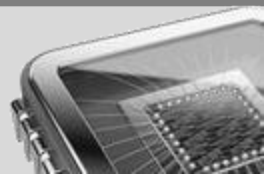
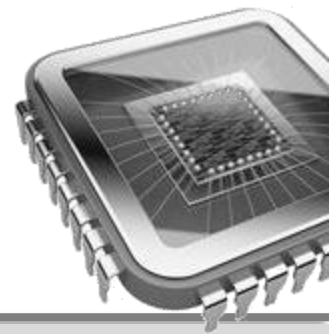


# Introdução à Programação

## Aula 02

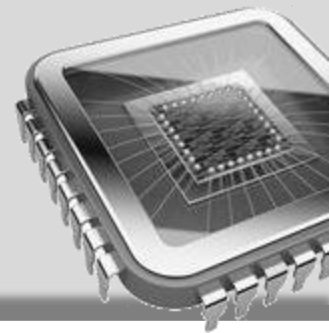
Prof. Max Santana Rolemberg Farias  
max.santana@univasf.edu.br  
Colegiado de Engenharia de Computação



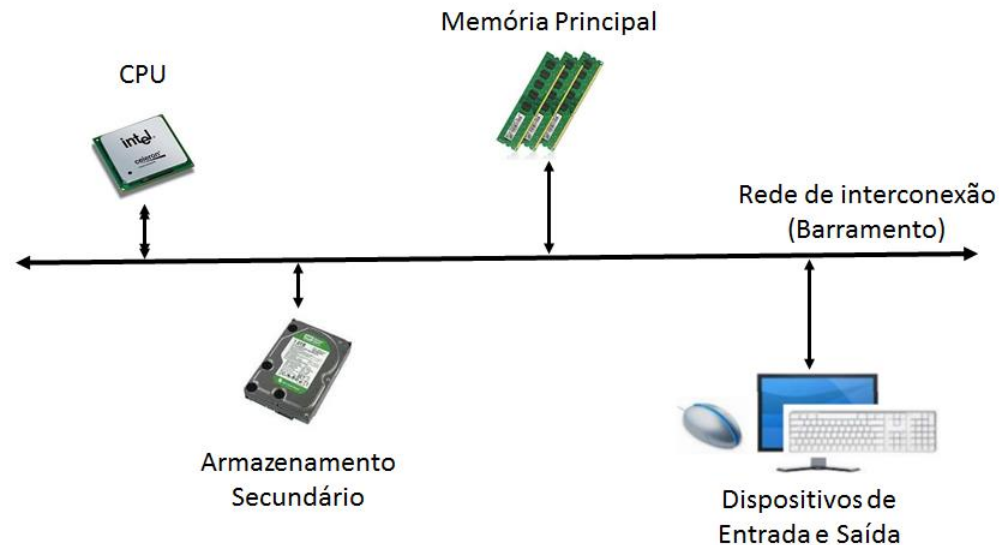


# **QUAIS SÃO OS COMPONENTES BÁSICOS DO HW DE UM SISTEMA COMPUTACIONAL?**

# Modelo Básico de um Sistema Computacional

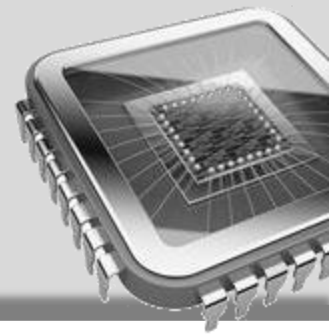


- O modelo básico é constituído por:
  - Unidade de Entrada
  - Unidade de Saída
  - Unidade Central de Processamento
  - Memória Principal
  - Memória Secundária

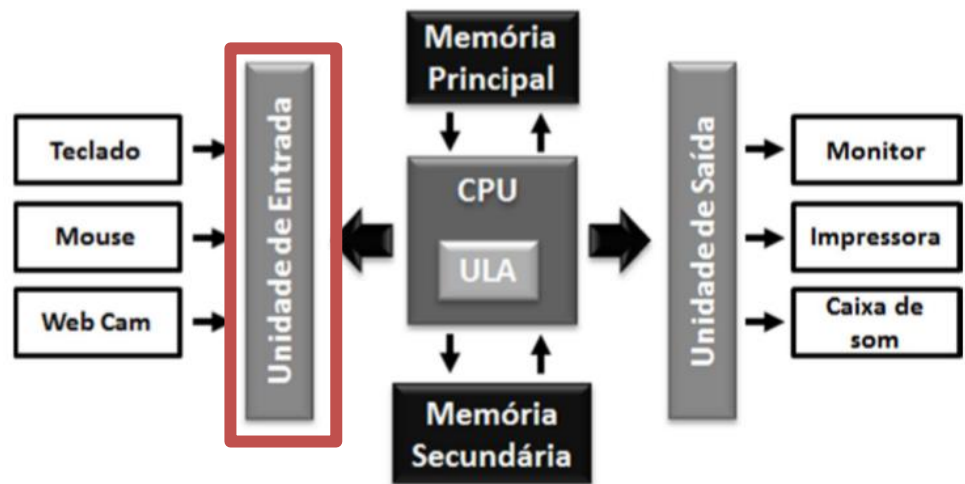


Von Neumann

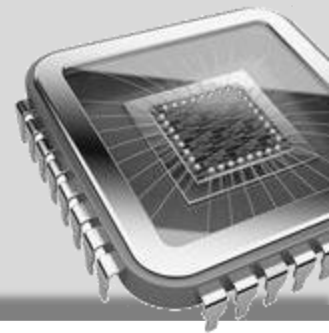
# Organização do Hardware de um Sistema Computacional



