
Introdução à Ciência da Computação: aula introdutória

Prof. Dr. Danilo Medeiros Eler
danilo.eler@unesp.br

Conteúdo

- Modelos Computacionais
- Componentes de um Modelo
- História da Computação e dos Computadores
- Exercícios

Introdução

- O que você poderia fazer com os ingredientes abaixo?
 - Água
 - Farinha
 - Ovo
 - Fermento

Introdução

- O que você poderia fazer com os ingredientes abaixo?
 - Água
 - Farinha
 - Ovo
 - Fermento
- Poderíamos seguir uma receita e fazer:
 - Pão, pizza, torta, entre outros tipos de alimento

Introdução

- Se você pudesse desenvolver uma máquina para fazer esses tipos de alimento
 - Pão, Pizza, Torta, entre outros
- Como seria um possível modelo para esse tipo de máquina?

Introdução

- Se você pudesse desenvolver uma máquina para fazer esses tipos de alimento
 - Pão, Pizza, Torta, entre outros
- Como seria um possível modelo para esse tipo de máquina?



Introdução

- Se você pudesse desenvolver uma máquina para fazer esses tipos de alimento
 - Pão, Pizza, Torta, entre outros
- Como seria um possível modelo para esse tipo de máquina?

Água
Farinha
Ovos
Fermento



MÁQUINA
DE PÃO



Pão

Introdução

- Se você pudesse desenvolver uma máquina para fazer esses tipos de alimento
 - Pão, Pizza, Torta, entre outros
- Como seria um possível modelo para esse tipo de máquina?



Introdução

- Se você pudesse desenvolver uma máquina para fazer esses tipos de alimento
 - Pão, Pizza, Torta, entre outros
- Como seria um possível modelo para esse tipo de máquina?

Água
Farinha
Ovos
Fermento



MÁQUINA
DE SORVETE



Sorvete

Introdução

- Se você pudesse desenvolver uma máquina para fazer esses tipos de alimento
 - Pão, Pizza, Torta, entre outros
- Como seria um possível modelo para esse tipo de máquina?

Água
Farinha
Ovos
Fermento



MÁQUINA
DE SORVETE



Sorvete

Introdução

- Se você pudesse desenvolver uma máquina para fazer esses tipos de alimento
 - Pão, Pizza, Torta, entre outros
- Como seria um possível modelo para esse tipo de máquina?

Leite condensado
Leite em pó
Leite
Açúcar
Emulsificante



Introdução

- Essa proposta não é um modelo genérico
- Faz somente aquilo que está proposto



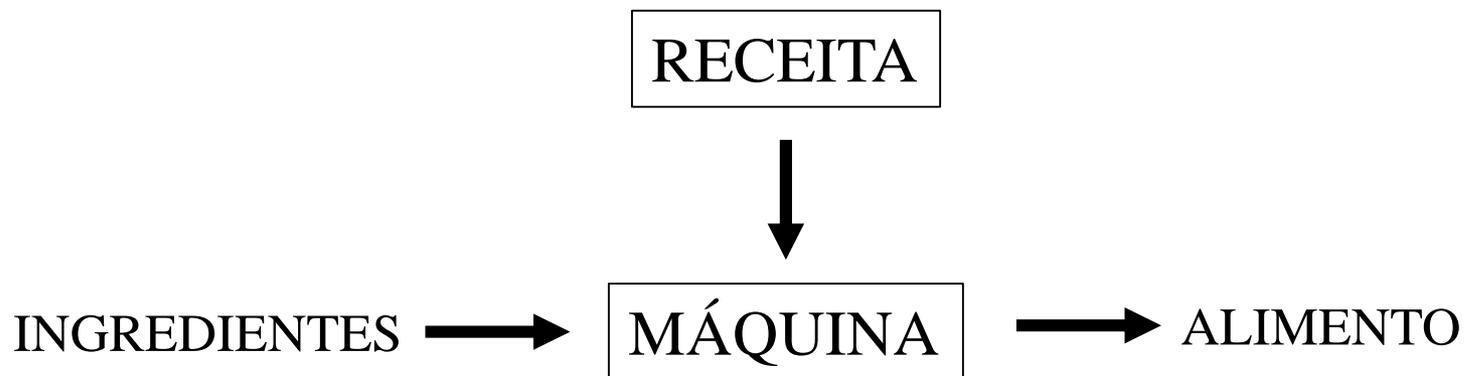
- Como poderíamos melhorar?

Introdução

- Modelo Específico



- Modelo Genérico



Computador

■ Processador de Dados



Computador

■ Processador de Dados



■ Uma caixa-preta

- Aceita inserção de dados
- Processa-os
- Cria informações (resultados)

Computador

- Processador de Dados



- Modelo muito geral

- Na atualidade, uma calculadora seria um computador

Computador

■ Processador de Dados



■ Problemas desse modelo

- ❑ Não especifica o tipo de processamento
- ❑ Não especifica se é possível mais de um tipo de processamento
- ❑ Esse modelo foi elaborado para propósitos específicos

Computador

- Atualmente, computadores são máquinas de propósito geral
 - Podem desempenhar muitos tipos de tarefas diferentes
- O modelo precisa ser transformado para refletir os reais computadores da atualidade

Computador Programável

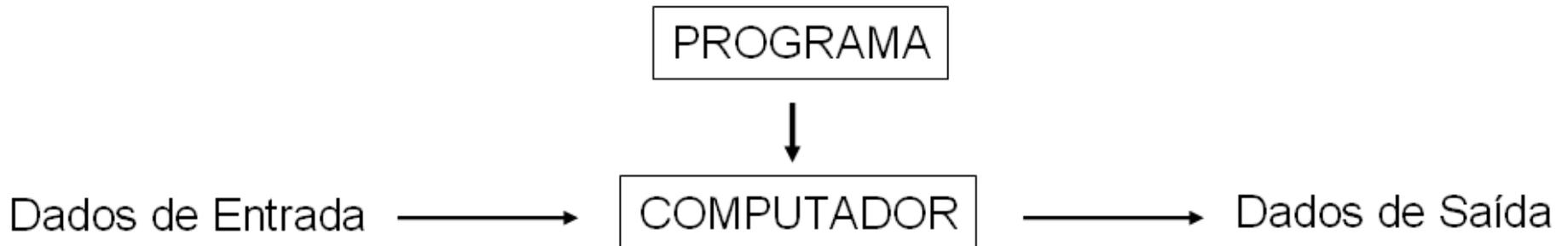
- A ideia de um dispositivo de computação universal foi descrita, pela primeira vez, por Alan Turing, em 1937
- Toda computação poderia ser realizada por um tipo especial de máquina
 - Máquina de Turing
- Estava mais interessado na definição filosófica do que na máquina propriamente dita

Computador Programável

- Modelo de Turing
 - Acrescenta um elemento extra de computação específica: **PROGRAMA**
- Um **PROGRAMA** é um conjunto de instruções que diz ao computador o que fazer com os dados

