

Algoritmos e Programação Estruturada

Conceitos de algoritmos e programação

Você sabia que seu material didático é interativo e multimídia? Isso significa que você pode interagir com o conteúdo de diversas formas, a qualquer hora e lugar. Na versão impressa, porém, alguns conteúdos interativos ficam desabilitados. Por essa razão, fique atento: sempre que possível, opte pela versão digital. Bons estudos!

Nessa webaula, vamos ver o conceito de algoritmo, como funcionam e quais suas aplicações dentro da programação.

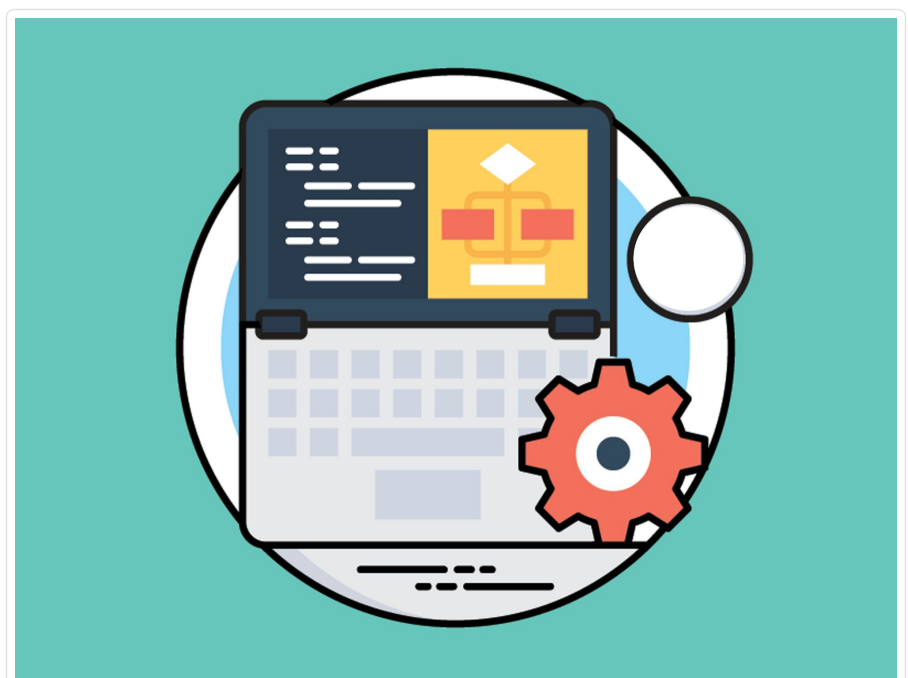
Algoritmo

Os algoritmos são as bases para criação de um programa de computador, onde diversas aplicações poderão ocorrer. Um algoritmo bem estruturado vai gerar um programa para solução de um problema que antes, parecia complexo. Todas as áreas estão voltadas para a tecnologia e são através de diversas formas de pensamentos que os algoritmos são realizados.

Algoritmo é uma sequência ordenada de passos que deve ser seguida para a realização de uma tarefa (BERG e FIGUEIRÓ, 1998).

Os algoritmos nortearão a descobrir qual o melhor percurso para solucionar um problema computacional.

A elaboração de algoritmos é um passo importante para o desenvolvimento de um programa de computador (ou *software*), pois, a partir da construção de algoritmos para a resolução de algum problema, é possível traduzir o algoritmo para alguma linguagem de programação.



Fonte: Shutterstock.

A seguir, veja alguns exemplos de algoritmo:

Algoritmo para efetuar o cozimento de um arroz

1. Acender o fogo;
2. Refogar os temperos;
3. Colocar o arroz na panela;
4. Colocar a água;
5. Cozinhar o arroz;
6. Abaixar o fogo;
7. Esperar o ponto;
8. Desligar o fogo;