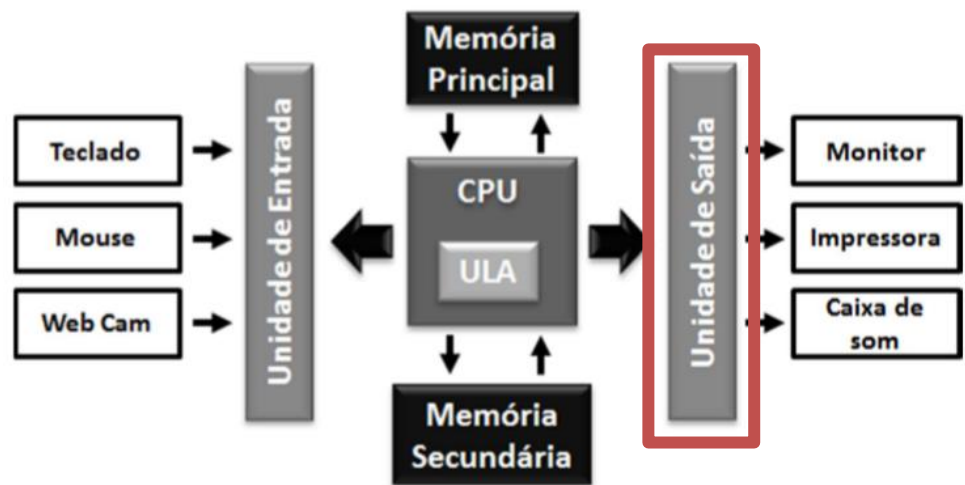
- Unidade de entrada
  - Utilizada pelo sistema para receber instruções ou dados externos que serão processados.
  - Permite que o usuário interaja com o sistema através dos dispositivos de entrada.
    - Teclado
    - Mouse



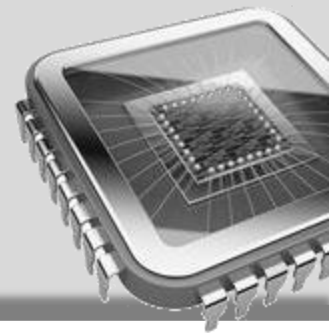
# Organização do Hardware de um Sistema Computacional (continuação)



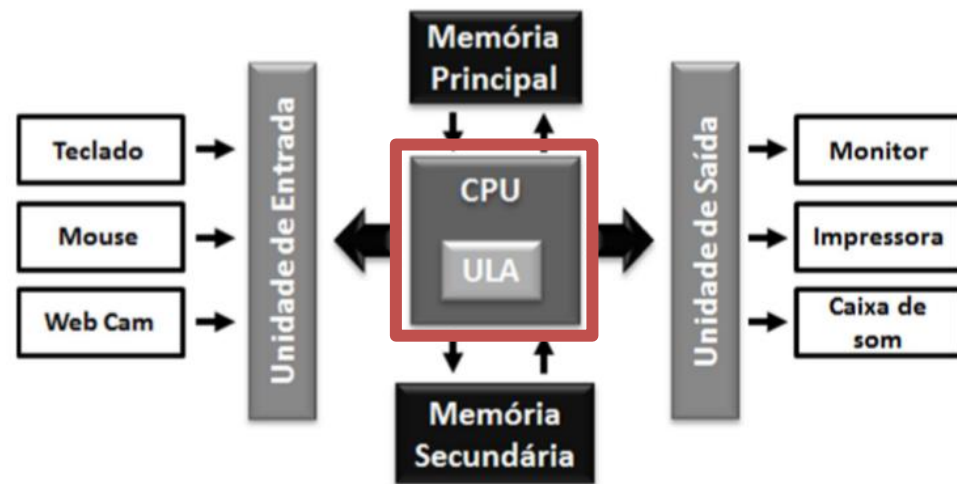
- Unidade de saída
  - Usada para exibir os resultados do processamento nos dispositivos de saída.
    - Monitor
    - Impressora
    - Caixa de som
    - etc



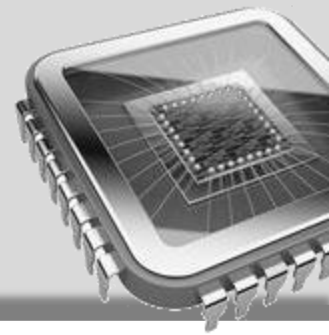
# Organização do Hardware de um Sistema Computacional (continuação)



