

## Introdução e Definições de algoritmos

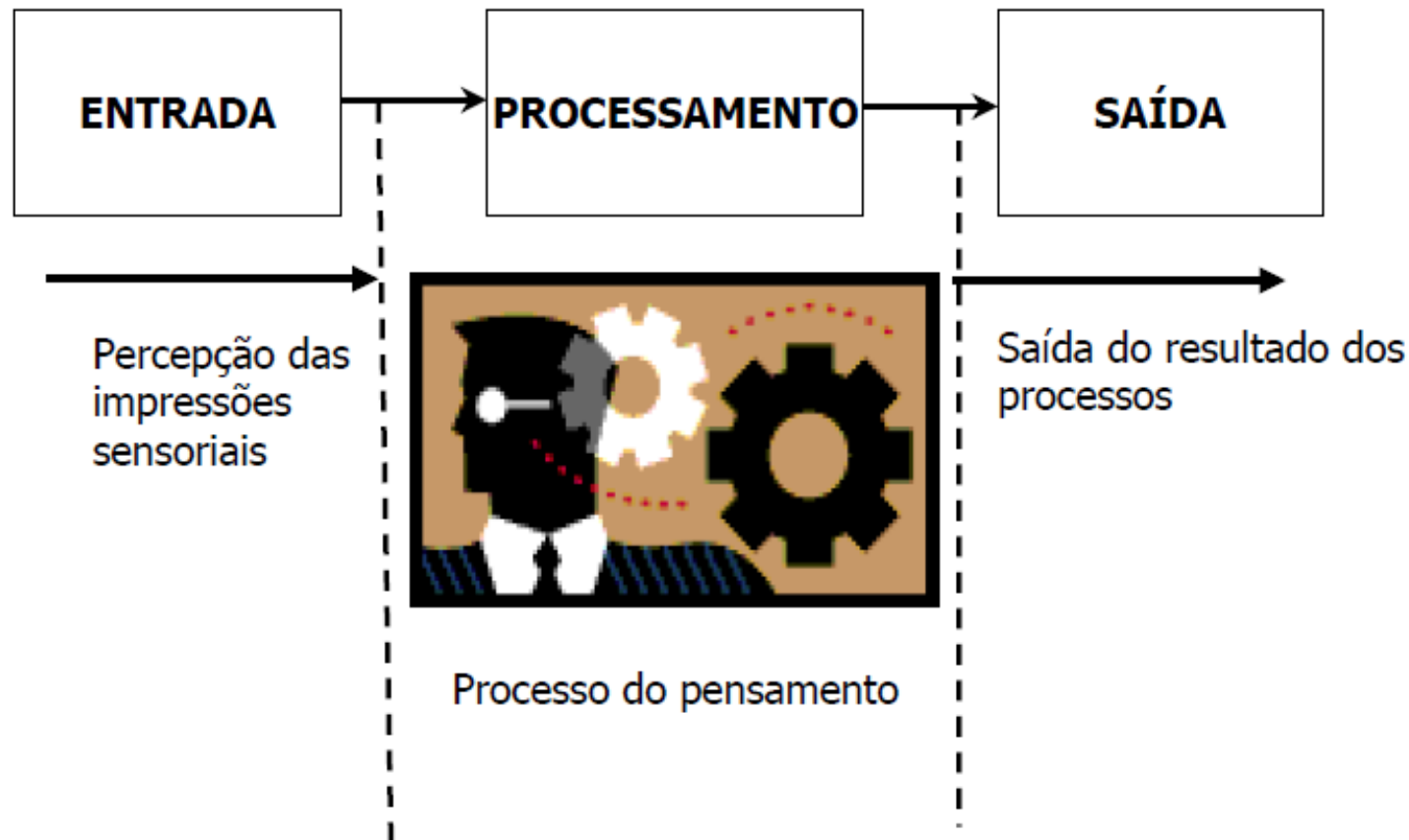
## Introdução

### O que é algoritmo?

Um algoritmo pode ser definido como qualquer procedimento computacional bem definido que toma algum valor ou conjunto de valores como entrada e produz algum valor ou conjunto de valores como saída. Portanto, um algoritmo é uma sequência de passos computacionais que transformam a entrada na saída.

