmula  $média = (nota1 + nota2 + nota3) / 3$ .

# Fluxograma

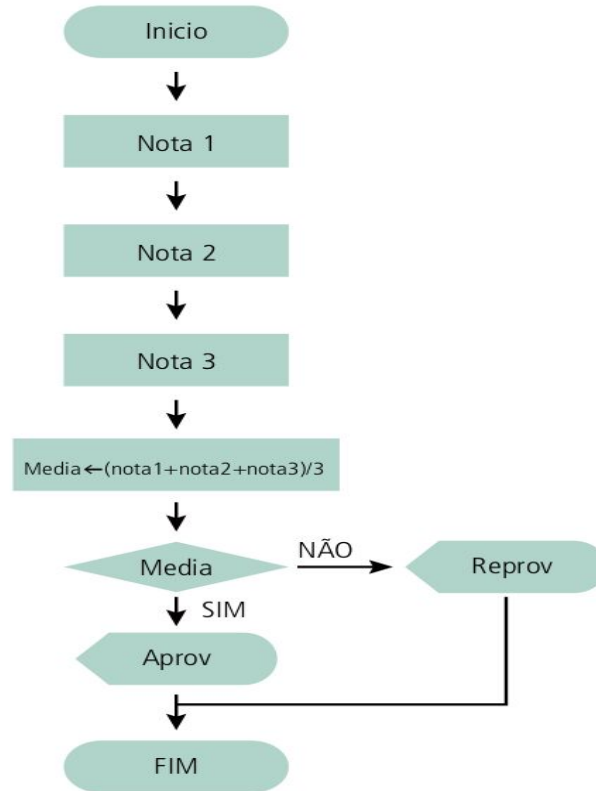
Em um fluxograma, devemos sempre representar a atribuição de valor a uma variável por meio do símbolo “←”, na forma variável = valor.

Em um fluxograma, a expressão variável = valor representa o teste se o valor da variável é igual ao valor informado do lado direito da expressão e não uma atribuição.

# Fluxograma

Agora que o valor da média já foi calculado, é hora de testar o seu valor a fim de definir se o aluno foi aprovado ou reprovado. Nesse momento, sentimos a necessidade de controlar o fluxo de instruções, pois caso o valor da média seja superior a 7 (sete) devemos executar a instrução de impressão da mensagem “Aprovado”, senão devemos apresentar a mensagem “Reprovado”. Para fazer isso é necessário utilizar o símbolo de decisão, como mostrado na Figura 2.5. Como podemos notar, dependendo do resultado da condição  $média \geq 7$ , o fluxo de instruções é devidamente desviado.

# Fluxograma



**Figura 2.5: Fluxograma final**

Fonte: Adaptado de Sampaio, 2008

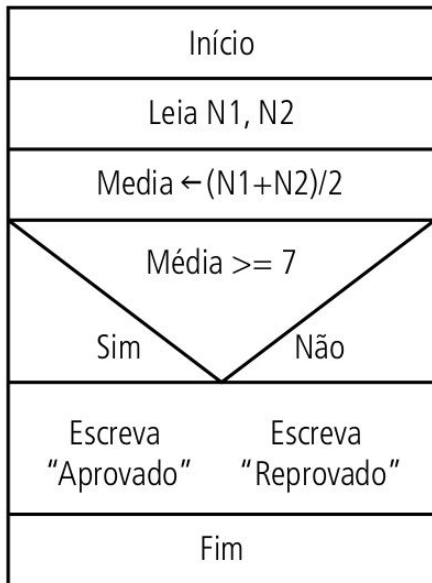
# DIAGRAMA DE CHAPIN

# Diagrama Chapin

O diagrama de Chapin foi criado por Ned Chapin a partir de trabalhos de Nassi-Shneiderman, os quais resolveram substituir o fluxograma tradicional por um diagrama que apresenta uma visão hierárquica e estruturada da lógica do programa.

A grande vantagem de usar este tipo de diagrama é a representação das estruturas que tem um ponto de entrada e um ponto de saída e são compostas pelas estruturas básicas de controle de sequência, seleção e repetição.

# Diagrama Chapin



**Figura 2.7: Diagrama de Chapin para o cálculo da média das notas de um aluno**

Fonte: Elaborado pelo autor

# PSEUDOCÓDIGO



# Pseudocódigo

Esta forma de representação de algoritmos, também conhecida como português estruturado ou portugol, é bastante rica em detalhes e, por assemelhar-se bastante à forma em que os programas são escritos, tem muita aceitação, sendo portanto a forma de representação de algoritmos mais utilizada.

O pseudocódigo é um código de escrita em que se utilizam termos convencionais para indicar as instruções do programa. Esses termos são geralmente uma mistura de palavras da nossa linguagem natural com palavras e notações típicas das linguagens de programação.

# Pseudocódigo

Esta forma de representação de algoritmos, também conhecida como português estruturado ou portugol, é bastante rica em detalhes e, por assemelhar-se bastante à forma em que os programas são escritos, tem muita aceitação, sendo portanto a forma de representação de algoritmos mais utilizada.

O pseudocódigo é um código de escrita em que se utilizam termos convencionais para indicar as instruções do programa. Esses termos são geralmente uma mistura de palavras da nossa linguagem natural com palavras e notações típicas das linguagens de programação.

