

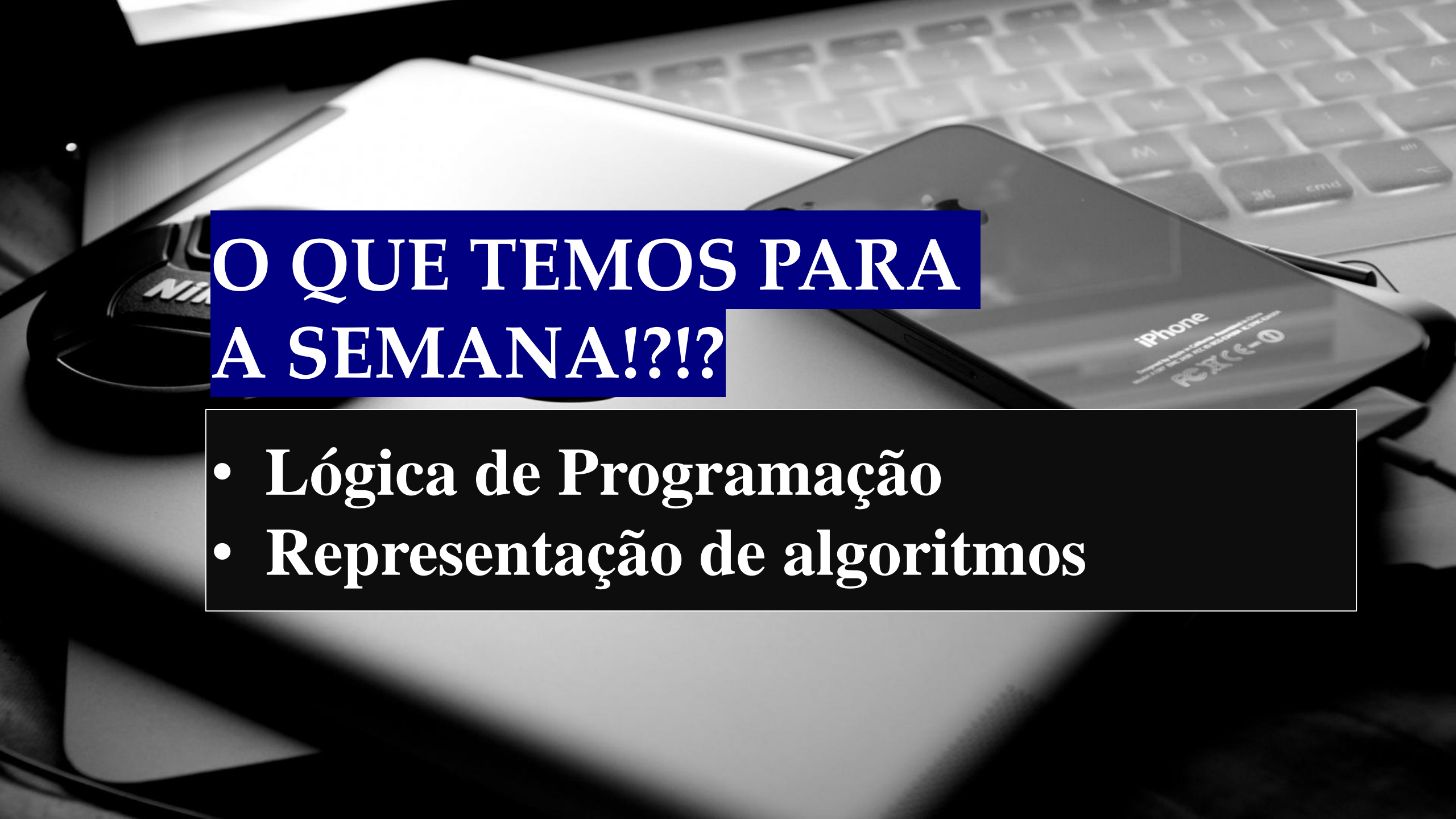
An abstract digital graphic featuring concentric circles and circuit-like lines in shades of blue and white against a dark background. The design is futuristic and tech-oriented, with glowing points and binary code visible in the background.