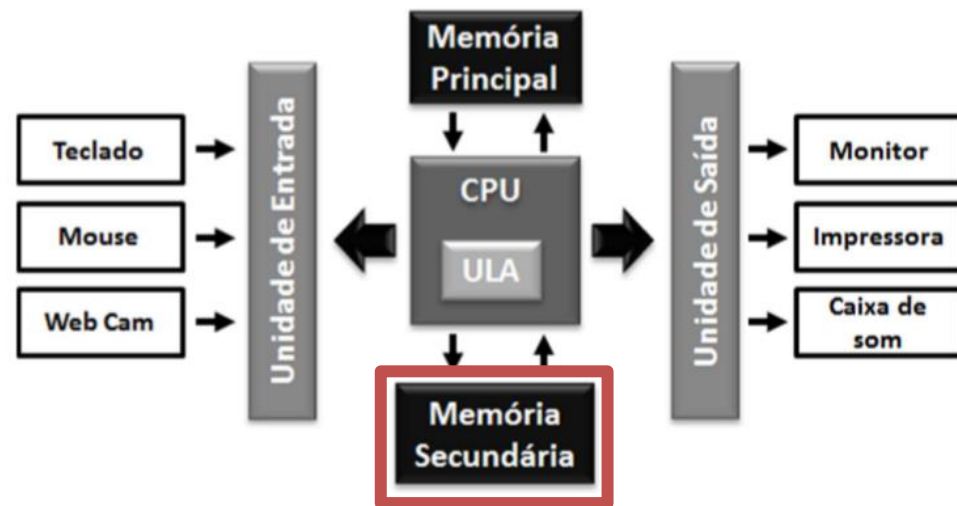
- Processador (CPU)
  - Responsável por todo o processamento e gerenciamento do sistema.
- Unidade Lógica Aritmética (ULA)
  - Responsável pelo cálculo matemáticos.



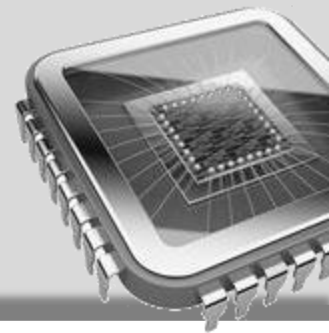
# Organização do Hardware de um Sistema Computacional (continuação)



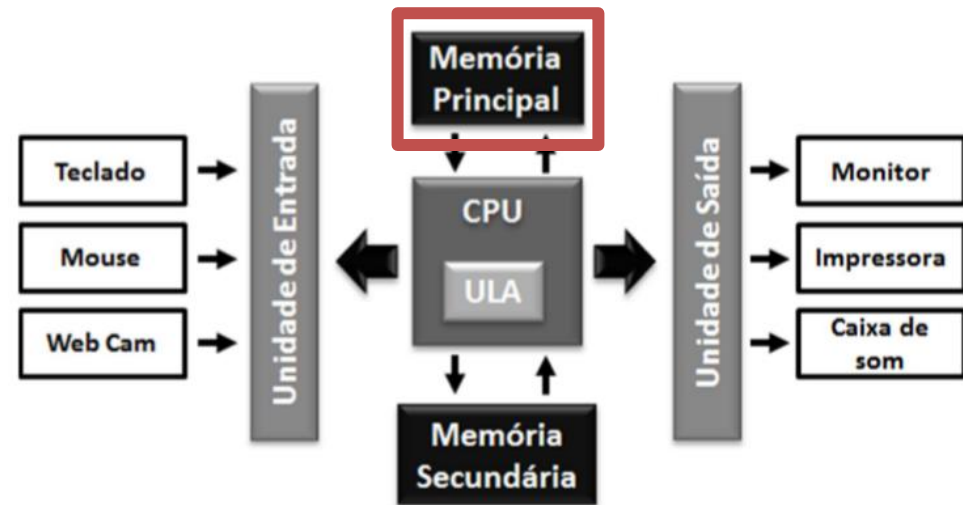
- Memória secundária
  - Utilizada pelo sistema para armazenar instruções e informações por prazo indeterminado.
  - Maior capacidade de armazenamento.
  - Apresenta um acesso lento.
    - HD
    - Disquetes

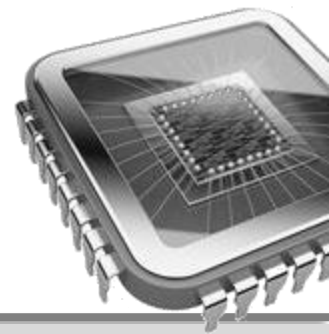


# Organização do Hardware de um Sistema Computacional (continuação)



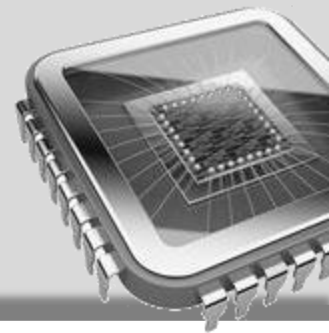
- Memória principal
  - Usada pelo processador para armazenar instruções e informações enquanto o sistema está ligado.
  - Essa memória também é conhecida como memória RAM