# Algoritmos e Lógica de Programação

## Analogia de um algoritmo com o homem.



## Representação do funcionamento do computador



caracterizada pelos elementos (dados) a serem fornecidos inicialmente.

definido pela execução das ações relativas aos dados de entrada e aos dados intermediários gerados durante a execução.

solução do problema ou objetivo atingido.

## A importância do algoritmo

O algoritmo é extremamente importante, pois o **algoritmo é a relação entre entrada e saída do programa, é o rascunho do programa, o projeto.** E em um projeto antes de colocar a mão na massa é indispensável. **Enquanto a implementação é a função dos pedreiros, o algoritmo é a função dos engenheiros.**



## Introdução

### O que é Programação Estruturada?

A programação estruturada é uma técnica de programação, independente da linguagem de programação, que tem como objetivo construir programas claros, legíveis, eficientes e de fácil manutenção.

## Programas de computador

Um programa é um conjunto de instruções escrito em uma determinada linguagem que diz ao computador o que deve ser feito. Existem muitas formas e diferentes tipos de linguagens de programação, cada qual com uma finalidade específica.

As linguagens podem ser classificadas em níveis, que vão desde o nível de dispositivo e lógico digital (conhecido como nível das portas lógicas, formadas pelos transistores e demais componentes de hardware), chamados de **baixo nível**, até o de linguagem orientada a problemas, chamado de **alto nível**.

*As linguagens de alto nível estão associadas ao fato de que nelas utilizam-se linguagens que procuram aproximar-se das linguagens naturais, usadas pelas pessoas.*

## Programas de computador

*As linguagens de **alto nível** estão associadas ao fato de que nelas utilizam-se linguagens que procuram aproximar-se das linguagens naturais, usadas pelas pessoas.*

*Já as linguagens de **baixo nível** estão associadas ao fato de que nelas utilizam-se linguagens que procuram aproximar-se das linguagens máquina, ou seja, bit's e bytes.*



## Programas de computador

A maioria dos computadores modernos usam dois ou mais níveis de programação, sendo que, os programas escritos em linguagem de alto nível precisam ser traduzidos para o nível mais baixo, de forma que possam ser “entendidos” ou executados pela máquina.