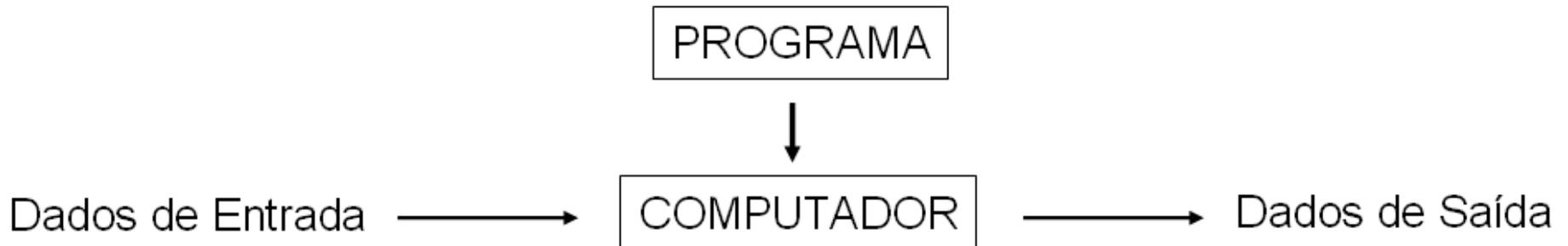
Computador Programável

- Modelo de Turing



Computador Programável

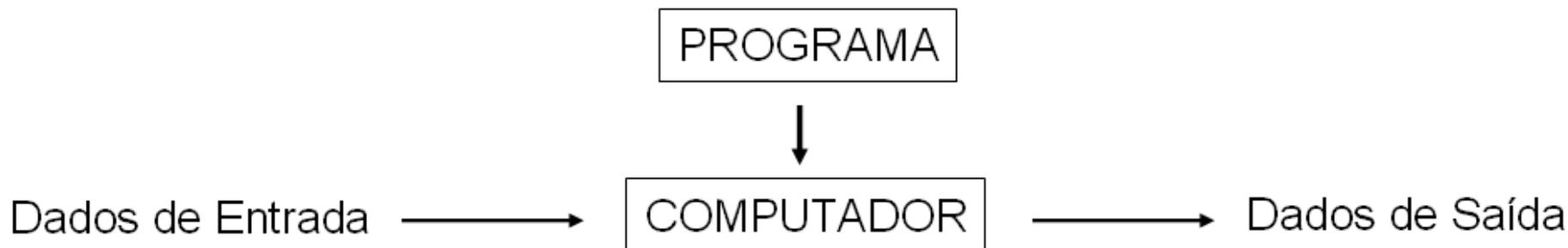
■ Modelo de Turing



- Os **Dados de Saída** dependem de dois fatores
 - Os **Dados de Entrada** e o **Programa**
 - Diferentes resultados podem ser gerados

Comparação entre modelos

■ Modelo de Turing

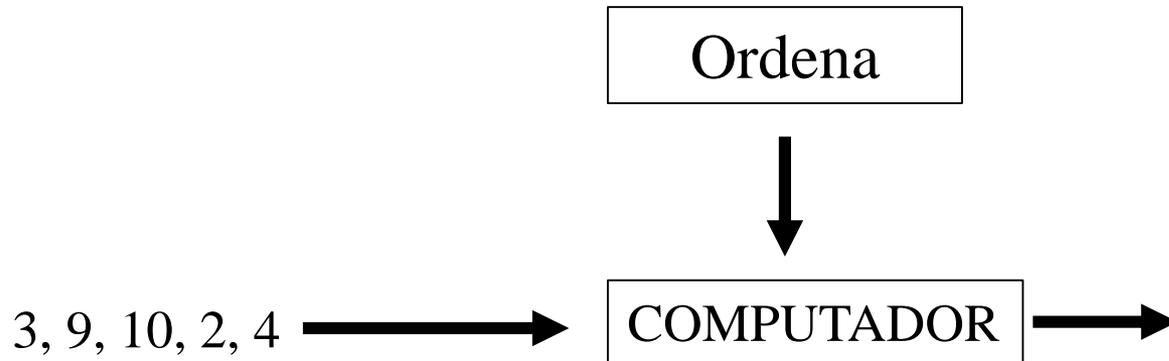


■ Modelo de propósito único



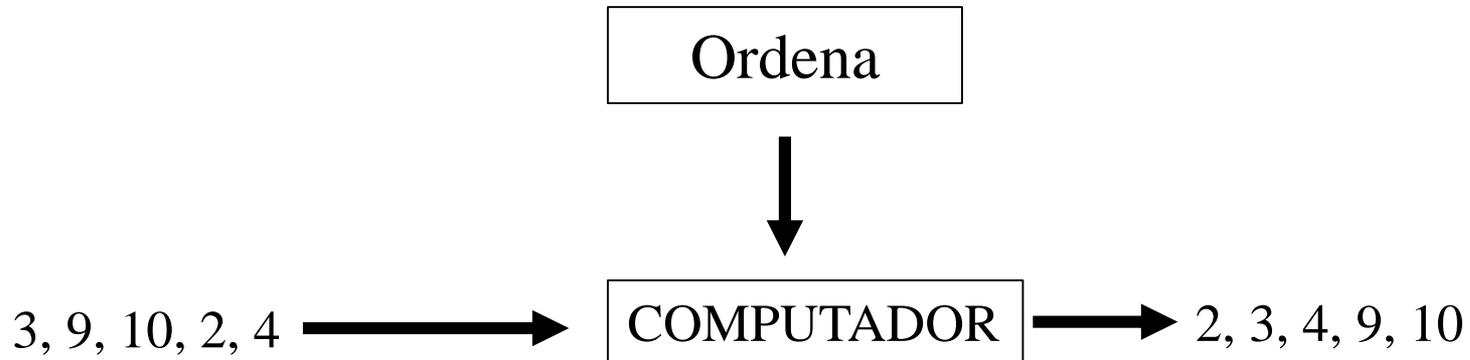
Modelo de Turing - Exemplos

- Mesmo programa e dados de entrada diferentes



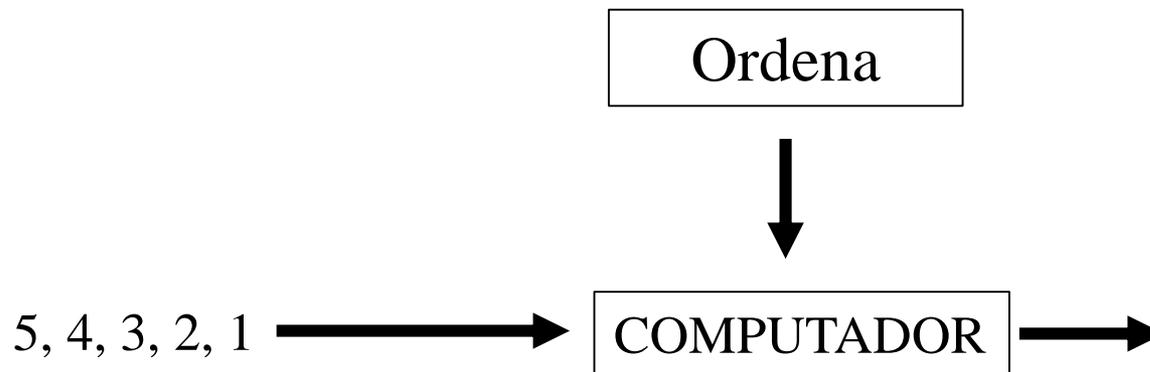
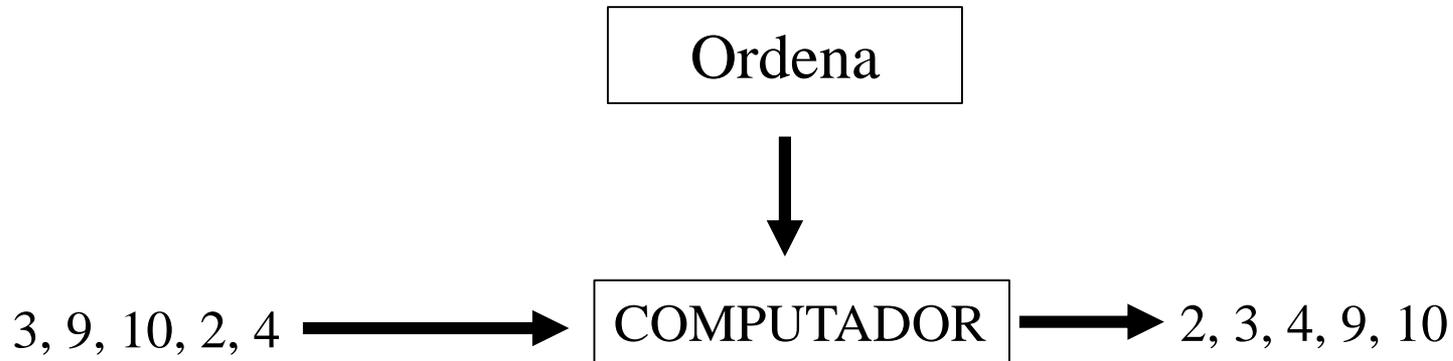
Modelo de Turing - Exemplos