# **COMO OS SISTEMAS COMPUTACIONAIS DIVIDE E ORGANIZA A MEMÓRIA PRINCIPAL?**

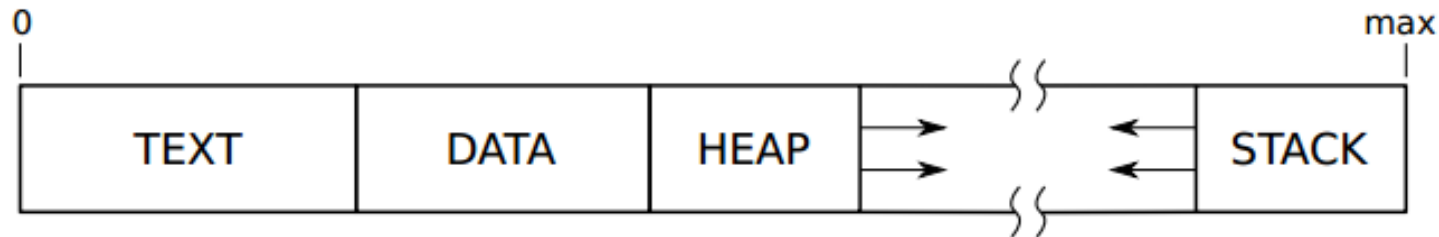
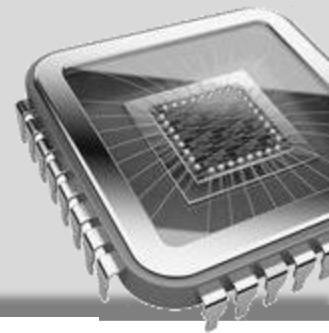
# Divisão da Memória de um Sistema Computacional



- Uma memória é um conjunto de bytes que armazenam informações.
  - Para utilizar esses conjuntos de bytes é necessário a utilização de um SO.
  - O SO é o responsável por dividir todas as utilizações do HW em Processos

Endereços									
Dec.	Binário								
255	11111111								
254	11111110								
253	11111101								
252	11111100								
251	11111011								
250	11111010								
<hr/>									
130	00000000								
129	00000000	0	0	0	1	1	1	0	0
128	00000000	1	1	0	0	1	0	0	0
<hr/>									
117	01110101								
116	01110100								
115	01110011								
<hr/>									
2	00000010								
1	00000001								
0	00000000								

# Divisão da Memória de um Sistema Computacional (continuação)



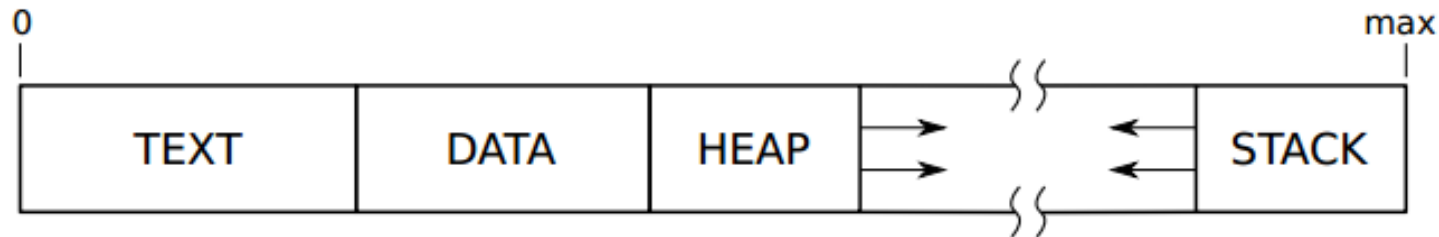
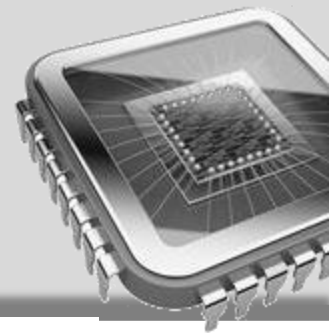
- **TEXT**

- Contém o código a ser executado.
- Gerado durante a compilação e a ligação com as bibliotecas
- Esta área tem tamanho fixo, calculado durante a compilação
- Só deve estar acessível para leitura e execução

- **DATA**

- Contém os dados estáticos usados pelo programa
  - Variáveis globais
  - Variáveis locais estáticas
- Esta área tem tamanho fixo
- Deve estar acessível para leitura e escrita.
- Não pode ser acessível para execução.