ESTRUTURA DE DADOS

Prof. Dr. Jean Nunes

Objetivos da Disciplina

- Fornecer uma base sólida sobre tipos de dados e as operações associadas aos tipos.
- Definir os objetos que constituem o dado e as operações aplicáveis fazendo uso das estruturas de dados estáticas básicas: lista, pilhas, filas e árvores.
- Demonstrar os principais algoritmos para construção, consulta e manipulação de estrutura de dados que envolvam ordenação e pesquisa.
- Orientar sobre quando e como aplicar corretamente as estruturas de dados para resolver problemas computacionais.



O QUE TEMOS PARA A SEMANA!?!?

- **Lógica de Programação**
- **Representação de algoritmos**

Introdução à Lógica de Programação

Quais **estratégias** você usa para solucionar os seus problemas?



**DEPENDE DO
PROBLEMA**

Solução lógica





A **lógica** é um dos **passos iniciais** para a construção de um programa.





O QUE É
LÓGICA?

- *“Estudo dos processos válidos e gerais pelos quais atingimos a verdade”.*
- *“Ciência das leis do pensamento”.*
- *“Parte da filosofia que estuda as leis do raciocínio.”*

Todo homem é mortal.
Sócrates é homem,
logo, Sócrates é mortal

João, ministro da educação, é acusado de corrupção e defende-se dizendo: ‘Esta acusação é um disparate’.



Vamos fazer um rápido exercício!

1. Falácia do espantalho
2. Você também (*tu quoque*)
3. Incredulidade pessoal
4. Ônus da prova
5. Apelo à popularidade
6. Apelo à autoridade
7. Escocês de verdade
8. Falsa dicotomia
9. Falácia anedótica
10. O atirador de elite do texas
11. Meio termo
12. Falsa causalidade
13. Apelo à emoção



Algoritmo: Conjunto de passos que definem como uma tarefa é realizada.

Tomar Banho



Fazer um Bolo



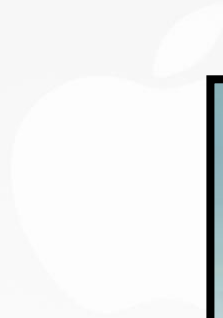
Calibrar um Pneu

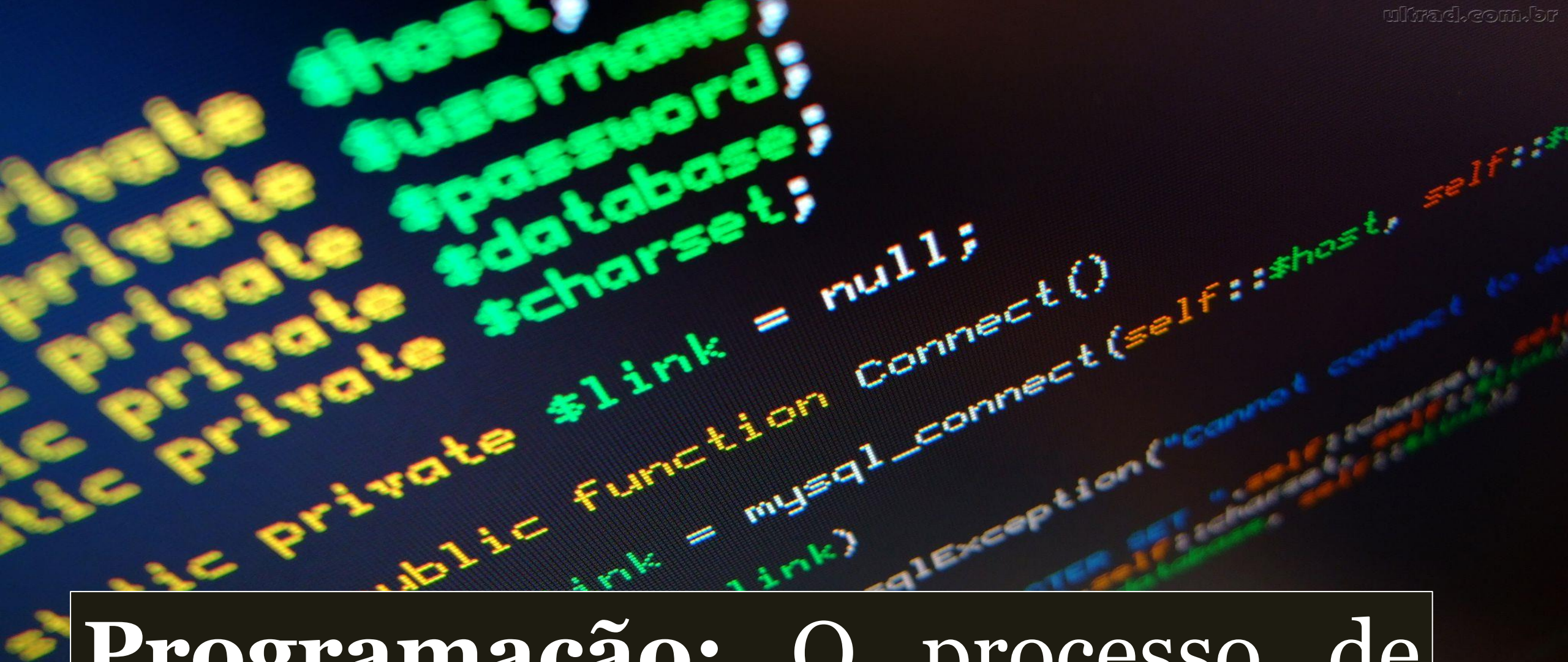
O algoritmo não é a solução do problema, mas uma forma de solucioná-lo.