- Mesmo programa e dados de entrada diferentes



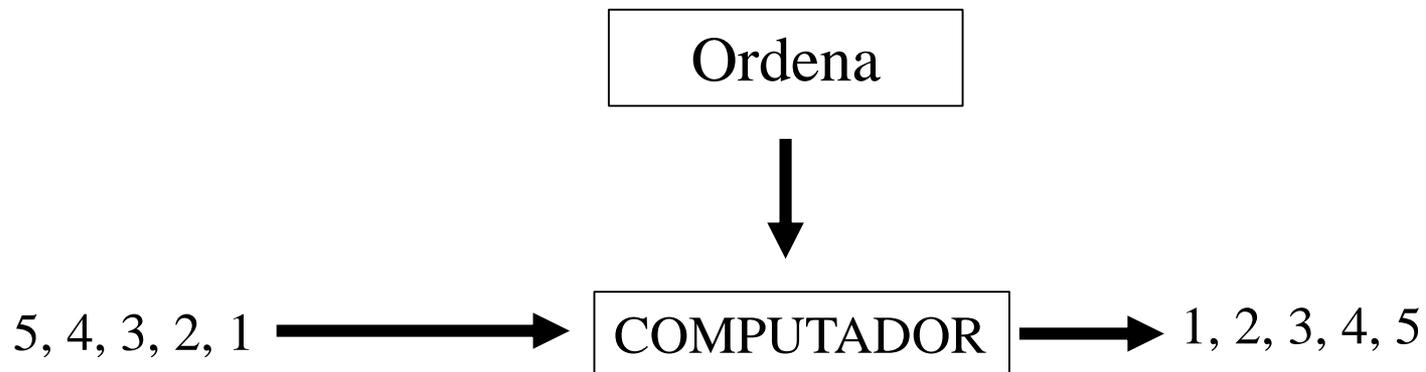
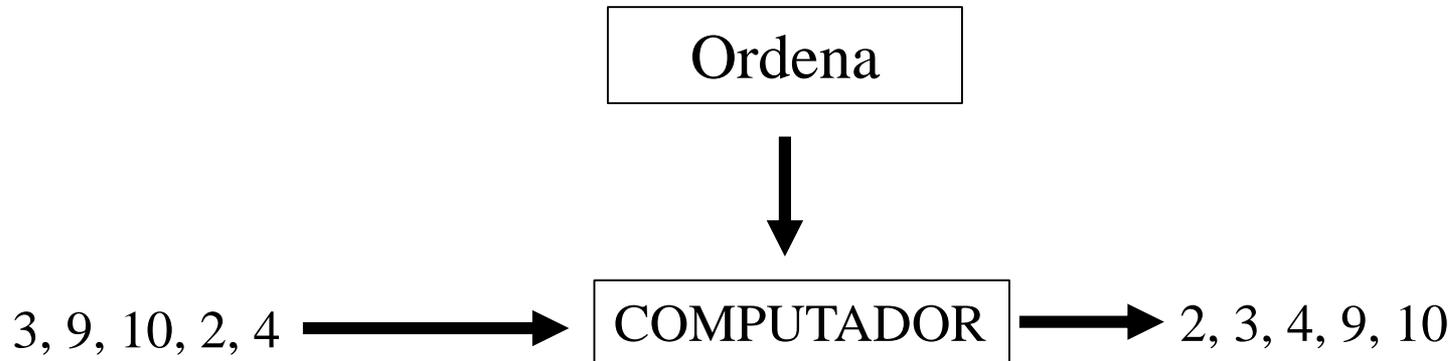
Modelo de Turing - Exemplos