## História da Programação Estruturada

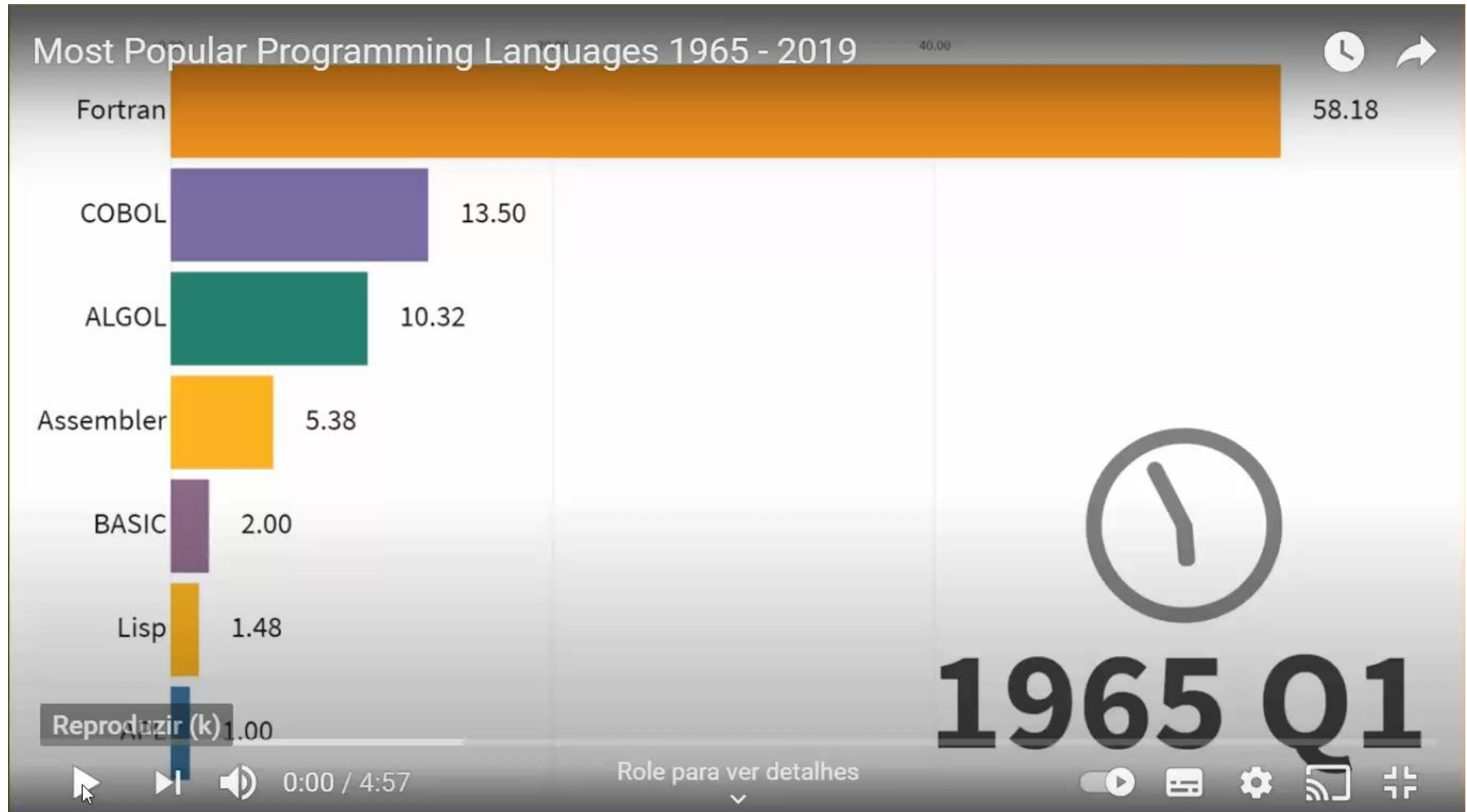
A programação estruturada foi apresentada no início dos anos 70 pelo suíço Niklaus Wirth. Seu uso foi ligado às linguagens de 3ª geração como PL/1, Fortran, Algol 68, Cobol, Linguagem C e Pascal.

## Programas de computador

### Linguagem de programação

A linguagem de programação pode ser entendida como um conjunto de palavras (vocabulário) e um conjunto de regras gramaticais (para relacionar essas palavras) usados para instruir o sistema de computação a realizar tarefas específicas e com isso, criar os programas.

# Algoritmos e Lógica de Programação



## Programação Estruturada é rígida.

A programação estruturada impõe uma disciplina rígida de programação que faz uso de três estruturas de controle para a construção da lógica de um programa:

- Estrutura de sequência
- Estruturas de seleção
- Estruturas de repetição

## Estrutura de Sequência

A estrutura de sequencia define que as instruções do programa são executadas sequencialmente, de cima para baixo, linha a linha, do programa, de forma sequencial.

...

instrução 1

instrução 2

instrução 3

instrução N

...

## Estrutura de Seleção

A estrutura de seleção permite que o fluxo de execução das instruções seja executado de acordo com uma condição lógica que é avaliada e caso seja verdadeira permite que uma ou um grupo de instruções seja executado.

Se <condição for verdadeira> Então

instrução 1

instrução 2

instrução N

FimSe

## Estrutura de Repetição

Esta estrutura de controle de repetição permite que um grupo de instruções seja executado repetidamente de acordo com uma condição lógica.

Enquanto <condição for verdadeira> Faça

    instrução 1

    instrução 2

    instrução N

FimEnquanto



# Algoritmos e Lógica de Programação

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.