Assim, podemos criar diferentes algoritmos usando diferentes abordagens.



Programa: Uma representação de um algoritmo. Interação.





Programação: O processo de desenvolver um programa

O estudo dos algoritmos foi originalmente um assunto da matemática.

- **Euclidean Algorithm**

$$M = 48 / N = 30$$

$$48 / 30 = 1 \text{ (R = 18)}$$

$$M = 30 / N = 18$$

$$30 / 18 = 1 \text{ (R = 12)}$$

$$M = 18 / N = 12$$

$$18 / 12 = 1 \text{ (R = 6)}$$

$$M = 12 / N = 6$$

$$12 / 6 = 0 \text{ (R = 0)}$$

$$\text{Então } \text{MDC}(48, 30) = 6$$

Método criado por Euclides por volta de 300 a.C.

Description: This algorithm assumes that its input consists of two positive integers and proceeds to compute the greatest common divisor of these two values.

Procedure:

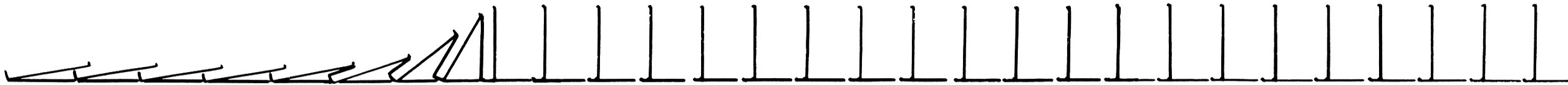
Step 1. Assign M and N the value of the larger and smaller of the two input values, respectively.

Step 2. Divide M by N, and call the remainder R.

Step 3. If R is not 0, then assign M the value of N, assign N the value of R, and return to step 2; otherwise, the greatest common divisor is the value currently assigned to N.

Formas de representação de algoritmos:

- Descrição narrativa
- Fluxograma convencional



DESCRIÇÃO NARRATIVA

Os algoritmos são expressos diretamente em linguagem natural – português/inglês!


- Usar somente um verbo por frase
- Imaginar que você está desenvolvendo um algoritmo para quem nunca desenvolveu
- Usar frases curtas e simples
- Ser objetivo
- Usar palavras que não tenham sentido dúbio

DESCRIÇÃO NARRATIVA

1. Início de programa
2. Leia os valores de A e B
3. Se A e B forem maiores que zero
 - a) Calcula o valor da média aritmética simples a partir da fórmula [$\text{média} = (A+B)/2$]
 - b) Imprime o valor da média
4. Senão imprime "os valores devem ser positivos"
5. Fim do programa

DESCRIÇÃO NARRATIVA

1. Faça um algoritmo narrativo que peça ao usuário para digitar três valores positivos e imprima a soma deles.



Caneta na mão!