- Mesmo programa e dados de entrada diferentes



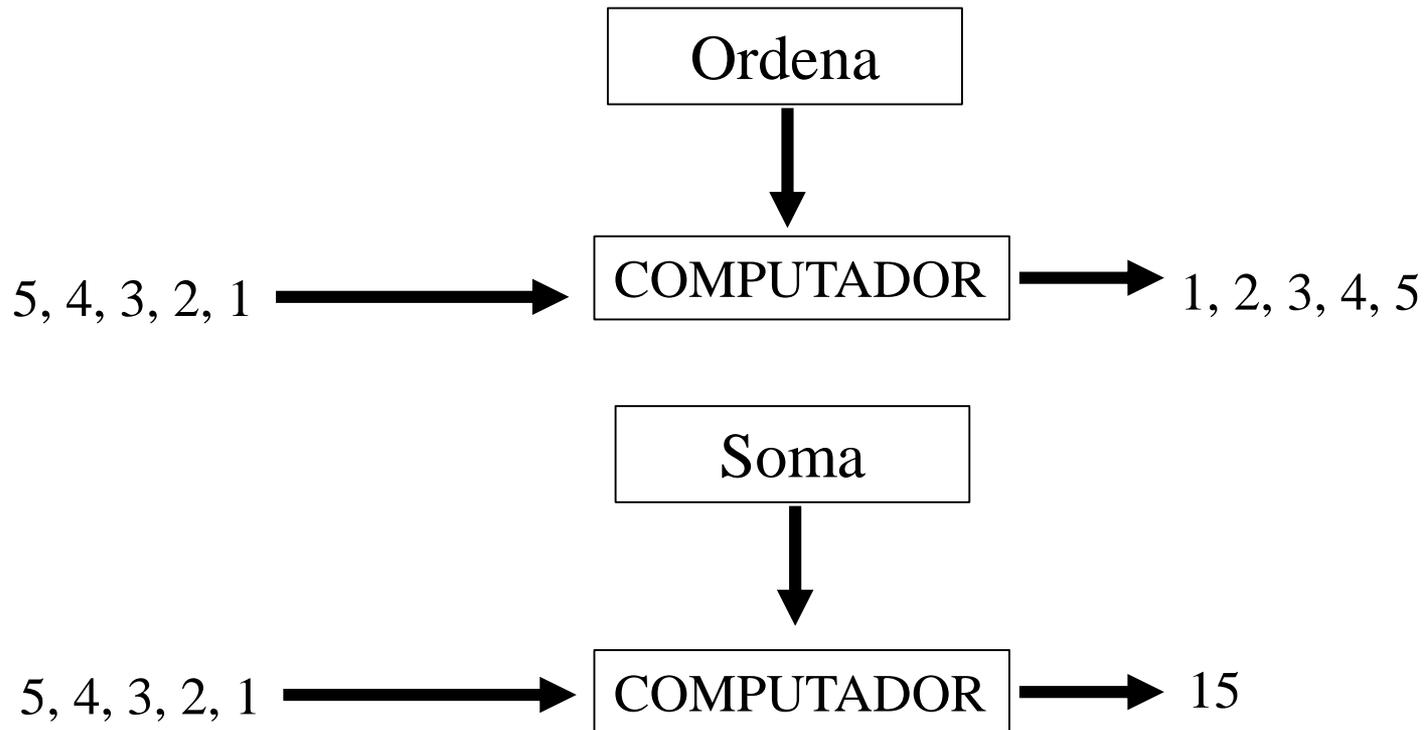
Modelo de Turing - Exemplos

- Mesmo programa e dados de entrada diferentes



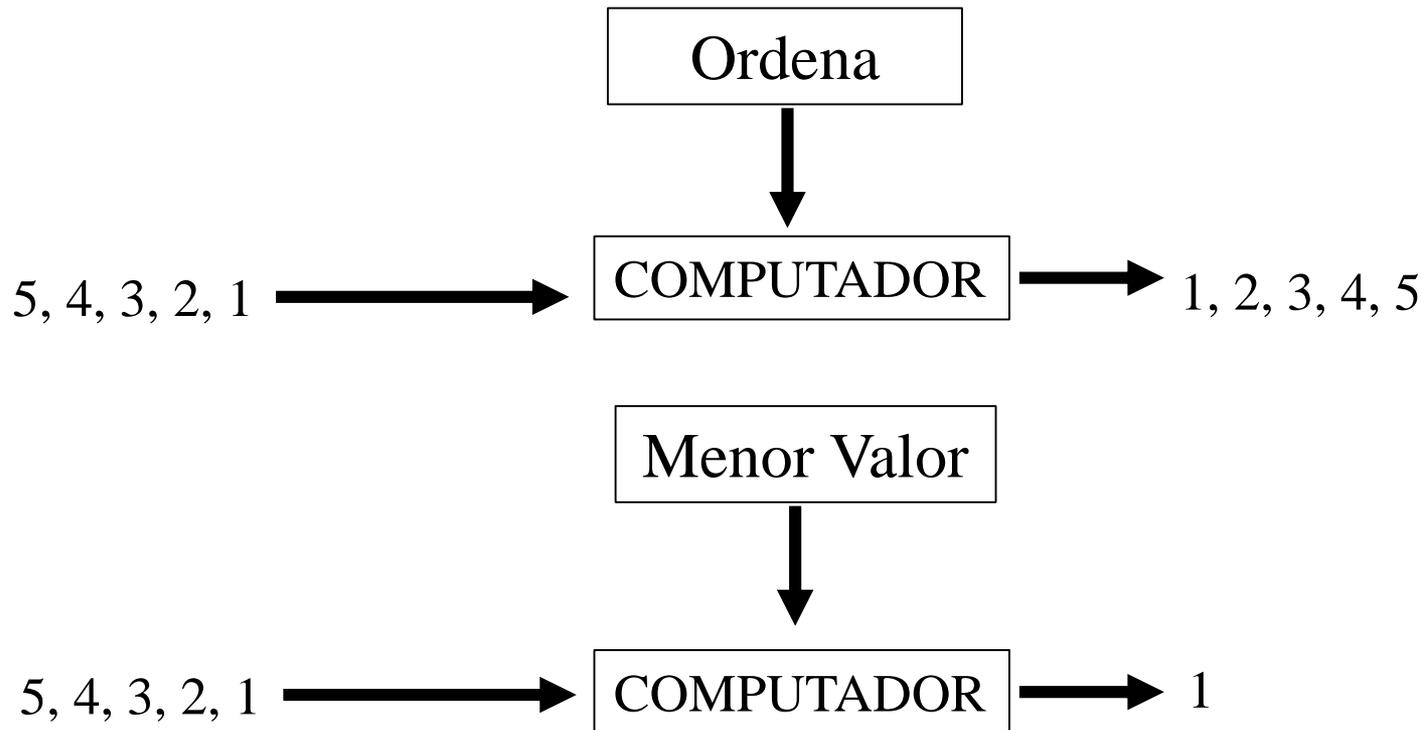
Modelo de Turing - Exemplos

- Programas diferentes e mesma entrada



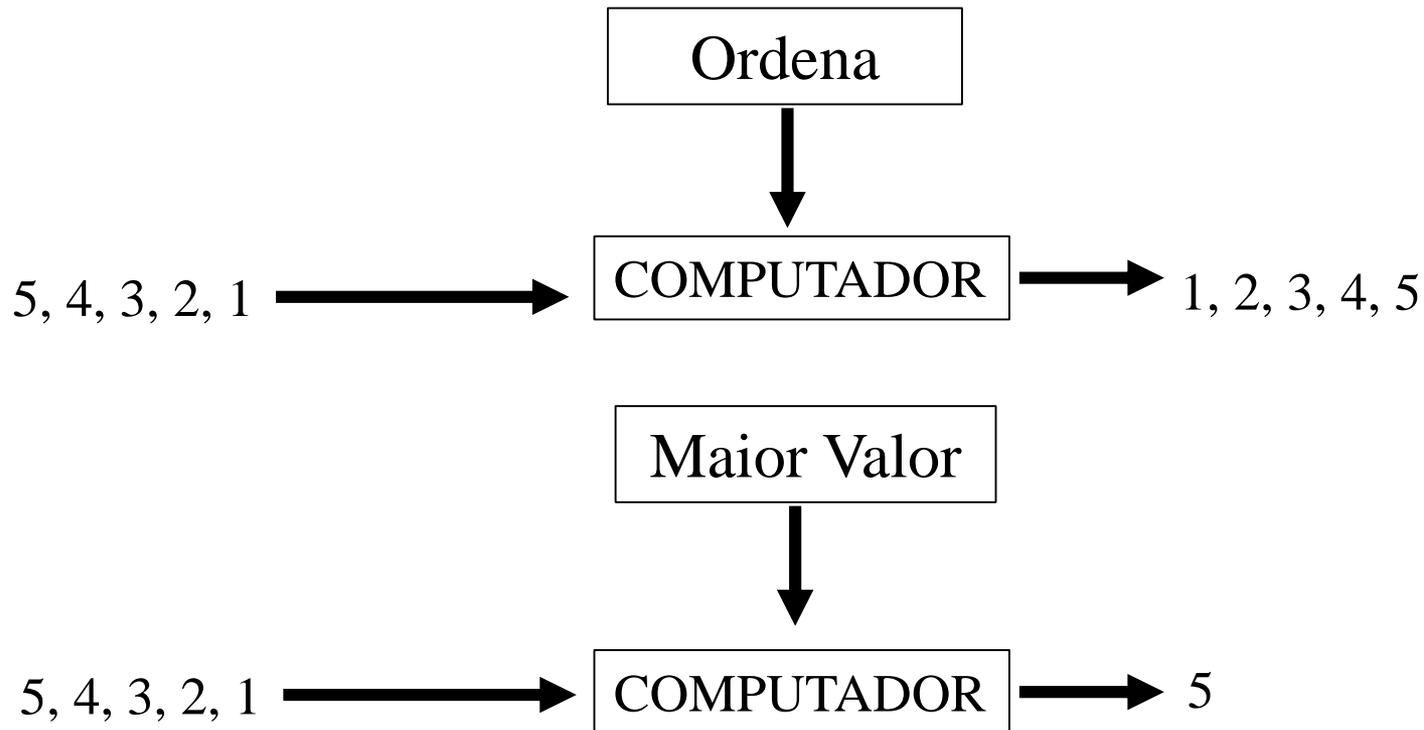
Modelo de Turing - Exemplos

- Programas diferentes e mesma entrada



Modelo de Turing - Exemplos

- Programas diferentes e mesma entrada



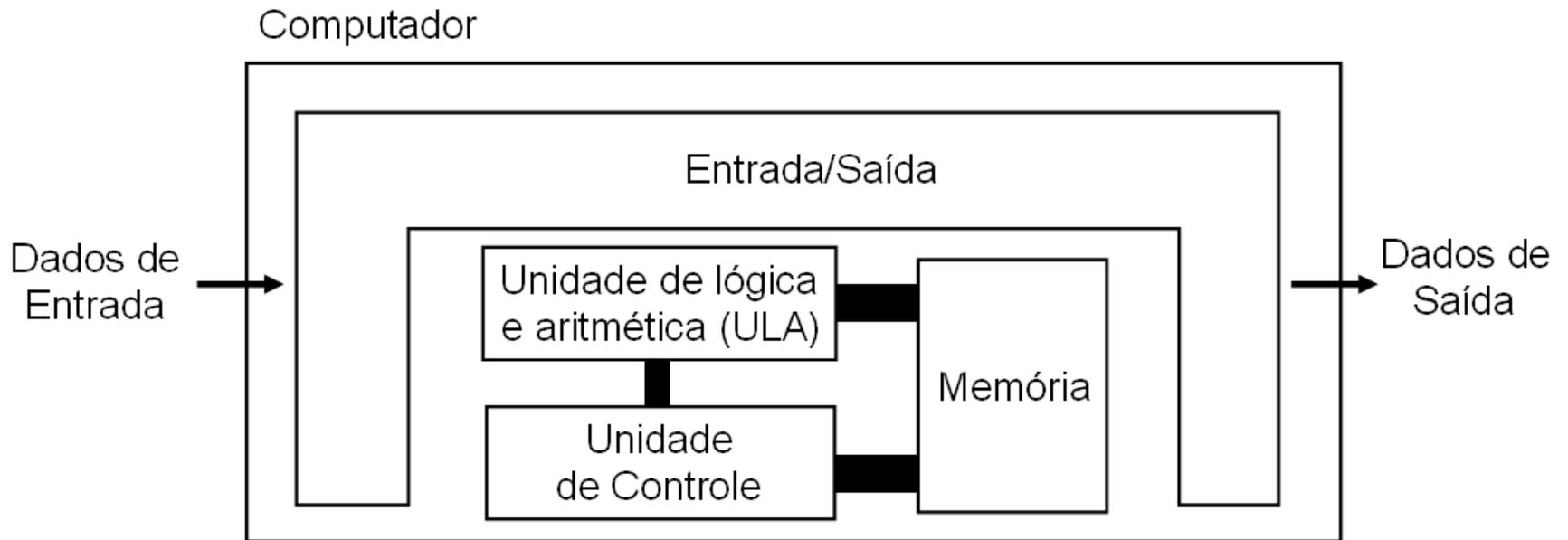
Modelo de Von Neumann