# Algoritmos e Lógica de Programação

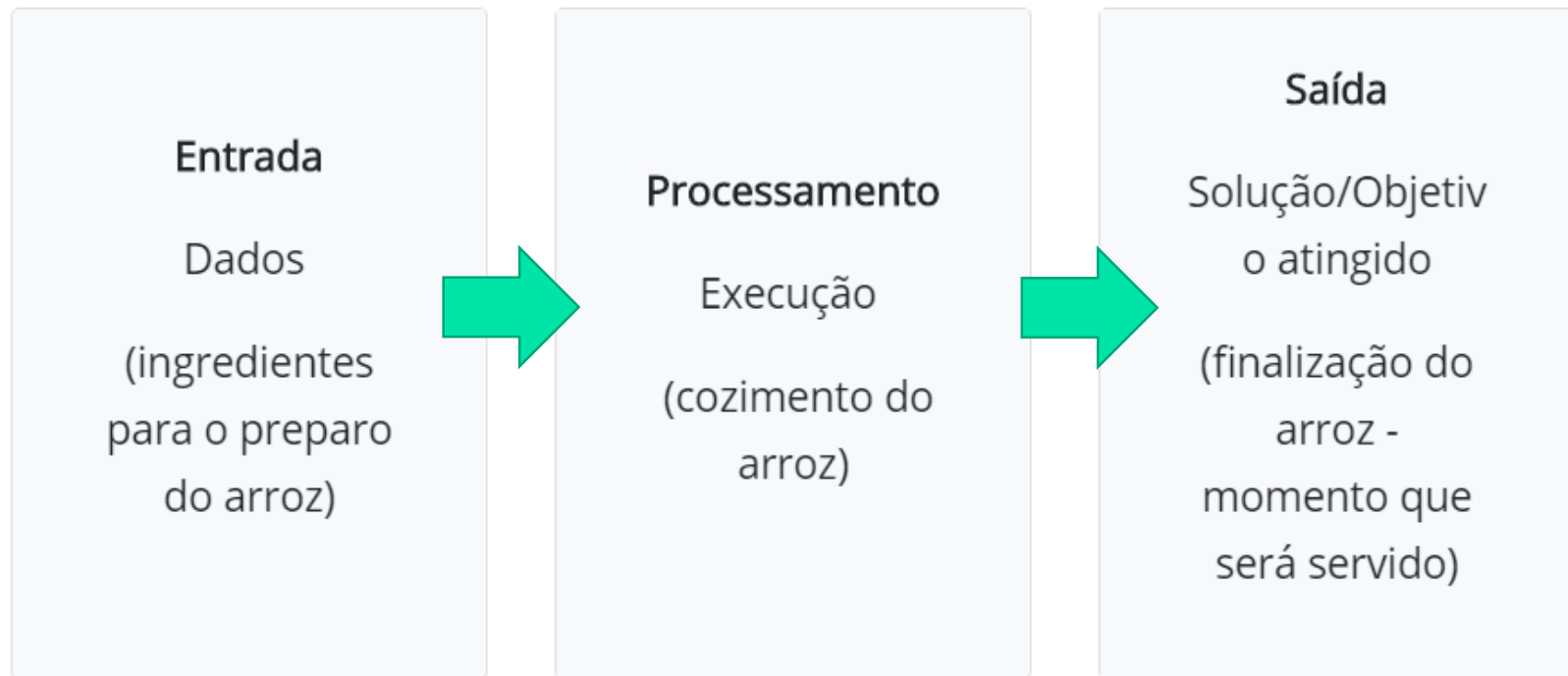
## Algoritmo para efetuar o cozimento de um arroz (mais detalhado) ^

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.

# Algoritmos e Lógica de Programação

## 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).

## Problema Solucionado com Linguagem Natural

### Problema:

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

### 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.

# Algoritmos e Lógica de Programação

## Problema Solucionado com Linguagem Natural

### Problema:

Exemplo é o algoritmo para calcular o máximo divisor comum, o famoso "MDC":

### Solução

1. Início
2. Entrar com o valor "a"
3. Entrar com o valor "b"
4. Atribuir em "r" o resto da divisão de "a" por "b"
5. Substituir a por b
6. Substituir b por r
7. Repetir os passos 4, 5 e 6 até que "a" não possa ser mais dividido por "b", então "a" é considerado o mdc.
8. Fim

De acordo com a solução, o resultado fica:

$$\text{MDC}(480,130) = 10$$

# Algoritmos e Lógica de Programação

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:

Valor1  $\leftarrow$  8

nome  $\leftarrow$  "marcio"

Significa que o número "8" está sendo atribuído para variável "valor1" e que o texto "marcio" está sendo 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).