# Divisão da Memória de um Sistema Computacional (continuação)



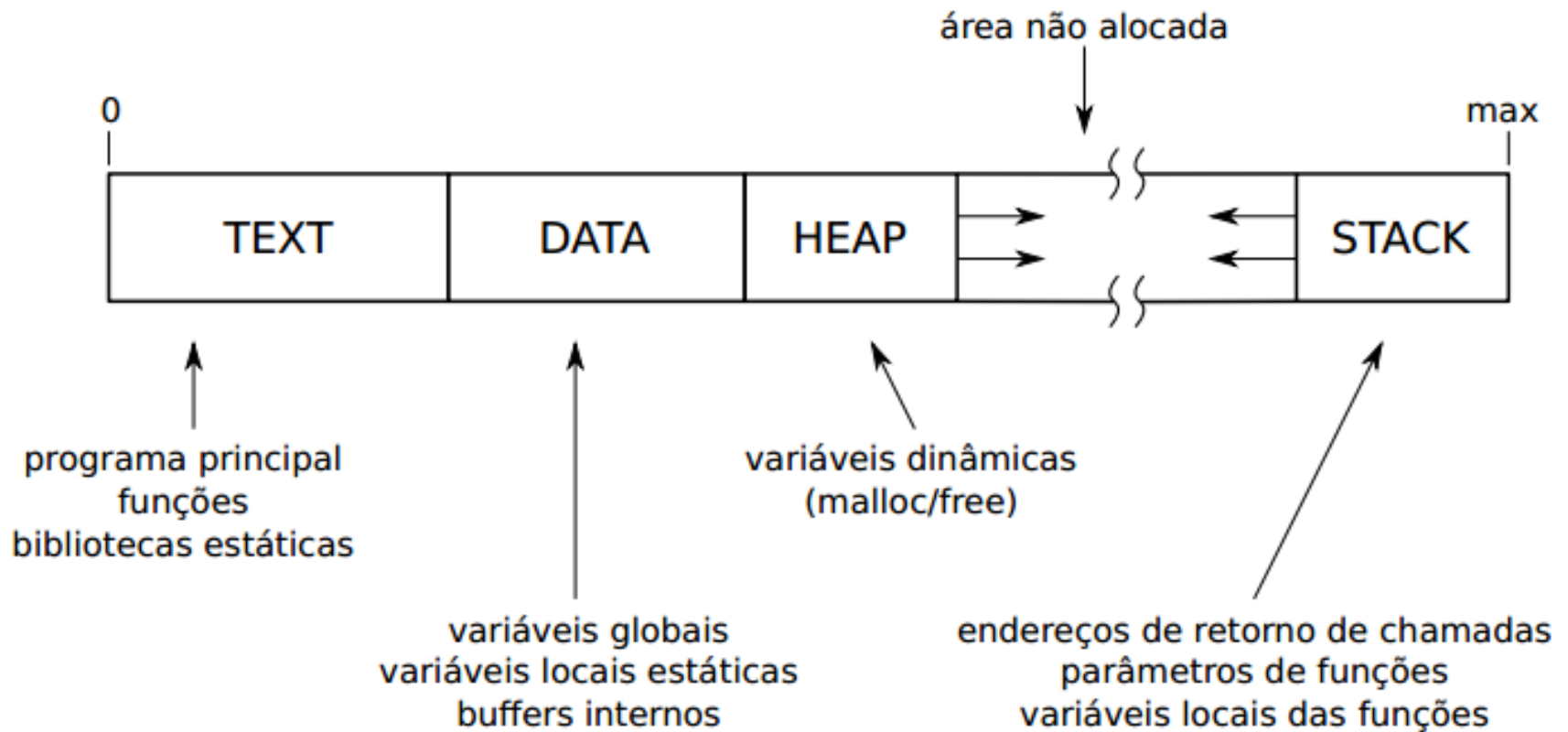
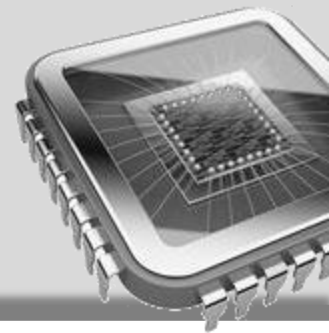
- **HEAP**

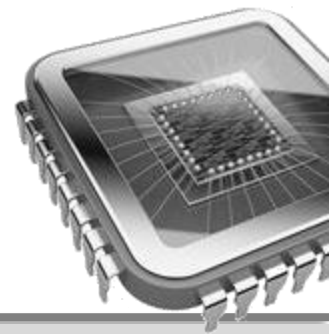
- Usada para armazenar dados alocados dinamicamente
- Esta área tem tamanho variável

- **STACK**

- Usada para manter a pilha de execução do programa
- Estrutura responsável por gerenciar o fluxo de execução
- Armazena os parâmetros, variáveis locais e o retorno

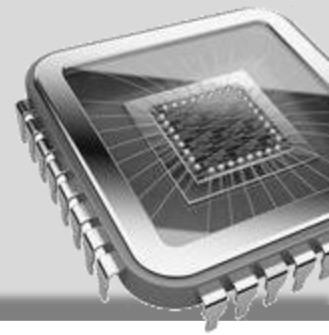
# Divisão da Memória de um Sistema Computacional (continuação)



