



IFPB – Campus Campina Grande
Curso Técnico Integrado em Informática



Algoritmos e Lógica de Programação

Teoria e Prática

Aula 3



Professor:
Marcos Vinicius C. M. de Andrade
(marcos.andrade@academico.ifpb.edu.br)
marcosvcma@ifpb.edu.br

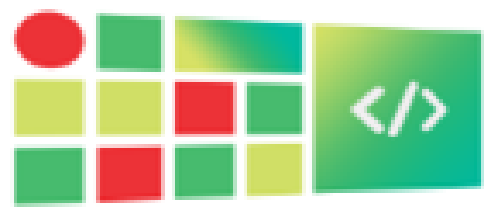




Conteúdo da aula

- ✓ Algoritmos
- ✓ Formas de Representação
- ✓ Escrevendo Algoritmos
- ✓ Algoritmos X Linguagem de Programação





Programação

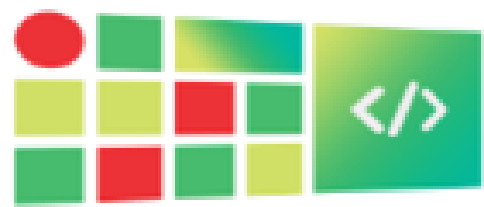
Algoritmos



Lembrando a definição:

“É uma sequência de passos (instruções, comandos, ordens, regras), finita e não ambígua, para solucionar um problema”.





Programação

Algoritmos



Sabendo o que é Algoritmos, a pergunta que fica é: **Como “fazer” algoritmos?**

Descrever esses passos finitos não é fácil. É preciso seguir regras e usar algumas definições. Além disso, precisamos escrever para “alguém que não entende o que irá fazer”, no nosso caso, para o computador. As vezes é possível que o “executor” não entenda como nós entendemos.

É preciso também ter em mente que esses passos precisam ser representados de alguma forma. Existem as formas de representação dos algoritmos

Formas:

- ❖ Descrição narrativa
- ❖ Fluxograma
- ❖ Linguagem Algorítmica



Tipos de Algoritmos



Descrição narrativa: São expressos diretamente em linguagem natural.

Exemplo: Receita de bolo.

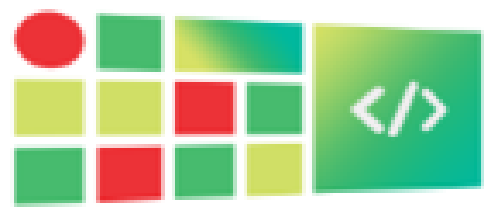
Fluxograma: São expressos diretamente utilizando representação gráfica (formas geométricas).

Exemplo: Fluxograma para uma empresa efetuar o pagamento.

Linguagem algorítmica: São expressos diretamente em uma linguagem artificial.

Exemplo: Calcular a média de duas notas.





Programação

Tipos de Algoritmos



► Exemplos: Descrição narrativa:

Separe os ingredientes da massa (xícara usada de 240ml):

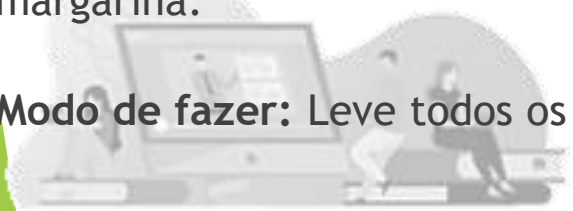
2 ovos; 1 xícara (chá) de leite; 1 xícara (chá) de óleo; 3 colheres (sopa) de chocolate em pó; 1 e ½ xícara (chá) de farinha de trigo; ½ xícara (chá) de Amido de Milho (Maizena); 1 e ½ xícara (chá) de açúcar; 1 colher (sopa) de fermento em pó.