# Pseudocódigo

A utilização de pseudocódigo permite ao programador expressar as suas ideias sem ter de se preocupar com a sintaxe da linguagem de programação.

# Pseudocódigo

**ALGORITMO** <nome> - primitiva usada para nomear o algoritmo representado;

**INICIO** (do inglês **START**)- esta primitiva é usada para identificar o ponto inicial do algoritmo;

**FIM** (do inglês **END**) - identifica o ponto final do algoritmo;

**LEIA** <mensagem>, <variavel> (do inglês **INPUT**) - primitiva usada para a leitura de dados do utilizador. Equivalente à entrada de dados manual do fluxograma;

**ESCREVA** <mensagem>, <expressão> (do inglês **OUTPUT**) – primitiva usada para a apresentação de dados ao utilizador. Equivalente ao símbolo de exibição do fluxograma;

<variavel> +- <expressão> - atribuição do resultado da expressão à variável indicada;

**SE** <condição> **ENTAO** (do inglês **IF ... THEN**) <instruções a executar se condição verdadeira>

**SENAO** (do inglês **ELSE**) <instruções a executar se condição falsa>

# Pseudocódigo

**FIM SE** (do inglês **END IF**) - primitiva de controlo de fluxo equivalente à decisão dos fluxogramas;

**ESCOLHA (<variavel>** (do inglês **SWITCH**)

CASO <valor 1>: <instruções a executar>

CASO <valor2>: <instruções a executar>

CASO <valorN>: <instruções a executar>

CASO CONTRARIO: <instruções a executar>

# Pseudocódigo

**FIM ESCOLHA** - primitiva de controlo de fluxo, usado para representar uma estrutura de controle de seleção múltipla;

**ENQUANTO <condição> FAÇA** (do inglês **WHILE**) <instruções a executar enquanto a condição for verdadeira>

**FIM ENQUANTO** (do inglês **END WHILE**) - primitiva de controlo de fluxo, sem equivalência direta em fluxogramas, que implementa um ciclo executado enquanto a condição referida seja verdadeira;

**REPITA** (do inglês **REPEAT**) <instruções a executar enquanto a condição for verdadeira>

**ATE <condição>** (do inglês **UNTIL**) - primitiva de controle de fluxo, sem equivalência direta em fluxogramas, que implementa um ciclo executado até que a condição referida seja verdadeira;

**PARA <condição inicial> ATÉ <condição final> FAÇA [PASSO <incremento>]** (do inglês **FOR ... TO ... [STEP ... ] DO**)

# Pseudocódigo

**<instruções a executar enquanto a condição final for falsa>**

**FIM PARA** (do inglês **END FOR**) - primitiva de controlo de fluxo, que executa as instruções nela contidas enquanto a condição final for falsa. O incremento pode ser omitido desde que seja unitário positivo;

**VAR <nome da variavel> [,]: <tipo da variavel>** (neste caso é usado uma redução do termo VARIÁVEL - do inglês **VARIABLE**) – primitiva utilizada na definição de variáveis. Os tipos de variáveis são discutidos em uma aula à frente;

**CONST <nome da constante> = <valor da constante> [,]** (neste caso é usado uma redução do termo CONSTANTE - do inglês **CONSTAND** - primitiva utilizada na definição de constantes. As constantes são discutidas à frente;

**ESTRUTURA <nome da variavel>: <tipo da variavel> [,]** (do inglês **STRUCT**)

[<nome da variavel>: <tipo da variavel>]

**FIM ESTRUTURA** (do inglês **END STRUCT**) - primitiva utilizada na definição de estruturas de variáveis. Os tipos de variáveis são definidos à frente;

# Pseudocódigo

Um exemplo de pseudocódigo é mostrado a seguir:

ALGORITMO Media

VAR N1, N2, N3, Media: real

INICIO

LEIA N1, N2, N3

Media  $\leftarrow (N1 + N2 + N3) / 3$

SE Media  $\geq 7$  ENTAO

    ESCREVA "Aprovado"

SENAO

    ESCREVA "Reprovado"

FIM SE

FIM.



# Pseudocódigo

Uma grande vantagem da utilização de pseudocódigo é o fato de permitir ao programador expressar as suas ideias sem ter de se preocupar com a sintaxe da Linguagem de Programação. Por outro lado, esse tipo de representação é o que mais se aproxima da codificação final em uma Linguagem de Programação e, por isso, é a mais utilizada na prática.

# Referências

ASCENCIO, Ana Fernanda Gomes; CAMPOS, Edilene Aparecida Veneruchi.

Fundamentos da Programação de Computadores. São Paulo. Pearson Pretience Hall, 2007.

MEDINA, Marco; FERTIG, Cristina. Algoritmos e Programação: Teoria e Prática. São Paulo. Novatec Editora, 2006.

MANZANO, José Augusto N. G.; OLIVEIRA, Jayr Figueiredo de. Algoritmos: Lógica Para Desenvolvimento de Programação de Computadores. São Paulo. Érica, 2000.

LOPES, Anita; GARCIA, Guto. Introdução à Programação. Rio de Janeiro. Elsevier, 2002.

SAMPAIO, Silvio Costa. Lógica para Programação. Palmas. Editora Educon, 2008.