9. Servir o arroz.

Algoritmo para efetuar o cozimento de um arroz (mais detalhado)▼

1. Comprar o arroz;

2. Analisar a qualidade;

3. Realizar a pré-seleção para o cozimento;

4. Preparar o tempero;

5. Pegar a panela;

6. Acender o fogo;

7. Colocar os temperos na panela para refogar;

8. Adicionar o arroz;

9. Colocar a água na medida considerada ideal par a quantidade;

10. Aguardar a água secar;

11. Baixar o fogo;

12. Fechar a panela com a tampa;

13. Aguardo o ponto;

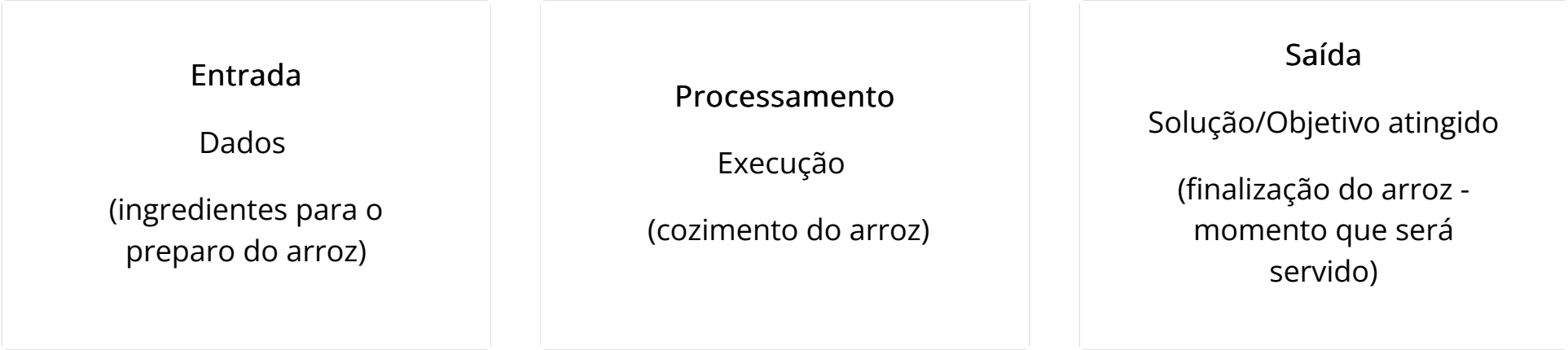
14. Desligar o fogo;

15. Servir o arroz.

Assim, percebemos que não existe somente uma forma de realizar um algoritmo, podem ser criada outras formas e sequências para obter o mesmo resultado, ou seja, eles são independentes, porém, com a mesma finalidade de execução.

Representação dos algoritmos

O algoritmo é representado em três partes:



Linguagem natural

A linguagem natural na definição geral é uma forma de comunicação entre as pessoas de diversas línguas, ela pode ser falada, escrita, gesticulada entre outras formas de comunicação. A linguagem natural tem uma grande contribuição quando vamos desenvolver uma aplicação computacional, pois ela pode direcionar de forma simples e eficiente as descrições dos problemas e suas soluções (SANTOS, 2001).

Para reforçar os conceitos de linguagem natural podemos ver como exemplo o cadastro de notas de alguns alunos de um curso.

Problema:

O usuário deverá entrar com dois valores (as notas) e o computador retorna o resultado da média destes valores (média das notas).

Perceba que a linguagem natural é muito próxima da nossa linguagem.

Solução

1. Início;

2. Entrar com o primeiro valor (nota do primeiro bimestre);

3. Entrar com o segundo valor (nota do segundo bimestre);

4. Realizar a soma do primeiro valor com o segundo;

5. Realizar a divisão do total dos valores por dois (média das notas dos bimestres);

6. Armazenar o valor encontrado;

7. Mostrar na tela o resultado da média;
8. Se a média do aluno for maior ou igual a seis;
9. O aluno será considerado aprovado;
10. Senão está reprovado;
11. Fim.

Outro exemplo é o algoritmo para calcular o máximo divisor comum, o famoso “MDC”:

1. Dividir um número “a” por “b”, onde o resto é representado por “r”
2. Substituir a por b
3. Substituir b por r
4. Continuar a divisão de a por b até que um não possa ser mais dividido, então “a” é considerado o mdc.

De acordo com a solução, o resultado fica:
MDC (480,130) = 10

Solução do “MDC” do algoritmo		
a	b	R
480	130	90
130	90	40
90	40	10
40	10	0
10	0	

Fonte: adaptada de Piva (2012).