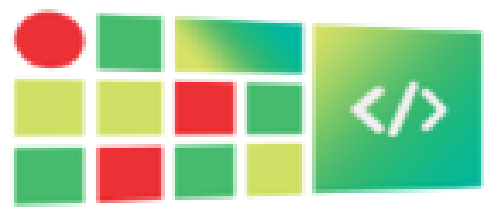
Modo de fazer: Adicione os líquidos e misture e reserve. Em outro recipiente junte os ingredientes secos (menos o fermento) e depois acrescente os líquidos reservados. Por fim acrescente o fermento e mexa só para misturar. Coloque em forma untada e enfarinha (ou polvilhada com chocolate) e leve ao forno que já estava preaquecido em 180°C por aproximadamente 45 min. Deixe amornar e desenforme.

Separe os ingredientes da Cobertura:

1/2 lata de leite condensado; 120 ml de leite; 2 colheres (sopa) de chocolate em pó; 1 colher (sopa) de manteiga ou margarina.

Modo de fazer: Leve todos os ingredientes ao fogo médio/baixo. Quando ferver retire. Despeje sobre o bolo.



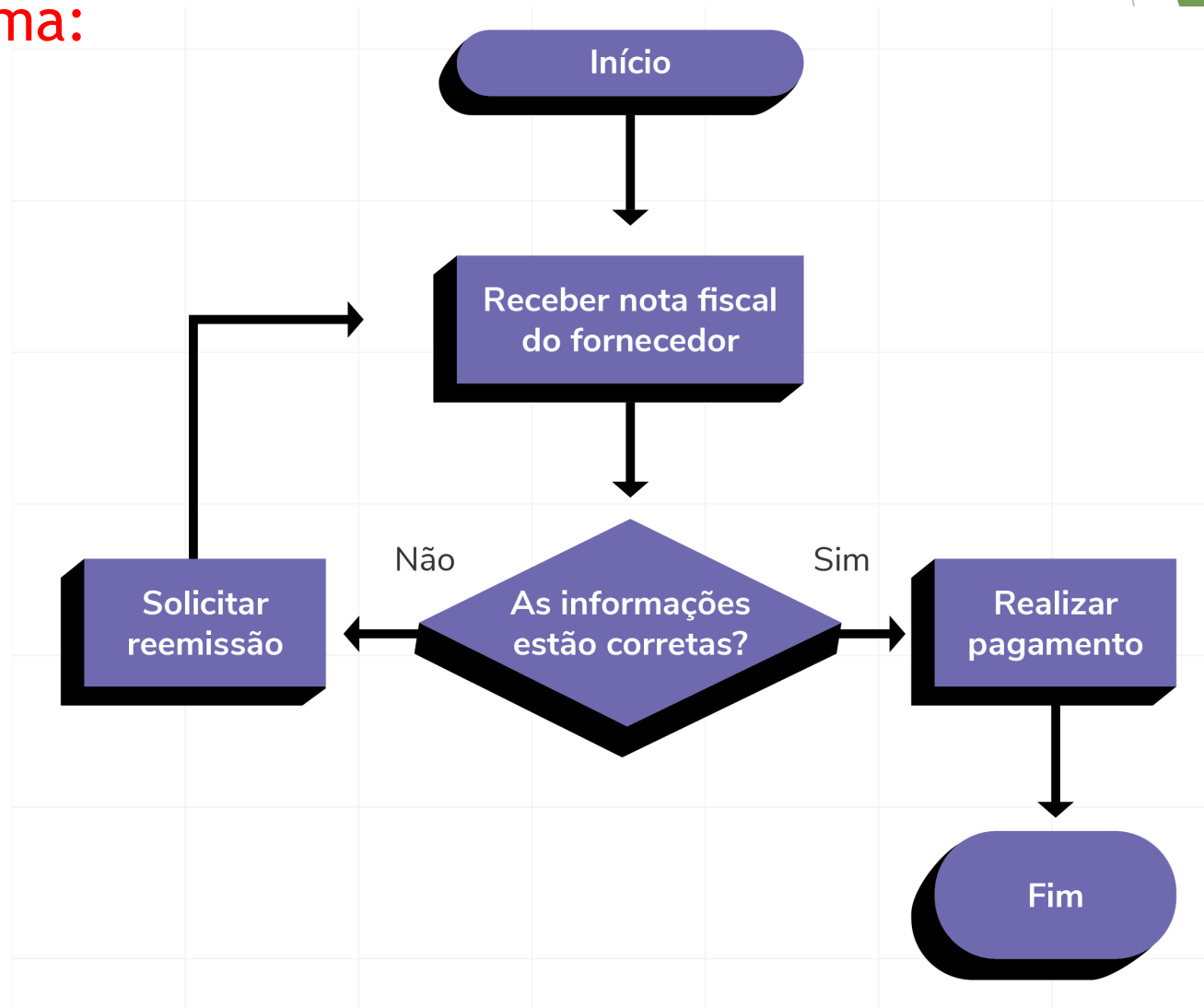


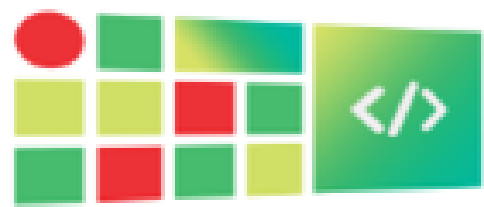
Programação

Tipos de Algoritmos



► Exemplos: Fluxograma:





Programação

Tipos de Algoritmos



► Exemplos: Linguagem Algorítmica:

Algoritmo Média

Var N1, N2, Média

Início

Leia N1, N2

Média := $(N1 + N2) / 2$

Se Média ≥ 7 Então