- No modelo da Máquina Universal de Turing os dados são armazenados na memória e o programa fora dela
- Por volta de 1944 – 1945, John von Neumann propôs que os programas também devem ser armazenados na memória de um computador
 - Visto que o programa e os dados são logicamente os mesmos

Modelo de Von Neumann

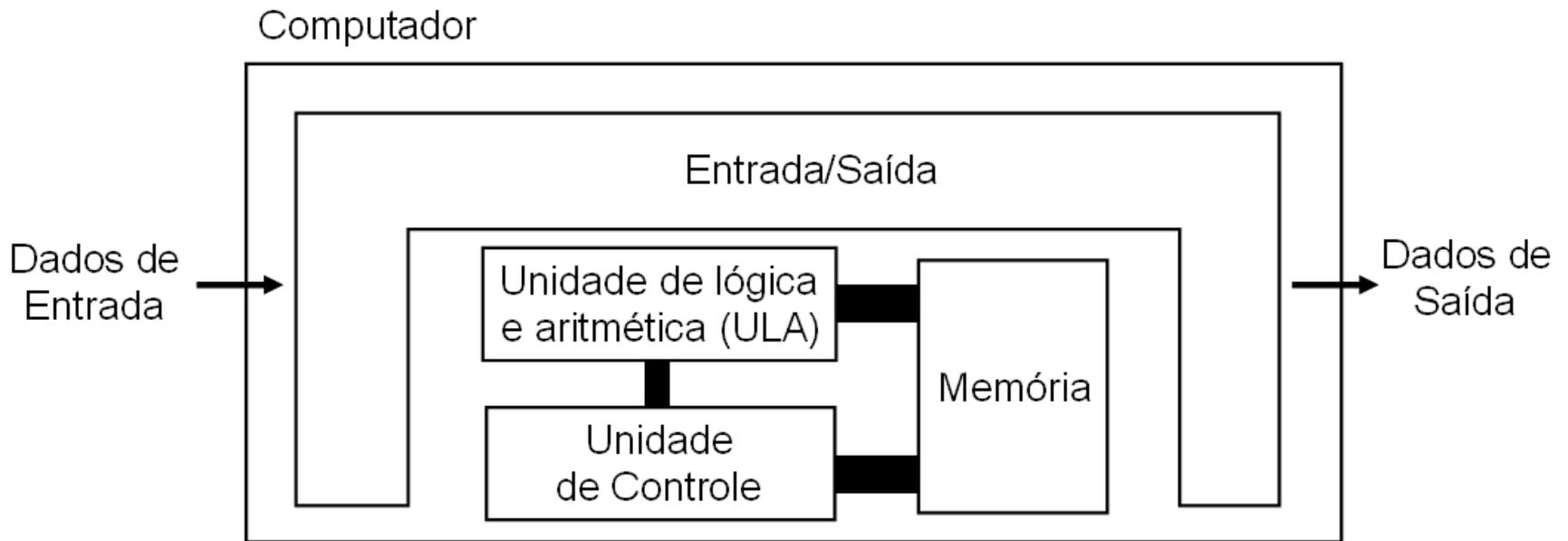
- Computadores construídos com base no Modelo de von Neumann dividem o hardware do computador em quatro subsistemas
 - Memória
 - Unidade lógica e aritmética
 - Unidade de controle
 - Entrada/Saída