Introdução à Lógica de Programação

FLUXOGRAMA

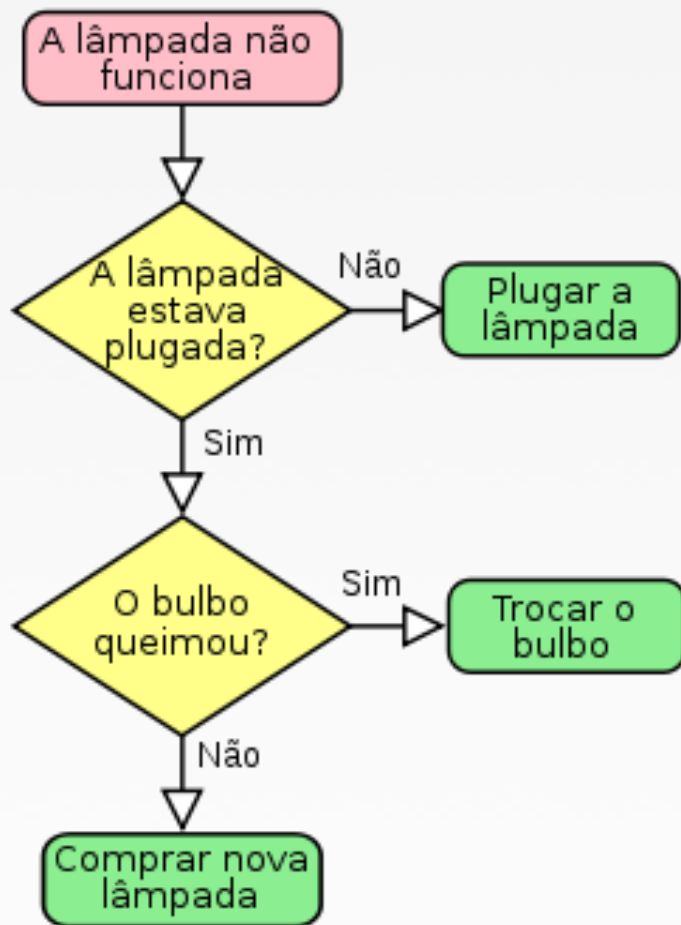
2. Faça uma descrição narrativa que descreva o cálculo da média final de um aluno que realiza três provas no semestre. É considerado “aprovado” se a média for igual ou superior a 7,00, senão “reprovado”.

Caneta na mão!



DESCRIÇÃO NARRATIVA

1. Início de programa
2. Leia os valores de A, B e C
3. Se A, B e C forem maiores que zero
 - a) Calcula o valor da soma a partir da fórmula [$soma = (A + B + C)$]
 - b) Imprime o valor da soma
4. Senão imprime "os valores devem ser positivos"
5. Fim do programa



FLUXOGRAMA

Representação gráfica que emprega formas geométricas padronizadas para indicar as diversas ações e decisões que devem ser executadas (ISO 9000).



Introdução à Lógica de Programação

FLUXOGRAMA

Terminal



- Utilizado para representar o início e/ou fim do fluxo lógico de um programa.
- De forma mais genérica, representa o início e o fim da resolução de um problema.



Linha Básica

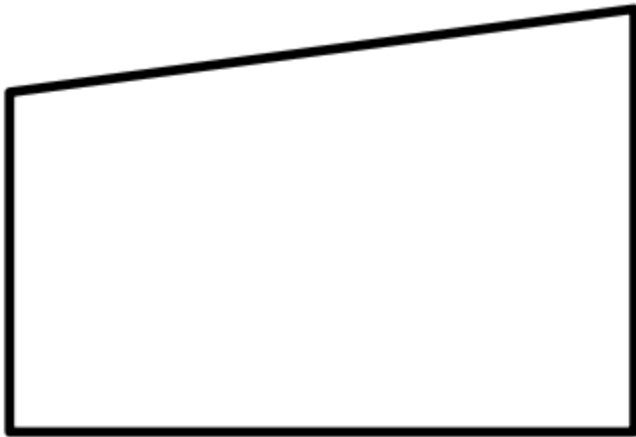
- Utilizada com setas nas extremidades para representar direção do fluxo, facilitando sua leitura,
- De uma maneira mais genérica, representa a passagem de dados ou controles.



Introdução à Lógica de Programação

FLUXOGRAMA

Entrada Manual



- Representa dados, de qualquer tipo de mídia que podem ser alimentados manualmente (teclado, mouse, etc.)



Introdução à Lógica de Programação

FLUXOGRAMA

Dados



- Representa dados de uma forma genérica, sendo em entrada ou saída dos mesmos uma vez que foram identificados.