## Principais símbolos utilizados em um diagrama de blocos/fluxograma



### 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

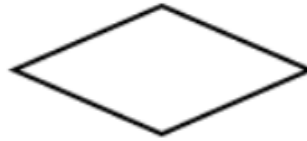
Representa a execução de ações de processamento.

## Principais símbolos utilizados em um diagrama de blocos/fluxograma



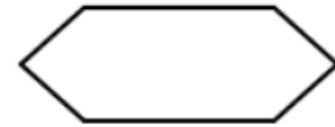
### Exibição

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



### Decisão

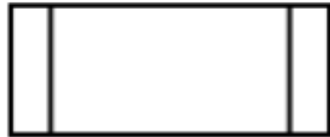
Representa os desvios condicionais nas operações de tomada de decisão e laços condicionais para repetição de alguns trechos do programa.



### Preparação

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

## Principais símbolos utilizados em um diagrama de blocos/fluxograma



### Processo definido

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



### Conector

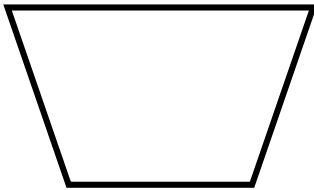
Representa pontos de conexões entre trechos de programas, que podem ser apontados para outras partes do diagrama de bloco.



### Linha

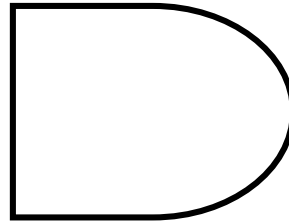
Representa os vínculos existentes entre os símbolos de um diagrama de blocos.

## Principais símbolos utilizados em um diagrama de blocos/fluxograma



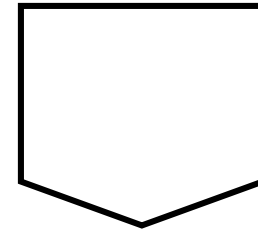
Operação Manual

Representa uma etapa do processo que deve ser executada manualmente, não automaticamente.



Atraso

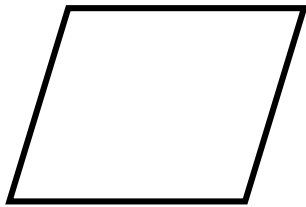
Representa um atraso no processo. No mapeamento de processos, os atrasos costumam ser importantes, pois podem aumentar o custo do produto ou simplesmente atrasar sua produção.



Conector Fora de Página

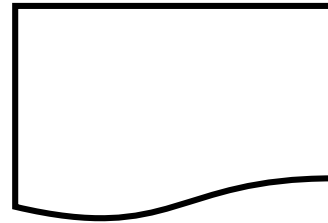
Use este símbolo para conectar processos separados em páginas diferentes com o número da página escrito na forma ou dentro dela.

## Principais símbolos utilizados em um diagrama de blocos/fluxograma



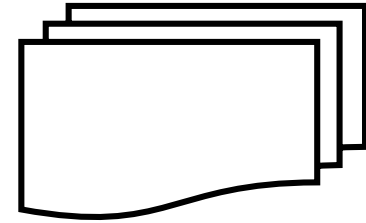
Símbolo de dados  
(entrada / saída)

Indica que as informações estão entrando no processo de fora ou saindo do processo.



Símbolo de Documento

Representa os dados que podem ser lidos por pessoas, como um documento ou relatório impresso.



Símbolo de vários documentos

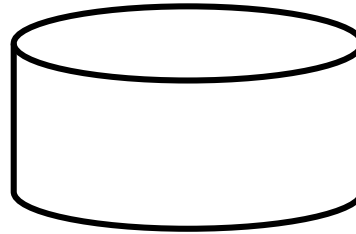
Representa vários documentos no processo.

## Principais símbolos utilizados em um diagrama de blocos/fluxograma



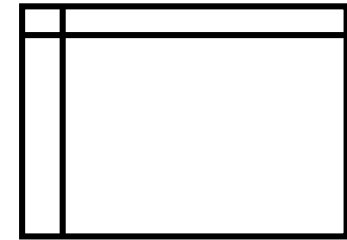
Símbolo de Dados Armazenados

Também conhecido como símbolo de armazenamento de dados. Use este símbolo para indicar onde os dados são armazenados.