Modelo de Von Neumann



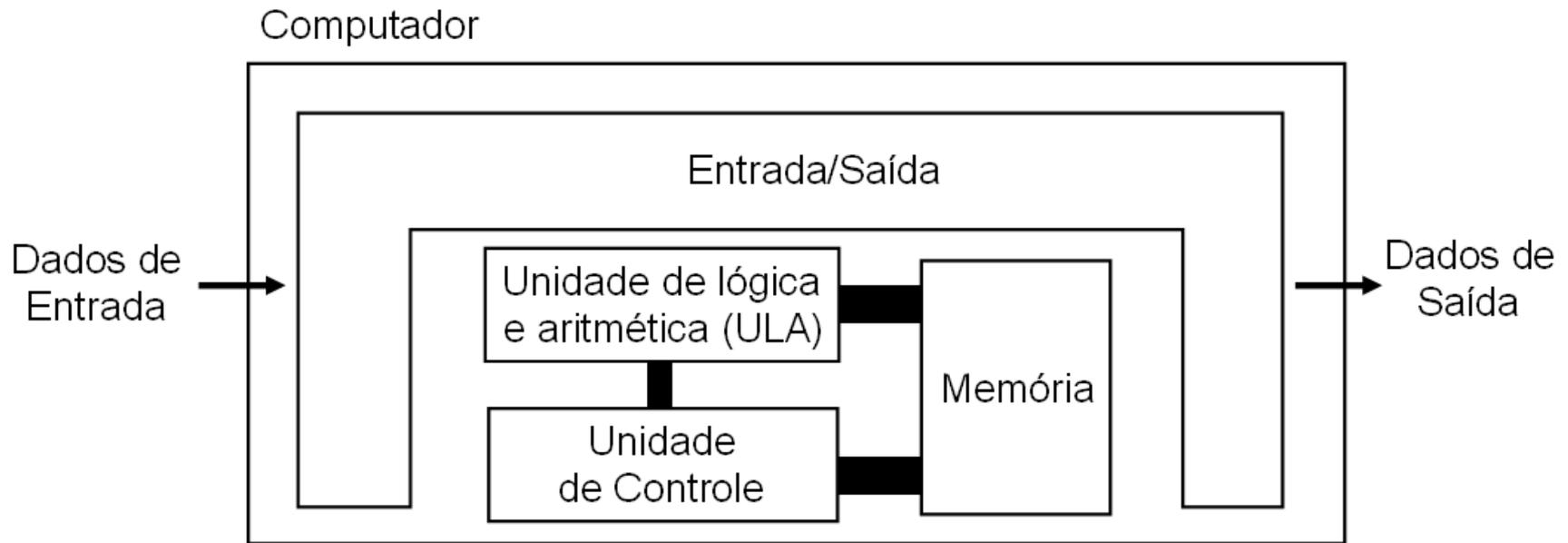
Modelo de Von Neumann

- Memória: área de armazenamento
 - onde programas e dados são armazenados



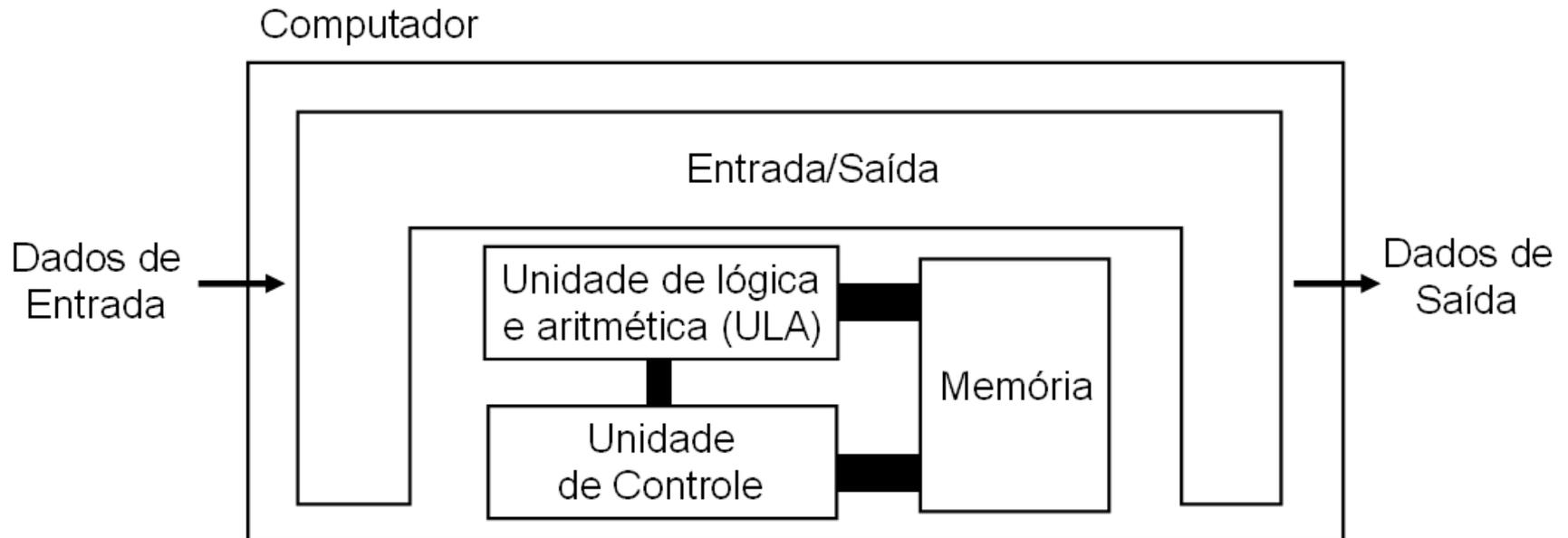
Modelo de Von Neumann

- Unidade de lógica e aritmética (ULA)
 - Onde ocorrem as operações de lógica e de cálculos sobre os dados



Modelo de Von Neumann

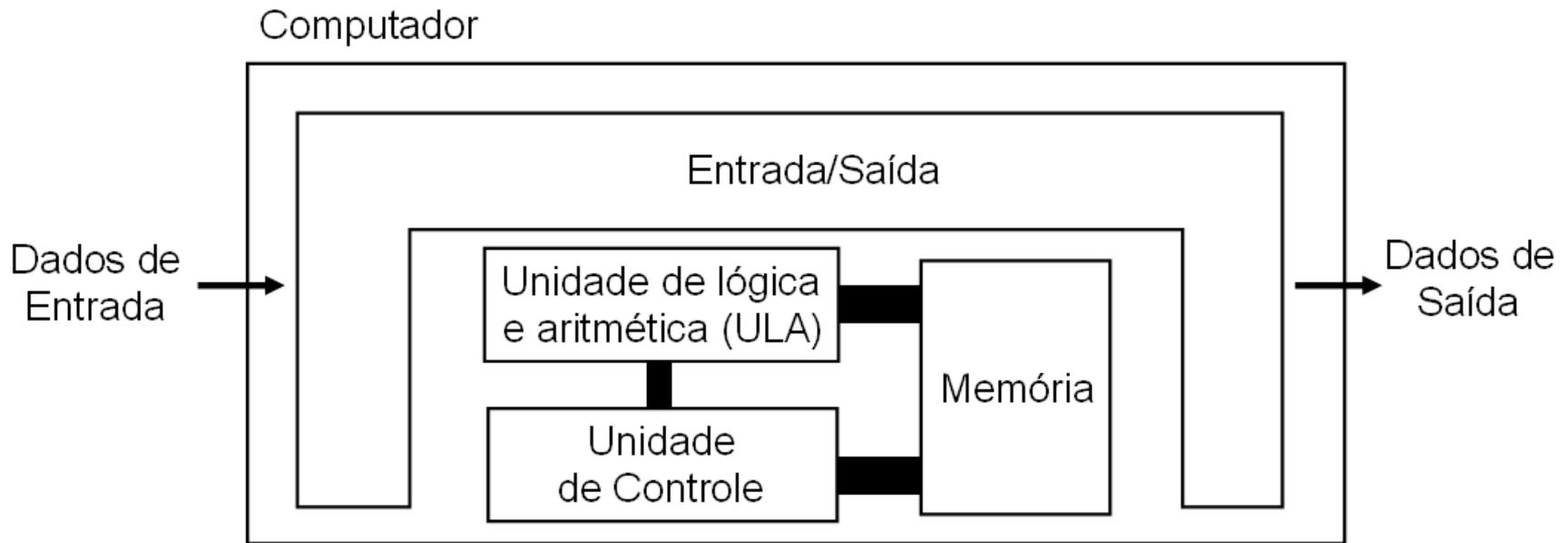
- Unidade de controle
 - Unidade que controla as operações de memória, a ULA e o subsistema de entrada/saída



Modelo de Von Neumann

■ Subsistemas de Entrada e de Saída

- Entrada de dados e programas de fora do computador
- Envia o resultado do processamento
 - Monitor, impressora, discos, fitas



Modelo de Von Neumann

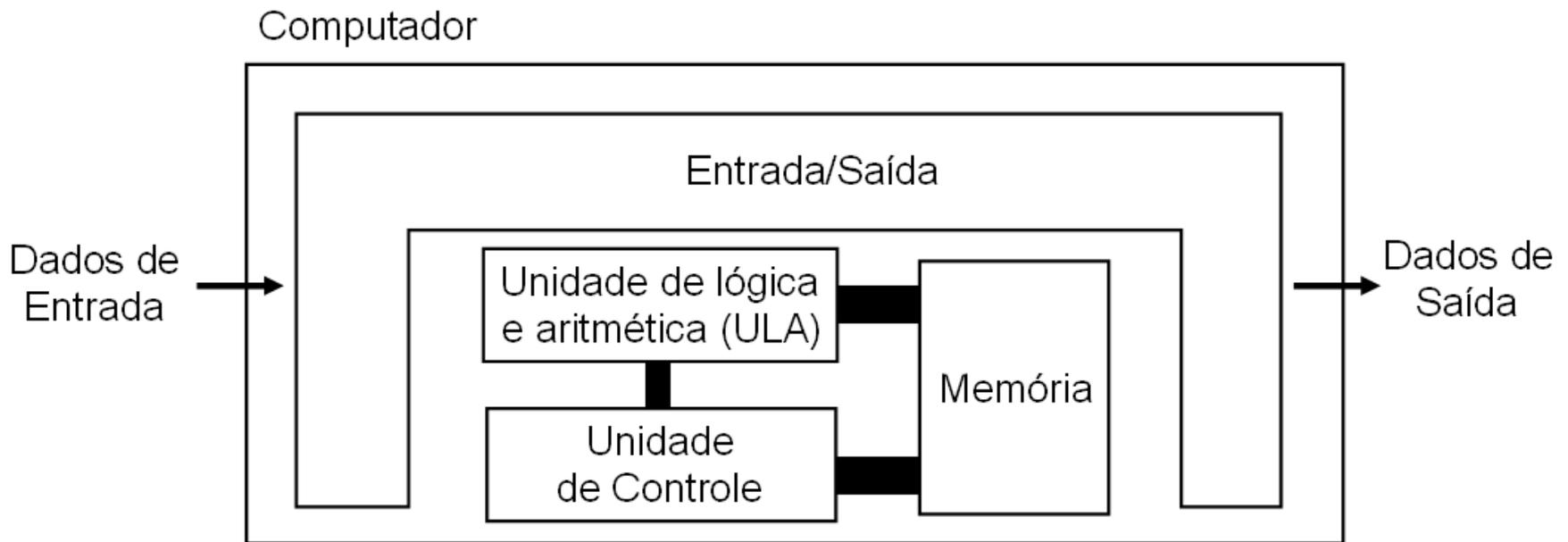
■ Programa Armazenado

- A memória dos computadores modernos mantém o programa e seus dados
 - Ambos devem ter o mesmo formato
- Totalmente diferente da arquitetura dos primeiros computadores
 - Somente os dados eram armazenados na memória
 - Programas eram implementados pela manipulação de um conjunto de comutadores ou modificação de sistema de fios