Introdução à Lógica de Programação

FLUXOGRAMA

Processo



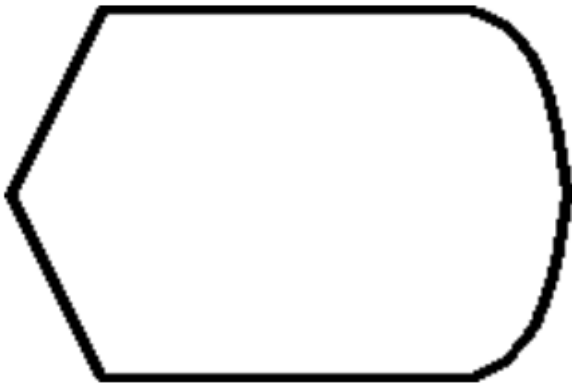
- Representa de forma genérica quaisquer tipos de processos, designando um trabalho específico ou um conjunto de operações que tem como objetivo a mudança de valor, formato ou localização de uma informação.



Introdução à Lógica de Programação

FLUXOGRAMA

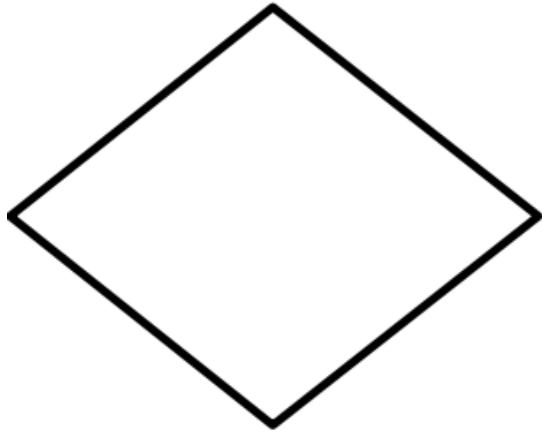
Exibição



- Usado em dados cuja mídia seja de qualquer tipo onde a informação seja mostrada para uso humano, tais como monitores de vídeo, indicadores, telas, etc.



Desvio condicional



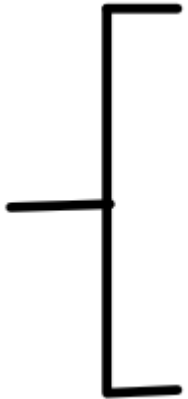
- Representar uma decisão ou um desvio
- Pode ter uma série de saídas alternativas, uma única das quais deverá ser ativada como consequência da avaliação das condições internas ao símbolo.



Introdução à Lógica de Programação

FLUXOGRAMA

Anotação



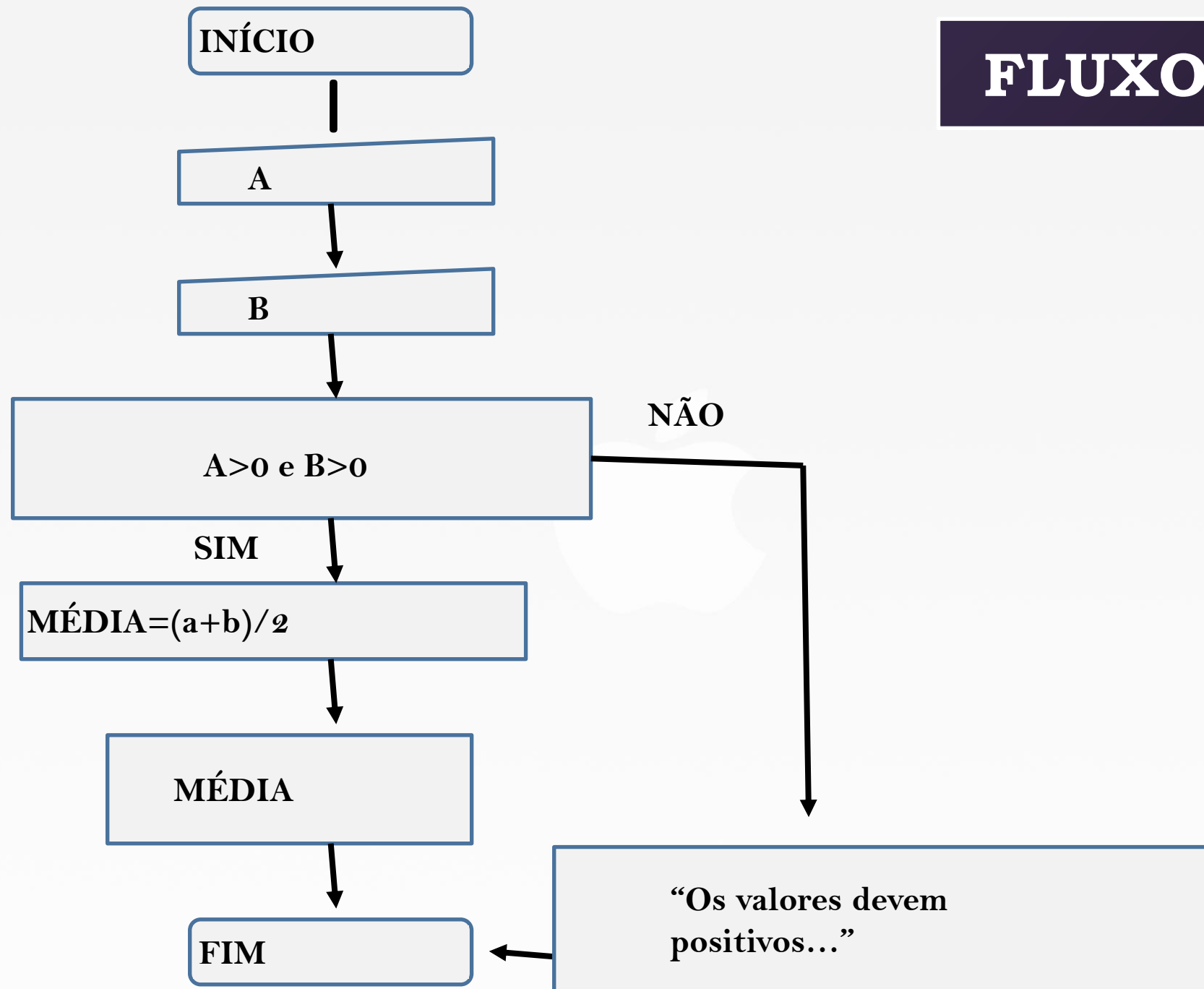
- Representa adição de comentários para esclarecimento ou explanação de observações.