Variáveis e Atribuições

As **variáveis** como o próprio nome sugere, é algo que pode sofrer variações, ou seja, estão relacionadas a identificação de uma informação. Exemplos: valor1, nome.

A **atribuição** (\leftarrow) tem a função de indicar valores para as variáveis, ou seja, atribuir informação para variável. Exemplos:

Valor \leftarrow 8

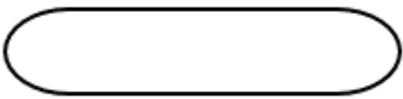
Valor1 \leftarrow marcio

Significa que o número “8” está sendo atribuído para variável “valor1” e que o texto “marcio” está atribuído para variável “nome”.

Diagrama de Blocos (Fluxograma)

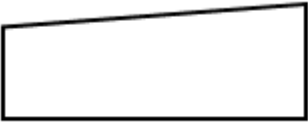
Diagrama de blocos é um conjunto de símbolos gráficos, onde cada um desses símbolos representa ações específicas a serem executadas pelo computador. Determina a linha de raciocínio utilizada pelo programador para resolver problemas. Os símbolos dos diagramas de bloco foram padronizados pela ANSI (Instituto Norte Americano de Padronização).

A seguir, veja a descrição dos principais símbolos utilizados em um diagrama de blocos, de acordo com Manzano (2015):




Terminal

Representa o início ou o fim de um fluxo lógico. Em alguns casos define as sub-rotinas.



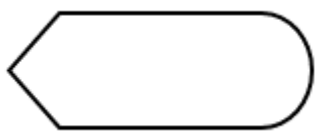
Entrada manual

Determina a entrada manual dos dados, geralmente através de um teclado.



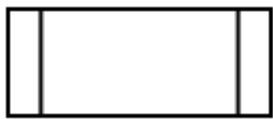
Processamento

Representa a execução de ações de processamento.



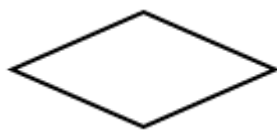
Exibição

Mostra o resultado de uma ação, geralmente através da tela de um computador.



Processo definido

Mostra o resultado de uma ação, geralmente através da tela de um computador.



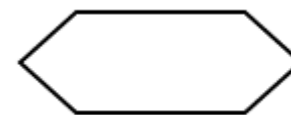
Decisão

Representa os desvios condicionais nas operações de tomada de decisão e laços condicionais para repetição de alguns trechos do programa.



Conector

Representa pontos de conexões entre trechos de programas, que podem ser apontados para outras partes do diagrama de bloco.



Preparação

Representa a execução de um laço incondicional que permite a modificação de instruções do laço.