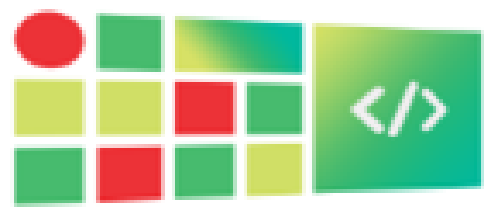
Escreva "Aprovado"

Senão

Escreva "Reprovado"

Fim.





Programação

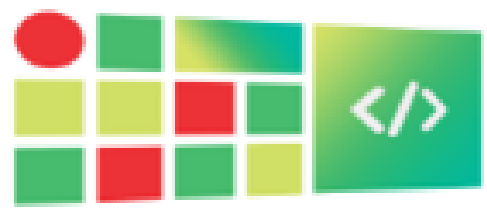
Tipos de Algoritmos



► Qual forma usar?

	Descrição Narrativa	Fluxograma	Linguagem Algorítmica
Vantagens	<ul style="list-style-type: none">• O português é bastante conhecido por nós	<ul style="list-style-type: none">• Uma das ferramentas mais conhecidas;• Figuras dizem muito mais que palavras;• Padrão mundial	<ul style="list-style-type: none">• Usa o português como base;• Pode-se definir quais e como os dados vão estar estruturados;• Passagem quase imediata do algoritmo para uma linguagem de programação qualquer.
Desvantagens	<ul style="list-style-type: none">• Imprecisão;• Pouca confiabilidade;• Extensão.	<ul style="list-style-type: none">• Pouca atenção aos dados, não oferecendo recursos para descrevê-los ou representá-los;• Complica-se à medida que o algoritmo cresce .	<ul style="list-style-type: none">• Exige a definição de uma linguagem não real para trabalho;• Não padronizado.