DESCRIÇÃO NARRATIVA

1. Início de programa
2. **Leia** os valores de **A** e **B**
3. **Se** A e B forem maiores que zero
 - a) Calcula o valor da média aritmética simples a partir da fórmula [$média = (A+B)/2$]
 - b) **Imprime** o valor da média
4. **Senão imprime** "os valores devem ser positivos"
5. Fim do programa

FLUXOGRAMA



Introdução à Lógica de Programação

FLUXOGRAMA

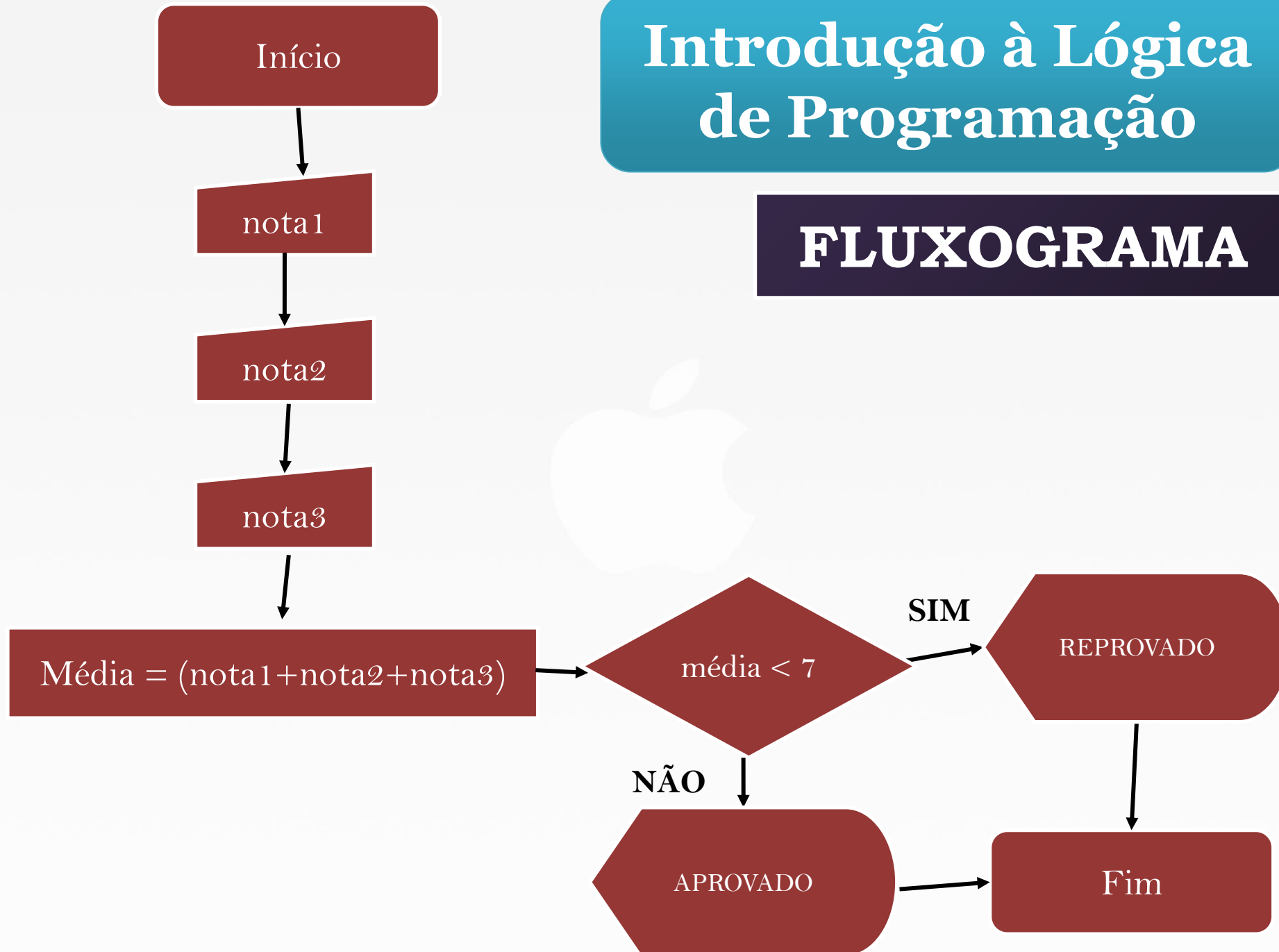
2. Faça um fluxograma que descreva o cálculo da média final de um aluno que realiza três provas no semestre. É considerado “aprovado” se a média for igual ou superior a 7,00, senão “reprovado”.

Caneta na mão!



Introdução à Lógica de Programação

FLUXOGRAMA



Introdução à Lógica de Programação

FLUXOGRAMA

3. Faça um fluxograma que descreva a leitura de três valores inteiros positivos e imprima a soma deles.

Caneta na mão!



Introdução à Lógica de Programação