Símbolo de banco de dados

Representa dados / informações armazenados no dispositivo de armazenamento que permite a pesquisa e classificação pelos usuários.



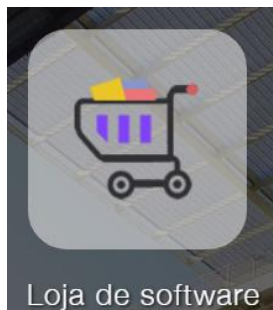
Símbolo de Armazenamento Interno

Normalmente usado nos fluxogramas de design de software e indica que os dados / informações são armazenados na memória durante um programa.

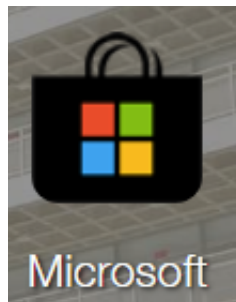
# Algoritmos e Lógica de Programação

## Ferramenta para desenvolvimento de fluxograma a ser instalada.

Passo 1



Passo 2



Passo 3



Acesso a loja

<a href="#">SharePoint Server 2019 ...</a>	Ferramentas de Produ...	Windows	64 bits	Português – Brasil
<a href="#">Skype for Business Serv...</a>	Ferramentas de Produ...	Windows	64 bits	Inglês
<a href="#">Skype for Business Serv...</a>	Ferramentas de Produ...	Windows	64 bits	Português – Brasil
<a href="#">Visio Professional 2019 ...</a>	Ferramentas de Produ...	Windows	64 bits	Inglês
<a href="#">Visio Professional 2019 ...</a>	Ferramentas de Produ...	Windows	64 bits	Português – Brasil
<a href="#">Visual Studio Communi...</a>	Developer Tools	Windows	64 bits	Multilíngue

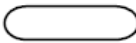





# Algoritmos e Lógica de Programação


Relacione os símbolos utilizados no fluxograma com seus respectivos nomes e, a seguir, assinale a alternativa que contém a sequência correta.


1 - Terminal ( ) 

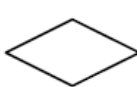
2 - Seta de fluxo de dados ( ) 

3 - Processamento ( ) 

4 - Entrada de dados ou operação manual ( ) 

5 - Entrada e saída de dados ( ) 

6 - Saída de dados em vídeo ( ) 

7 - Saída de dados em impressora ( ) 

8 - Decisão ( ) 