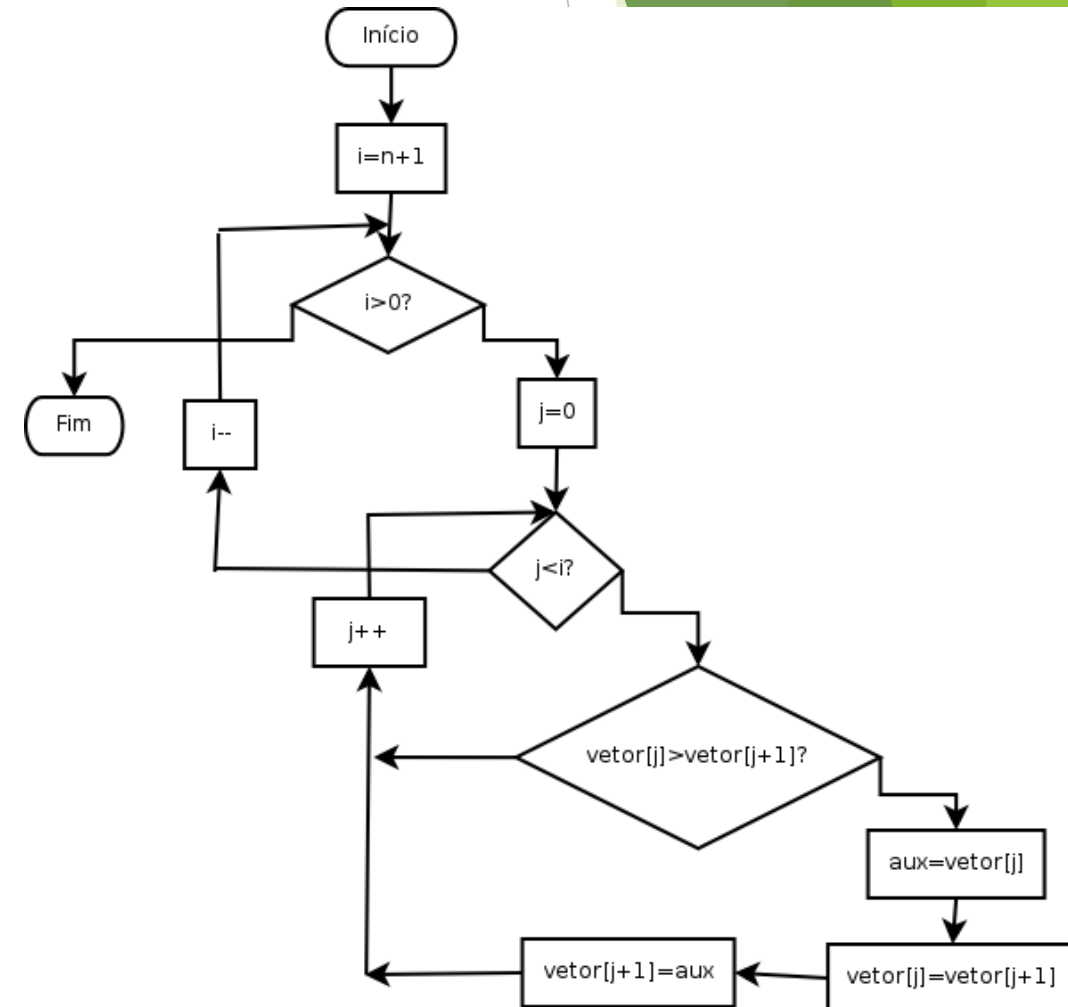
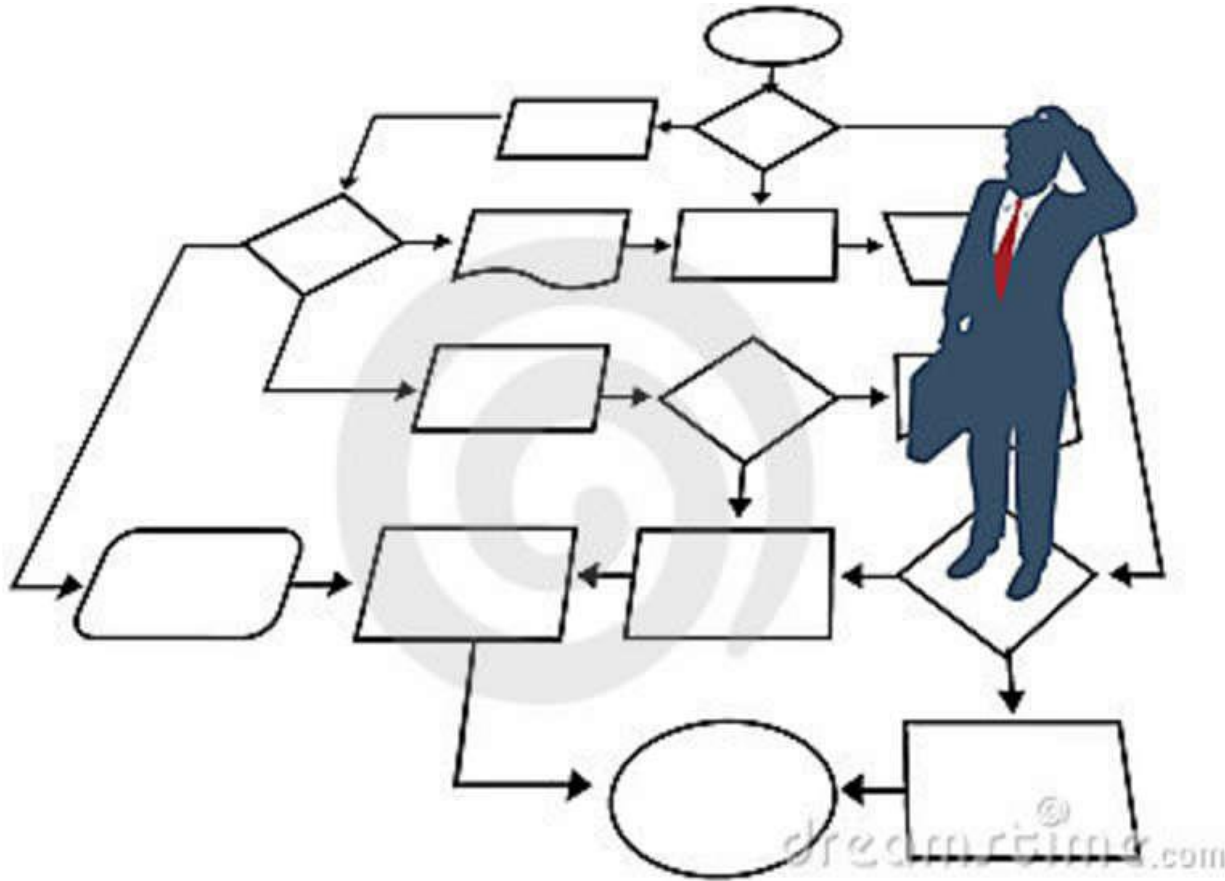