FLUXOGRAMA

4. Faça um fluxograma que descreva a nota de um aluno a ser calculada a partir da seguinte formula expressa pela Equação abaixo:

$$AV_1 = \left(\left(\frac{AP_{1^{a}mn} + AP_{2^{a}mn} + AP_{3^{a}mn} + AP_{4^{a}mn}}{4} \right) * 0,4 \right) + (NP * 0,6))$$

AP – Atividade Prática

Subscrito “1^amn”: “primeira maior nota”. Os demais subscritos seguem a ordem representada pelo número.

Programa: Uma representação de um algoritmo. Interação.



colab

programiz

eXtreme Programming (XP)

Boas práticas do XP:



1. Programação em pares: programadores

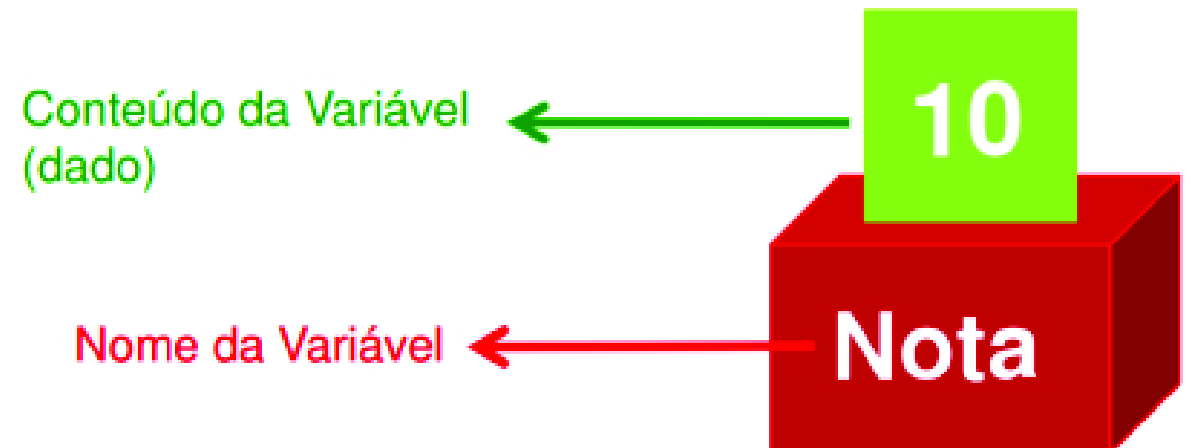
- programador codifica enquanto o outro faz sugestões pertinentes
- papéis podem ser trocados com alguma frequência para otimizar o processo
- a programação ganha em qualidade, pois ambos acompanham o desenvolvimento do código
- um segundo olhar está sempre mais apto a identificar falhas