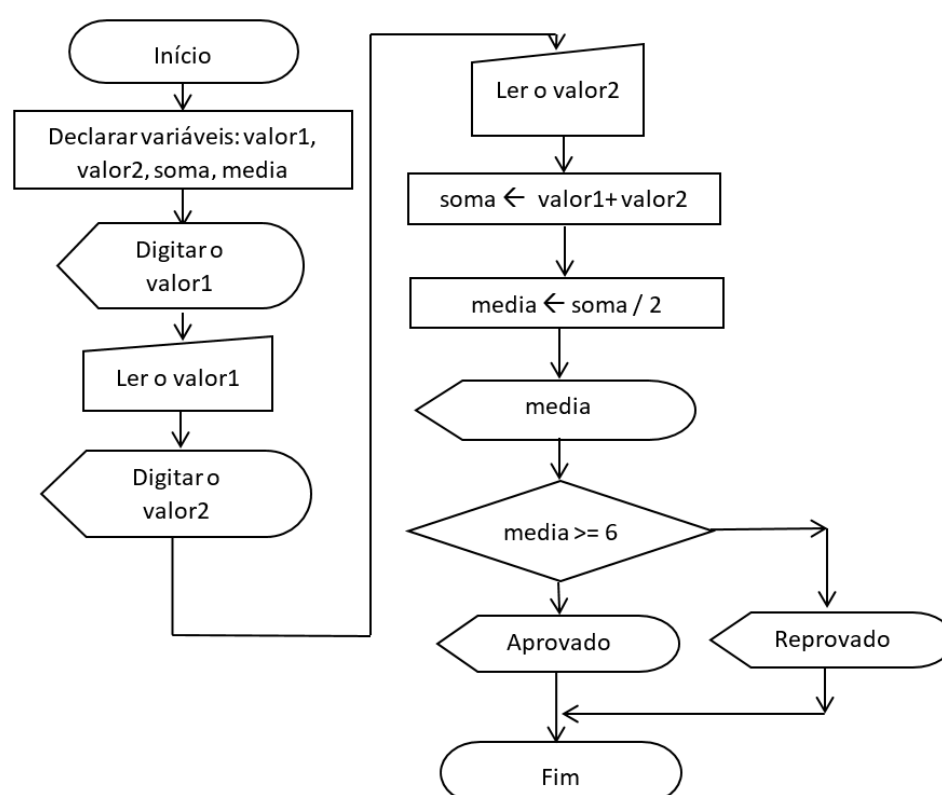


Linha

Representa os vínculos existentes entre os símbolos de um diagrama de blocos.

A seguir veja a solução de um algoritmo utilizando diagrama de blocos. Clique nos símbolos do diagrama para saber mais sobre cada passo.

Para visualizar o vídeo, acesse seu material digital.



Pseudocódigo

O pseudocódigo é considerado uma ferramenta que pode auxiliar a programação, ela pode ser escrita em palavras similares ao inglês ou português para facilitar a interpretação e desenvolvimento de um programa (AGUILAR, 2011).

Exemplo de pseudocódigo que calcula a média das notas dos alunos de um curso:

```
1. calculo_media;  
2.     var  
3.     real: valor1, valor2, soma, media;  
4.  Início  
5.     escreva (“Digite o valor 1”);  
6.     leia valor1;  
7.     escreva (“Digite valor 2”);  
8.     leia valor2;  
9.     soma ← valor1 + valor2;  
10.    soma ← soma/2;  
11.    escreva(“A media do aluno e:”; media);  
12.    se (media >=6)  
13.        então escreva(“Aluno Aprovado”);  
14.        senão escreva(“Aluno Reprovado”);  
15.    Fim se;  
16. Fim.
```

[Comentários](#)

Exemplo de algoritmo escrito em pseudocódigo e executado em [Visualg](#) :

```
algoritmo "media"  
  
var  
  
    valor1, valor2, soma, media: real  
  
inicio  
  
    Escreval("Digite o valor da nota 1: ")  
  
    Leia (valor1)  
  
    Escreval("Digite o valor da nota 2: ")  
  
    Leia (valor2)  
  
    soma ← (valor1 + valor2)  
  
    soma ← (soma / 2 )  
  
    soma ← (soma / 2 )  
  
    Escreval(“A media do aluno e:” media)
```

```
se media >=6 entao  
  
    escreval ("Aluno Aprovado média = ", media)  
  
senao  
  
    escreval ("Aluno Reprovado média = ",media)  
  
fimse
```

Finalgoritmo

Perceba que os parâmetros utilizados também são considerados um algoritmo do tipo português estruturado, ou seja, de fácil entendimento e interpretação.

Paradigmas de programação

Após os estudos de algoritmos e as suas formas de construções, Manzano (2015) coloca em destaque os paradigmas de programação, que são caracterizados pelos paradigmas da:

Programação estruturada

Onde o algoritmo é construído como sequência linear de funções ou módulo.

Programação Orientada a Objetos

Onde o programador abstrai um programa como uma coleção de objetos que interagem entre si.

Os algoritmos são as bases para criação de um programa de computador, onde diversas aplicações poderão ocorrer. Um algoritmo bem estruturado vai gerar um programa para solução de um problema que antes, parecia complexo. Todas as áreas estão voltadas para a tecnologia e são através de diversas formas de pensamentos que os algoritmos são realizados.

Pesquise mais!

Hoje podemos contar com a ajuda de *softwares* específicos para construção de diagrama de blocos (fluxogramas), entre eles, você pode usar o Lucidchart que é um gerador de fluxograma on-line e gratuito.

LUCIDCHART. 2020. Página inicial.

Outro *software* muito utilizado é o “Dia”. Você pode fazer o *download* no site de Dia Diagram Editor (2020).

DIA DIAGRAM EDITOR. *Dia installer*. 2020.