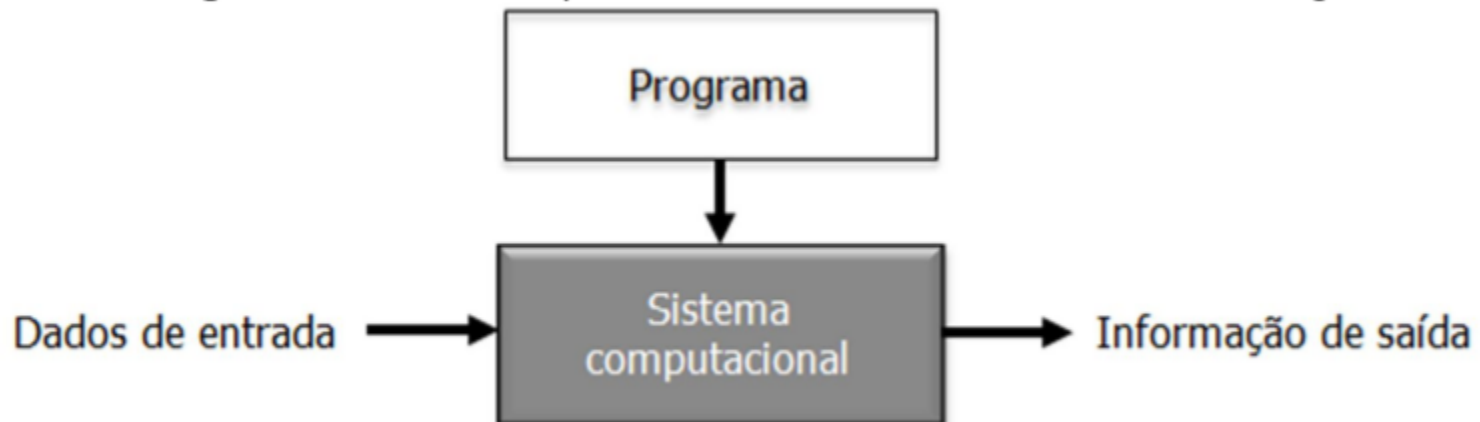


# **COMO INSTRUIR OS SISTEMAS COMPUTACIONAIS PARA FAZER DETERMINADA TAREFA DE PROCESSAMENTO?**

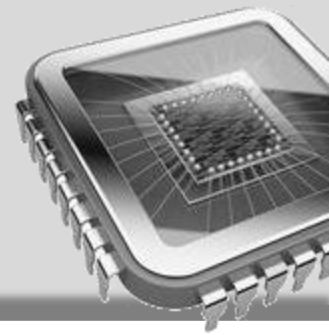
# Introdução à Programação



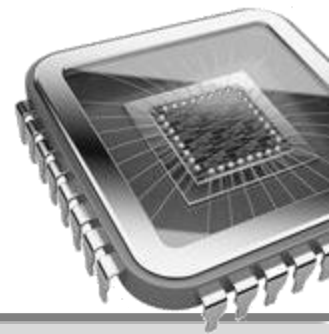
- Um sistema computacional necessita de um ou vários programas para funcionar corretamente.



# Introdução à Programação (continuação)

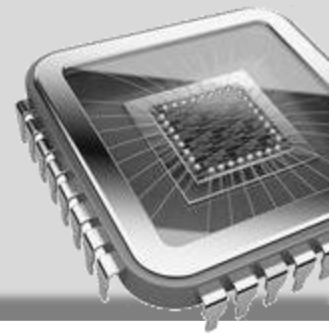


- Nos programas são escritas as instruções necessárias para que o sistema computacional possa gerar corretamente os resultados.
  - Essas instruções precisam ser escritas em uma linguagem que o sistema possa entender.



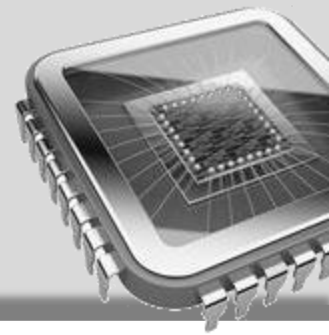
# **QUAIS AS LINGUAGENS COMPREENDIDAS PELOS SISTEMAS COMPUTACIONAIS?**

# Introdução à Programação (continuação)



- A linguagem compreendida pelos sistemas computacionais são chamadas de linguagem de programação.
- As linguagens de programação podem ser divididas em três tipos:
  - Linguagem de máquina.
  - Linguagem simbólica e linguagem de alto nível.

# Introdução à Programação Linguagem de Máquina



- Consiste em sequências de bits.

- Sequências de dígitos 0 (zero) e 1 (um).

- São dependentes do hardware do sistema.

- Cada sistema tem sua própria linguagem de máquina

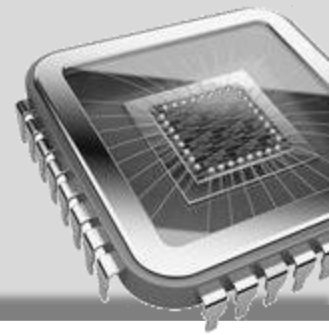
- Linguagem natural.

- Lenta e tediosa.

- Sujeita a erros.

1	4C	01	05	00	00	00	5C
2	01	00	00	0F	00	00	04
3	00	00	48	01	00	00	00
4	01	00	00	00	00	20	00
5	30	60	2E	64	00	00	00
6	00	00	00	00	40	00	30
7	C0	2E	62	73	73	00	00
8	00	00	00	30	00	40	00
9	55	00	00	00	00	89	00
10	E5	00	00	00	83	E4	00
11	F0	00	00	0F	00	00	40
12	2E	70	55	00	00	FF	FF

# Introdução à Programação Linguagem Simbólica



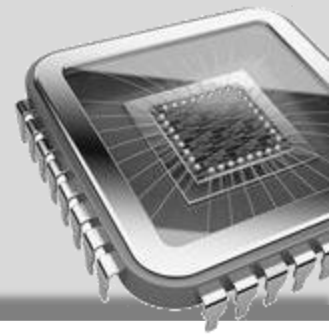
- Utilizam abreviações para representar as operações.
- São mais claras para os humanos

```
1 load $s0, 10
2 load $s1, 11
3 add $s2, $s1, $s0
```



A linguagem simbólica necessita de um montador (*assemblers*) de linguagem de máquina para ser entendido pelo sistema.

# Introdução à Programação Linguagem de Alto Nível

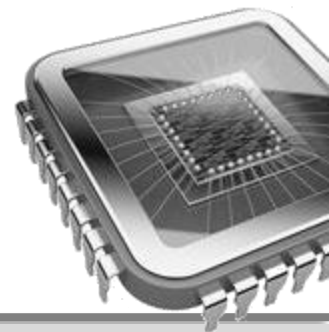


- Instruções parecidas com a linguagem cotidiana
  - Sem necessidade de conhecer muitas das instruções do sistema.
  - Uma única instrução realiza tarefas significativas

```
1  interge num1, num2;  
2  integer soma;  
3  num1 = 10;  
4  num2 = 10;  
5  soma = num1 + num2;
```



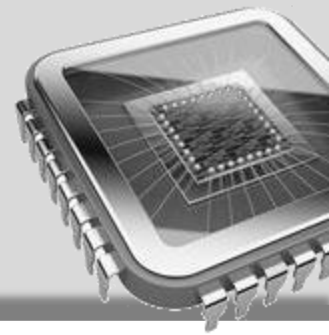
O programa que converte os programas de linguagem de alto nível em linguagem de máquina é chamado de compilador.