Estrutura Condicional e Operadores

Programação

Tipos de Variáveis

Tipo de Dados	Representação
Caracter	char
Inteiro	inteiro
Real	float
Cadeia de caracteres	string



Programação

Escrita e Leitura

Função	Descrição
print/printf	permite apresentar na tela os valores de qualquer tipo de dado, ou imprimir uma sequência de caracteres.
input/scanf	Usada para efetuar a leitura de valores do teclado.

Programação

Estrutura de Condição 1 (if - else)

```
// 5 Alternatives to this
if (condition) {
    // do stuff
} else if (otherCondition) {
    // do something else
} else {
    //Default
}
```

A variável, na condição, será avaliada. Se ela for maior ou igual a 0 (zero), as instruções de soma e impressão serão executadas. Senão, será executado uma instrução de impressão.

Programação

Estrutura de Condição 1 (if - else) - Operadores

I – Operador de igualdade (=)

Comum à maioria das linguagens, este operador (=) faz a variável da esquerda assumir o valor da variável, constante ou expressão da direita. Exemplo:

```
var = var + 2;
```

Programação

Estrutura de Condição 1 (if - else) - Operadores

II – Operadores aritméticos

Operador	Descrição	Exemplo
*	Multiplicação	<code>var = var * 2;</code>
/	Divisão	<code>var = var / 2;</code>
%	Resto	<code>var = var % 2;</code>
+	Adição	<code>var = var + 2;</code>
-	Subtração	<code>var = var - 2;</code>

Programação

Estrutura de Condição 1 (if - else) - Operadores

III – Operadores relacionais

Eles avaliam o relacionamento entre duas expressões e dão o resultado 1 se verdadeiro ou 0 se falso.

Operador	Descrição	Exemplo
<	Menor que	<code>if (var < 10)</code>
<=	Menor ou igual	<code>if (var <= 10)</code>
>	Maior que	<code>if (var > 0)</code>
>=	Maior ou igual	<code>if (var >= 0)</code>
==	Igual	<code>if (var == 0)</code>
!=	Diferente	<code>if (var != 0)</code>

Programação

Estrutura de Condição 1 (if - else) - Operadores

IV – Operadores lógicos

Usados normalmente com expressões booleanas, isto é, expressões que retornam verdadeiro ou falso (1 ou 0), para fins de testes em declarações condicionais.

Operador		Descrição	Exemplo
&&	and	E lógico	<pre>if (var < 10 && var > 0) if (var < 10 and var > 0)</pre>
	or	OU lógico	<pre>if (var1 > 10 var2 > 0) if (var1 > 10 or var2 > 0)</pre>

Programação

VAMOS PROGRAMAR!!!

Faça um programa que leia a distância em ***Km*** e a quantidade de litros de gasolina consumidos por um carro em um percurso. Calcule o consumo em ***Km/l*** ($\text{distancia} / \text{qtd_consumo_gasolina}$) e escreva uma mensagem de acordo com a tabela abaixo:

Consumo	(Km/l)	Mensagem
Menor que	7	Venda esse carro!!!
Entre	7 e 10	Existem opções de carro mais econômicos. Considere a troca do veículo!!!
Entre	10 e 13	Carro econômico!!!
Maior que	13	Super econômico!!!



1. Tipos de variáveis (char, integer, float, string)
2. Operadores aritméticos (multiplicação, divisão, resto, adição, subtração)
3. Operadores relacionais (menor que, menor ou igual, maior que, maior ou igual, igual, diferente)
4. Operadores lógicos
5. Estrutura de condição