Modelo de Von Neumann

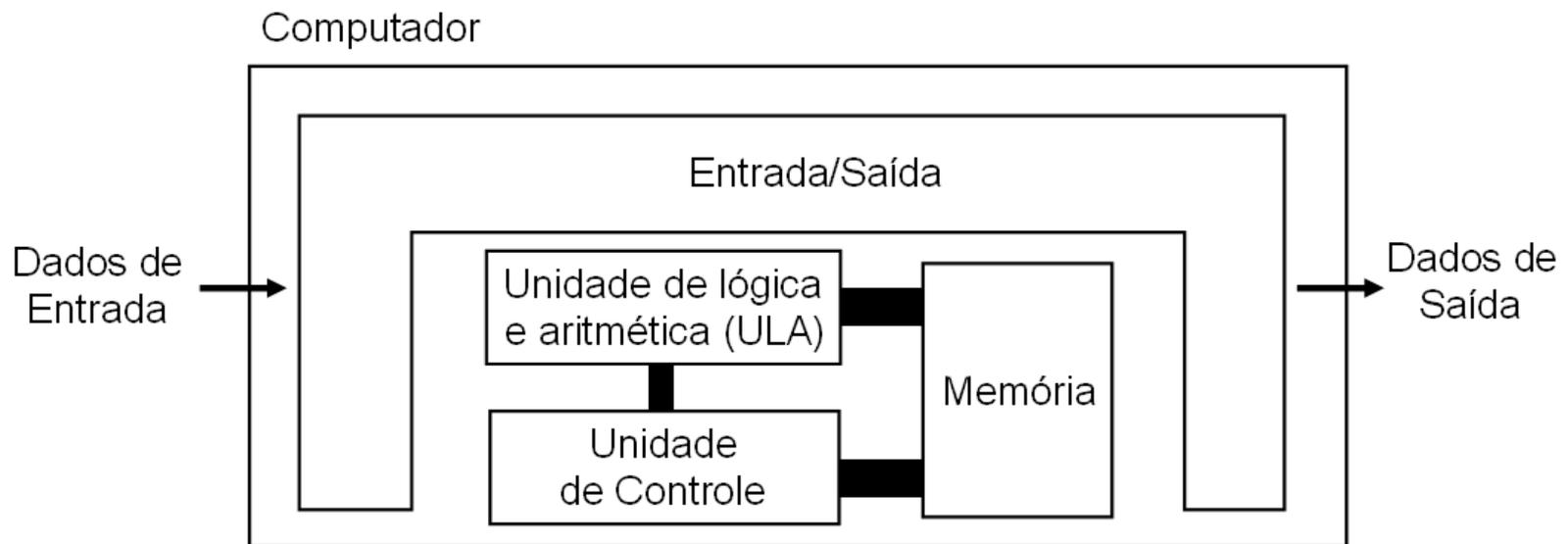
■ Execução sequencial de instruções

- O programa é composto por um número finito de instruções
- A unidade de controle busca uma instrução na memória, decodifica e então a executa
 - Uma após a outra



Componentes Computacionais

- Podemos pensar em um computador como sendo formado por três componentes
 - Hardware
 - Dados
 - Software



Hardware

- Na atualidade, o hardware tem quatro componentes, de acordo com o Modelo de von Neumann
 - Entrada/Saída
 - Unidade de Controle
 - Memória
 - Unidade de Lógica e Aritmética (ULA)

Dados

- O Modelo de von Neumann define claramente um computador como uma máquina de processamento de dados
 - Aceita dados de entrada, processa-os e fornece os resultados
- Esses dados devem ser armazenados
 - Isso não é definido pelo modelo

Armazenamento de Dados

- O computador é um dispositivo eletrônico
 - Melhor maneira de armazenar os dados é na forma de um **sinal elétrico**
 - Presença ou ausência dele
- Os dados que utilizamos na vida diária **não** são encontrados em apenas dois estados
 - Sistema de numeração (0..9)
 - Texto, som, imagem, etc.

Organização dos Dados

- Os dados externos a um computador podem assumir muitas formas
 - Mas devem estar todos em único padrão, quando estiverem armazenados no computador

Software de Computador

- Os modelos de Turing e von Neumann tem como principal característica o PROGRAMA (Software)
- Apesar dos primeiros computadores não armazenarem o programa na memória, eles utilizavam o conceito de Programas

Software de Computador

- No Modelo de von Neumann os programas são armazenados na memória do computador
 - Juntamente com os dados



Software de Computador

- Programar os primeiros computadores significava modificar o sistema de fios
 - Ou ligar ou desligar um conjunto de comutadores
- A programação era uma tarefa realizada por um operador ou um engenheiro
 - Antes de efetivamente iniciar o processamento de dados

Software de Computador

- Sequência de Instruções
 - Cada uma opera em um ou mais itens de dados
 - Uma instrução pode modificar o efeito da anterior
- Facilita a reutilização
 - Um programa pode combinar diferentes instruções para realizar qualquer número de programas
 - Um programa pode ser uma combinação de instruções diferentes

Software de Computador

■ Programador

- ❑ Precisa compreender a tarefa realizada por cada instrução
- ❑ Também precisa saber como combinar essas instruções para desempenhar determinada tarefa

Software de Computador

- O programador deve
 - Entender o problema
 - Resolver o problema etapa por etapa
 - Encontrar as instruções apropriadas
- Essa sequência de instruções é chamada de **ALGORITMO**

Software de Computador

- No início da era dos computadores, os algoritmos eram codificados em uma única linguagem: **a Linguagem de Máquina**
 - Instruções no padrão binário eram escritas para resolver um problema
 - Escrever programas longos era uma tarefa tediosa

Software de Computador

- Futuramente, os cientistas da computação apresentaram a ideia de utilizar símbolos para representar padrões binários
- Surgiu o conceito de **linguagem de computador** ou **linguagem de programação**
 - Essa linguagem tem um número limitado de símbolos e de palavras

Software de Computador

- Linguagem Assembly

```
mov ax,0
mov ax,cx
out 70,al
mov ax,0
out 71,al
inc cx
cmp cx,100
jb 103
int 20
```

Software de Computador

■ Linguagem C

```
#include <conio.h>
#include <stdio.h>
main()
{
    int i, num_anterior, num_atual, novo_numero;
    num_anterior = 0;
    num_atual = 1;
    printf("o 1. numero fibonaccci e 1\n");
    for(i=2;i<=10;i++)
    {
        novo_numero = num_atual + num_anterior;
        num_anterior = num_atual;
        num_atual = novo_numero;
        printf("o %d numero fibonaccci e %d\n",i,novo_numero);
    }

    printf("\n\n\n.....FIM.....");
    getch();
}
```