# O QUE É UM PROGRAMA?

# Introdução à Programação

## Programa

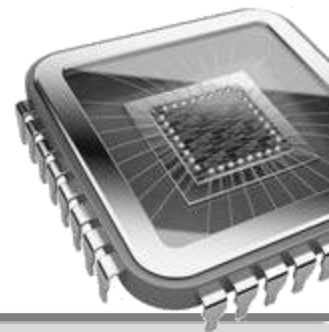


O QUE É?

É um conjunto de instruções que descrevem uma tarefa a ser realizada por um computador.

O QUE É?

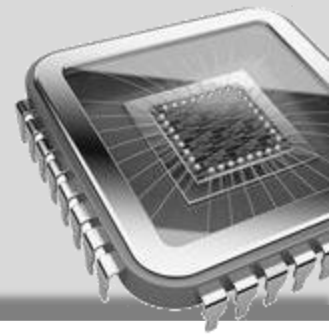
É a codificação de um algoritmo em uma determinada linguagem de programação.



# O QUE É UM ALGORITMO?

# Introdução à Programação

## Algoritmo



O QUE É?

É uma sequência bem definida de procedimentos computacionais (passos) que levam uma entrada a se transformar em uma saída. (Cornen et. al., 1991)

O QUE É?

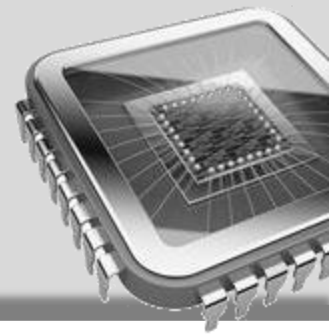
É uma sequência de passos que visa atingir um objetivo bem definido. (Forbellone, 1999)

O QUE É?

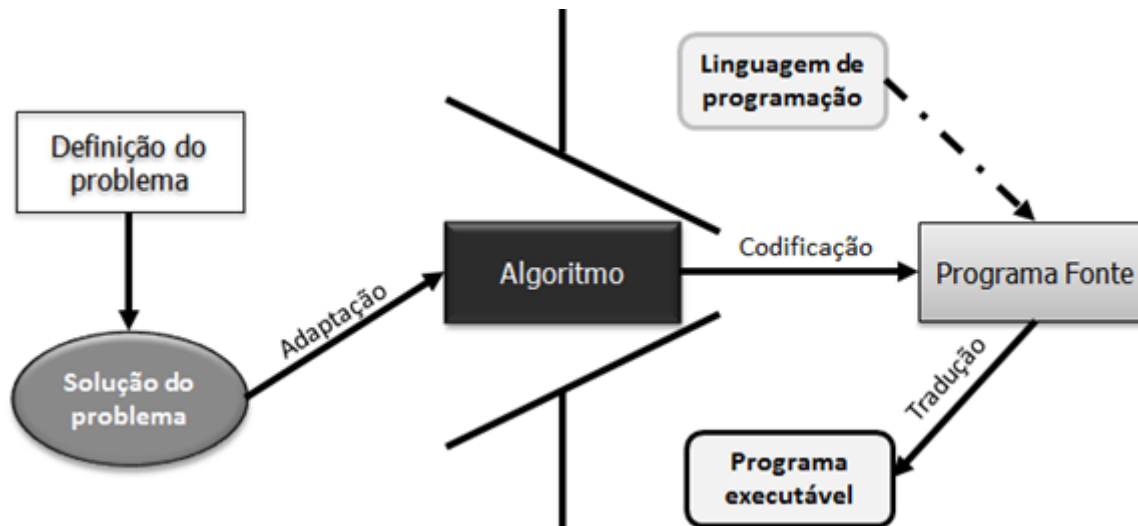
É uma sequência de instruções ou operações cuja execução, em tempo finito, resolve um problema computacional, qualquer que seja sua instância (Salvetti, 1999)

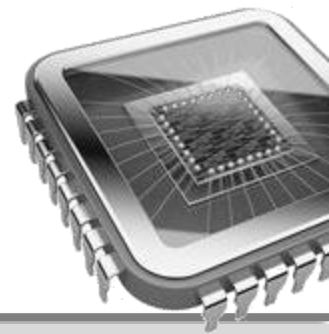
# Introdução à Programação

## Algoritmo



- O algoritmo tem um papel fundamental na programação.
  - É o elo de ligação entre o mundo real e o mundo computacional.