Tipos de Algoritmos



► **Exemplos:** Fluxograma quando o algoritmo é maior:





Escrevendo Algoritmos



Voltando um pouco para “como escrever um algoritmo”, vejamos algumas regras:

Usar um verbo por frase.

Usar frase simples e direta.

Ser efetivo, ou seja, resolver o problema.

Poder receber dados/informações do mundo exterior ao algoritmo.

Poder enviar dados/informações ao mundo exterior ao algoritmo.

Não dar margem a dupla interpretação.

Ter a possibilidade de ser refinado (melhorado, aperfeiçoado...).

Partindo dessas regras, vamos tentar escrever um algoritmo para o seguinte problema:



Escrevendo Algoritmos



Faça um algoritmo para trocar uma lâmpada.

Passo 1. Remova a lâmpada queimada;

Passo 2. Coloque a lâmpada nova.

A princípio, temos um algoritmo. Ele é **simples**, **claro**, **sem ambiguidade**, está escrito **na ordem** correta (os passos invertidos não resolvem o problema), é **efetivo**, **finito**, e **resolve o problema**.

Será?

Dependendo quem for trocar a lâmpada pode ser que não entenda como executa um ou os dois passos. É necessário então “**detalharmos**” um pouco mais.





Escrevendo Algoritmos



Vamos refinar o mesmo algoritmo:

Passo 1. Remova a lâmpada queimada;

Passo 1.1 Posicione a escada debaixo da lâmpada;

Passo 1.2 Suba na escada até que a lâmpada possa ser alcançada;

Passo 1.3 Gire a lâmpada no sentido anti-horário até que solte.