a) 3 – 1 – 6 – 4 – 7 – 5 – 8 – 2

b) 3 – 1 – 6 – 5 – 7 – 4 – 8 – 2

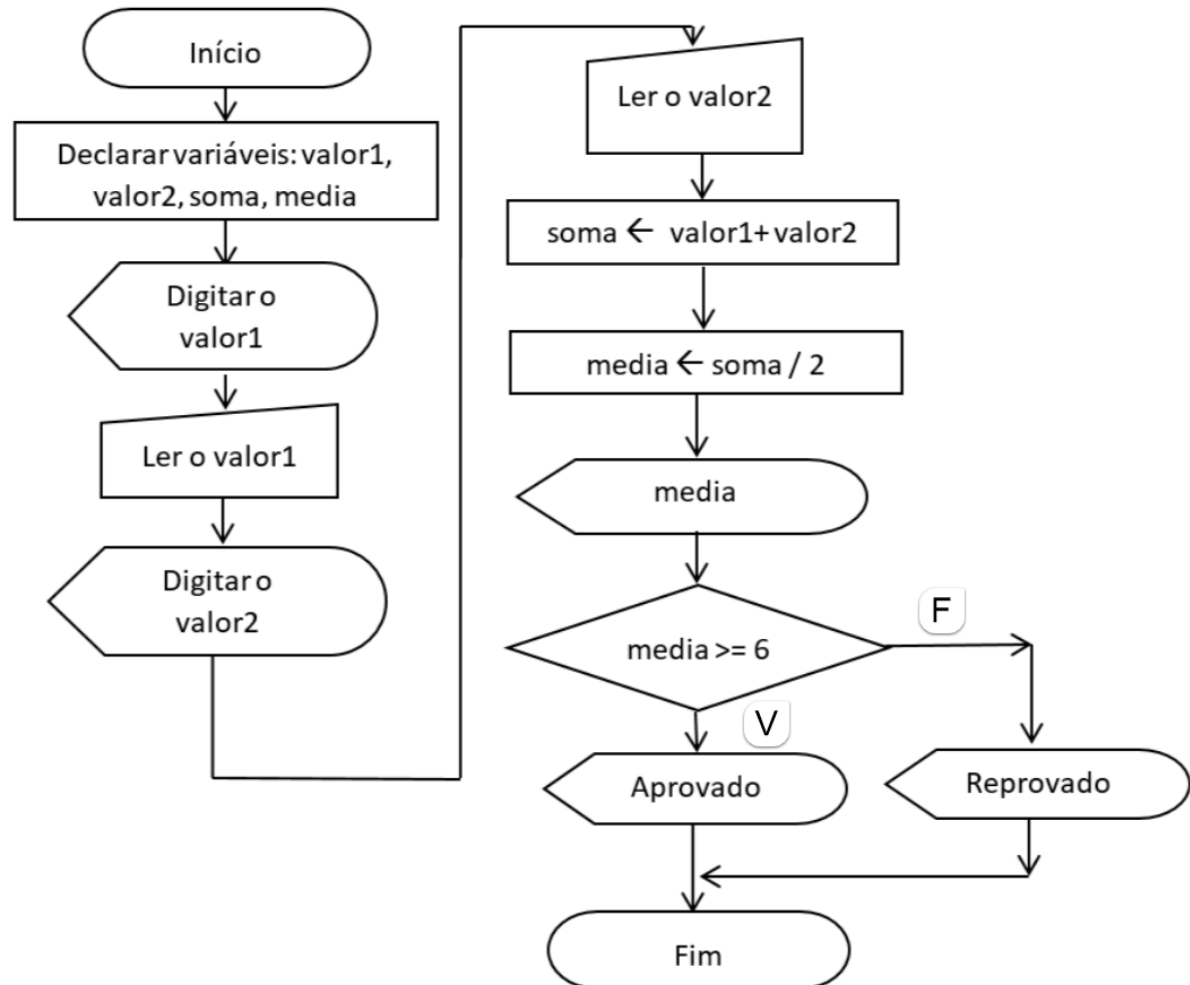
c) 1 – 3 – 4 – 6 – 5 – 7 – 2 – 8

d) 1 – 3 – 6 – 4 – 7 – 5 – 8 – 2

# Algoritmos e Lógica de Programação

## Exemplo de desenvolvimento de fluxograma

Desenvolver algoritmo que capture duas notas do aluno, calcule a média da nota. Se media for maior ou igual a 6 apresente o valor "Aprovado" caso contrário apresentar o valor "Reprovado".



## Paradigmas de Programação

### 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.

## Questionário de Apoio aos Estudos

1. O que é algoritmo?
2. Qual a definição de programação estruturada?
3. Por que a programação estruturada é classificada como rígida?
4. Quais as três estruturas de controle para a construção da lógica de um programa?
5. Descreva a estrutura de sequência na construção de um programa.
6. Descreva a estrutura de seleção na construção de um programa.
7. Descreva a estrutura de repetição na construção de um programa.
8. O que linguagem natural na aplicação computacional?
9. Resolva o problema descrito abaixo utilizando linguagem natural.  
Escreva um programa que leia dois números inteiros e compare-os. mostrando na tela uma mensagem:
  - O primeiro valor é maior
  - O segundo valor é maior
  - Não existe valor maior, os dois são iguais
10. O que são variáveis?
11. O que é, e para que serve o diagrama de blocos (fluxograma)?
12. Descreva os principais símbolos do diagrama de blocos (fluxogramas).
13. Desenvolva um fluxograma que resolva o seguinte problema: "O sistema deve perguntar o salário de um funcionário e calcule o valor do seu aumento. Para salários superiores a R\$1250,00, calcule um aumento de 10%. Para os inferiores ou iguais, o aumento é de 15%."