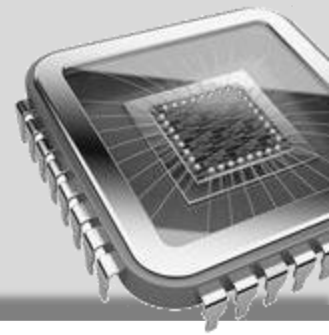




# COMO DESCREVER UM PROBLEMA EM FORMA DE ALGORITMO?

# Introdução à Programação

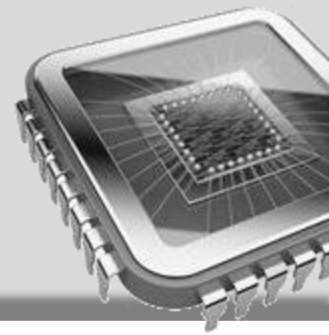
## Algoritmo (continuação)



- Tipos de algoritmos:
  - Descrição narrativa
  - Fluxograma
  - Pseudocódigo (português estruturado).

# Introdução à Programação

## Algoritmo (continuação)



- Algoritmo do tipo descrição narrativa.
  - Consiste em escrever em linguagem natural (português), o passo-a-passo para resolver um problema.



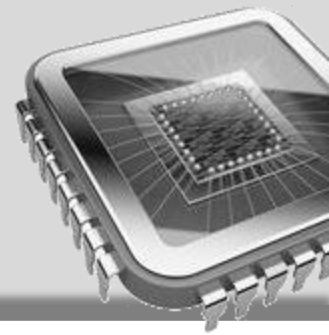
Não é necessário aprender nenhum conceito ou linguagem nova, visto que a linguagem natural já é conhecida.



Dificulta a transcrição para uma linguagem de programação. Porque a linguagem natural abre espaço para várias interpretações.

# Introdução à Programação

## Algoritmo (continuação)



- Algoritmo do tipo fluxograma.
  - Utiliza símbolos gráficos predefinidos para escrever o passo-a-passo a ser seguido para resolver um problema computacional.



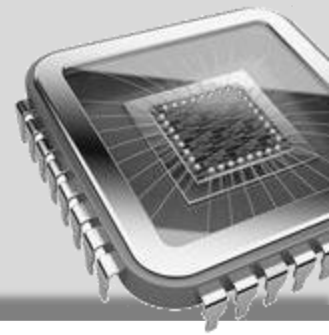
É muito mais fácil entender os elementos gráficos (símbolos) do que entender as palavras



É necessário aprender a simbologia dos fluxogramas. O algoritmo não apresenta muitos detalhes, e além disso, a transcrição para uma linguagem se complica à medida que o algoritmo cresce.

# Introdução à Programação

## Algoritmo (continuação)



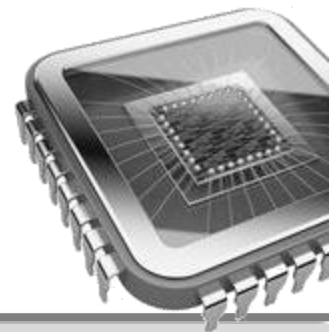
- Algoritmo do tipo pseudocódigo.
  - O passo-a-passo para resolver um problema é escrito de uma forma estruturada e obedecendo algumas regras predefinidas.



A tradução do algoritmo para qualquer linguagem de programação é quase imediata, bastando conhecer a estrutura e as regras da linguagem.



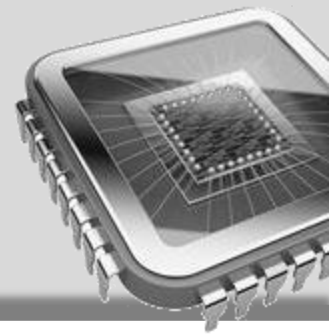
Precisa aprender as regras e a estrutura do pseudocódigo.



# COMO DESENVOLVER UM PROGRAMA?

# Introdução à Programação

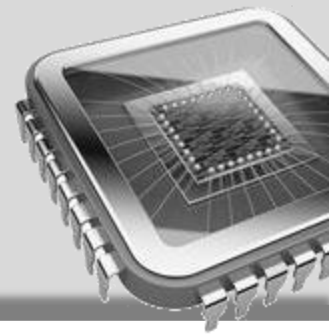
## Desenvolvimento



- Para desenvolver um programa são necessárias as seguintes etapas:
  - Análise
  - Algoritmo
  - Codificação

# Introdução à Programação

## Desenvolvimento (continuação)



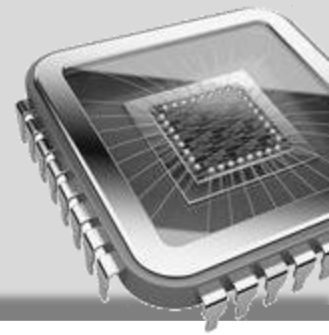
- **Etapa de análise**

– Estuda-se o enunciado do problema para definir os dados de entrada, o processamento e os dados de saída.



# Introdução à Programação

## Desenvolvimento (continuação)

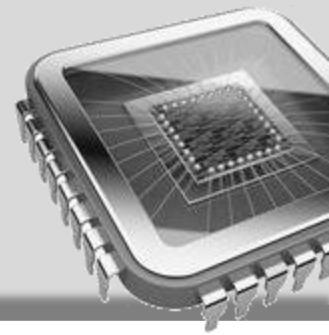


- **Etapa de algoritmo**

- É utilizada tipos de algoritmos para descrever o problema.

# Introdução à Programação

## Desenvolvimento (continuação)



- **Etapa de codificação**

- Nessa etapa o algoritmo é transformado em códigos de uma linguagem de programação.