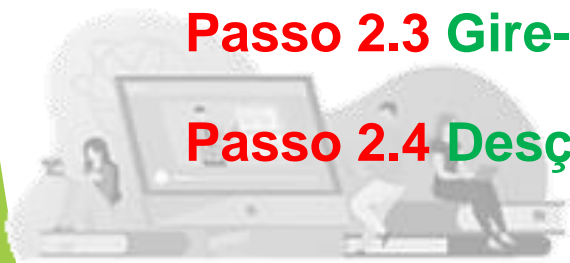
Passo 2. Coloque a lâmpada nova;

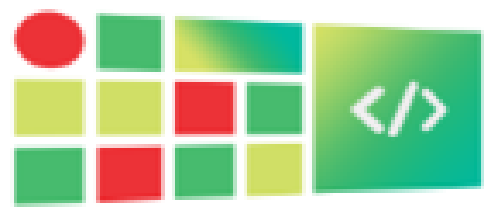
Passo 2.1 Escolha uma lâmpada nova da mesma potência que a queimada;

Passo 2.2 Posicione a nova lâmpada no soquete;

Passo 2.3 Gire-a no sentido horário até que se firme;

Passo 2.4 Desça da escada.





Programação

Escrevendo Algoritmos



É possível refinar mais? Sim, sempre é possível! Veja por exemplo como melhorar o passo 2.1

...

Passo 2. Coloque a lâmpada nova;

Passo 2.1 Escolha uma lâmpada nova da mesma potência que a queimada;

Passo 2.1.1 Selecione uma nova lâmpada na caixa de lâmpadas

Passo 2.1.2 Verifique a potência

Passo 2.1.3 Se for igual a da lâmpada queimada então pare o processo de busca na caixa

Passo 2.1.4 Senão, repita a operação a partir do passo 2.1.1

Passo 2.2 Posicione a nova lâmpada no soquete;

...





Escrevendo Algoritmos



Como saber se o algoritmo está detalhado o suficiente?

A resposta é: **depende de quem vai executá-lo.**

Um algoritmo é considerado **completo** se os seus comandos forem do entendimento do seu destinatário.

Não devemos definir os passos de forma **muito genérica** (ou abstrata), nem tampouco **detalhar demasiadamente** os passos a ponto de tornar o algoritmo óbvio e grande demais.

E agora? Já escrevi a “lógica”, os passos que precisam serem executados. O que preciso fazer agora?

Dependendo do problema, se **cotidiano** ou **computacional**, é preciso então definir a forma de representar esse algoritmo para ser executado: **Narrativa, Fluxograma ou Linguagem algorítmica.**



Escrevendo Algoritmos



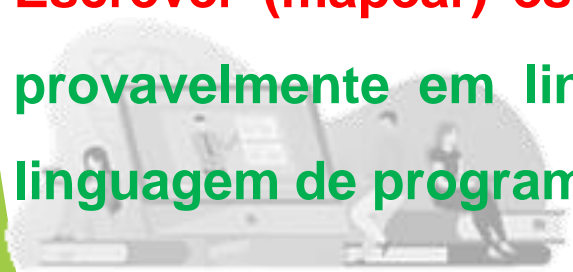
Nessa disciplina, iremos trabalhar com algoritmos computacionais, logo temos a seguinte sequencia:

- ✓ **Identificar e entender bem o problema que se está querendo resolver.**

Acredite, é um dos principais problemas encontrados pelos profissionais de informática quando se deseja construir um sistema de software para um cliente. Entender o problema pode requerer mais tempo do que a escrita em si.

- ✓ **Escrever uma solução algorítmica.** Quais passos precisamos executar para a solução do problema.

- ✓ **Escrever (mapear) este algoritmo em uma forma de representação.** Muito provavelmente em linguagem algorítmica para ser “traduzida” para uma linguagem de programação, ou em alguns casos, já mapear para a LP.





Escrevendo Algoritmos



Vejamos essa sequencia com o seguinte problema:

cálculo de média e exibir situação

Identificar e entender o problema.

O que se deseja? Como calculo a média? É aritmética ou ponderada? Que situação? Quais as possibilidades? Como encontrar a situação? Tem fórmulas?

Quais? Preciso receber algo para usar essa fórmula? ...

Escrever a solução, os passos para a solução, ou seja, Escrever o Algoritmo.

Passo 1: Receber o primeiro número

Passo 1.1: Armazenar o primeiro número em n1

Passo 2: Receber o segundo número

Passo 2.1: Armazenar o segundo número em n2.

Passo 3: Somar no n1 com o n2.

Passo 3.1: armazenar o resultado da soma em m.

Passo 4: Comparar m com um valor fixo 7.0.

Passo 5: Verificar o resultado da comparação de superioridade.

Passo 5.1: Caso o resultado seja m maior ou igual, escreva a situação APROVADO

Passo 5.2: Caso o passo 5.1 seja menor, escreva a situação REPROVADO.



Escrevendo Algoritmos



Escrever (mapear) este algoritmo em uma forma de representação.

Descrição Narrativa	Fluxograma Convencional	Pseudocódigo
<p>Para obter a média de cada aluno o professor deverá inicialmente identificar a primeira nota do aluno, localizada na nota1, em seguida o professor deve identificar a segunda nota, localizada na nota2. De posse dessas duas notas, o professor deverá efetuar o cálculo da média, ou seja, somar a nota1 com o valor da nota2, tendo esse resultado da soma sendo dividido pela quantidade de notas, no caso 2. Esse resultado será a média do aluno, que pode ser exibido para que o aluno veja.</p> <p>Para então exibir a situação do aluno, o professor deverá comparar a média obtida com uma média padrão que é 7.0, nessa comparação, sendo a média obtida pelo aluno superior ou igual a média padrão, então o professor deve exibir a situação APROVADO, caso contrário, essa média obtida pelo aluno não seja superior nem tão pouco seja igual a média padrão, então o professor deverá exibir a situação REPROVADO.</p>	<pre>graph TD; Inicio([Início]) --> N1N2[N1, N2]; N1N2 --> Calculo[Média := (N1 + N2) / 2]; Calculo --> Decisão{Média >= 7}; Decisão -- Sim --> Aprovado[/Aprovado/]; Decisão -- Não --> Reprovado[/Reprovado/]; Aprovado --> Fim([Fim]); Reprovado --> Fim;</pre>	<p>Algoritmo Média</p> <p>Var N1, N2, Média</p> <p>Início</p> <p> Leia N1, N2</p> <p> Média := (N1+N2)/2</p> <p> Se Média >= 7 Então</p> <p> Escreva “Aprovado”</p> <p> Senão</p> <p> Escreva “Reprovado”</p> <p> Fim.</p>



Algoritmos X Linguagem de Programação



Algoritmo = linguagem de programação?

Não!

Nunca confunda algoritmos com linguagem de programação.

Algoritmos: ideia geral que pode ser implementada em qualquer linguagem de programação. São os passos, a lógica, o que precisa ser feito.

Linguagem de programação: meio para enviar instruções para máquina (computador). A tradução quase que de imediato da representação do algoritmo na forma de linguagem algorítmica para uma linguagem que o computador entende.





Algoritmos X Linguagem de Programação



Vejamos um exemplo: Calcular a média entre 2 números.

Algoritmo de calcular a média de dois números

Programa para calcular a média de dois números na linguagem de programação Visualg

Passo 01: Receber o primeiro número

Passo 01.01: Armazenar o 1º número

Passo 02: Receber o segundo número

Passo 02.02: Armazenar o 2º número

Passo 03: Calcular o a média

Passo 03.01: Armazenar o resultado

Passo 04: Exibir o resultado

Algoritmo "calcule_a_media_de_2_numeros"

Var

n1,n2: inteiro

media: real

Inicio

escreval("Digite a nota 1: ")

leia(n1)

escreval("Digite a nota 2: ")

leia(n2)

media := (n1+n2)/2

escreva("A média entre ", n1, " e", n2, " é:")

escreval (media)

Fimalgoritmo

Mesmo programa em linguagem de Programação Python

```
num1,num2 = float(input("Digite a nota 1: ")),float(input("Digite a nota 2: "))
print("A média é entre {} e {} é: {}".format(num1,num2,(num1+num2)/2))
```