Software de Computador

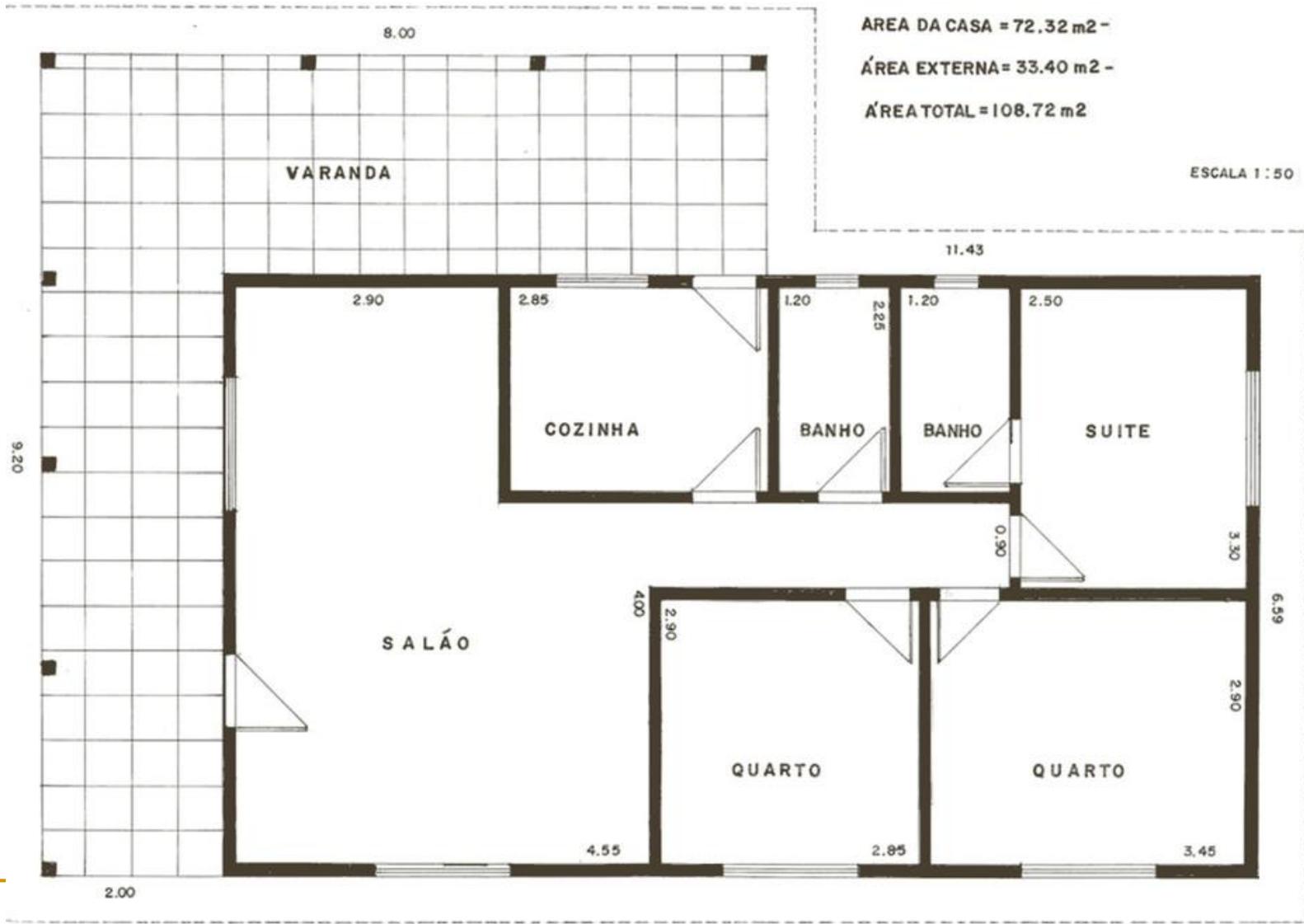
■ Linguagem Pascal

```
Program A_Media_Maor_q_60;
var
  n1,n2,n3,n4,md:real;

Begin
  writeln('Entra com a 1 Nota');
  readln(n1);
  writeln('Entra com a 2 Nota');
  readln(n2);
  writeln('Entra com a 3 Nota');
  readln(n3);
  writeln('Entra com a 4 Nota');
  readln(n4);
  md:=(n1+n2+n3+n4)/4 ;
  if md>60 then
  begin
    writeln('A Media é Maior',md);
  end;
  if md<60 then
  begin
    writeln('A Media é Menor',md);
  end;
end;
```

Software de Computador

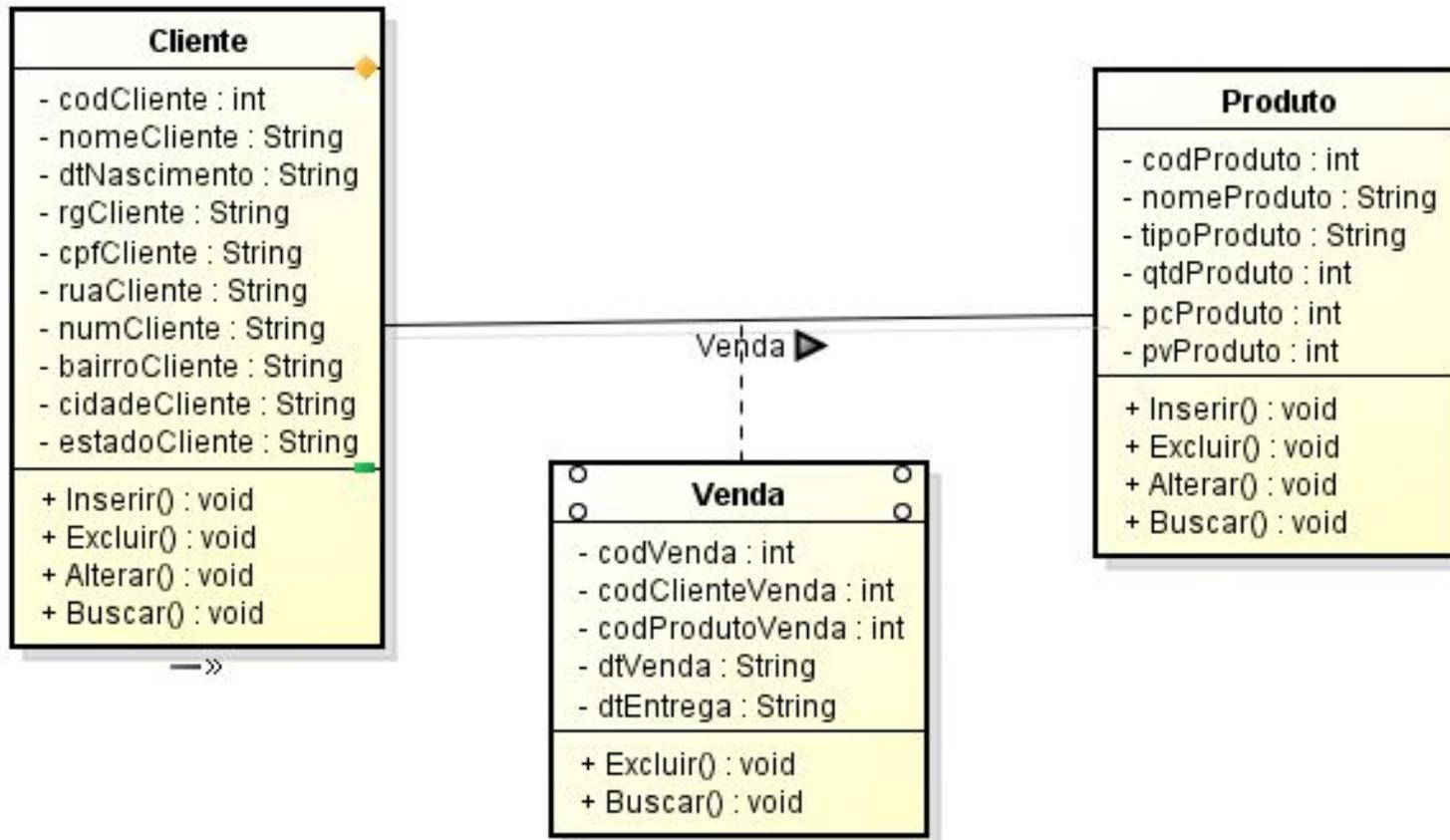
- Atualmente, escrever um programa requer seguir regras e princípios estritos
- O projeto e escrita de programas é foco de estudo da **Engenharia de Software**



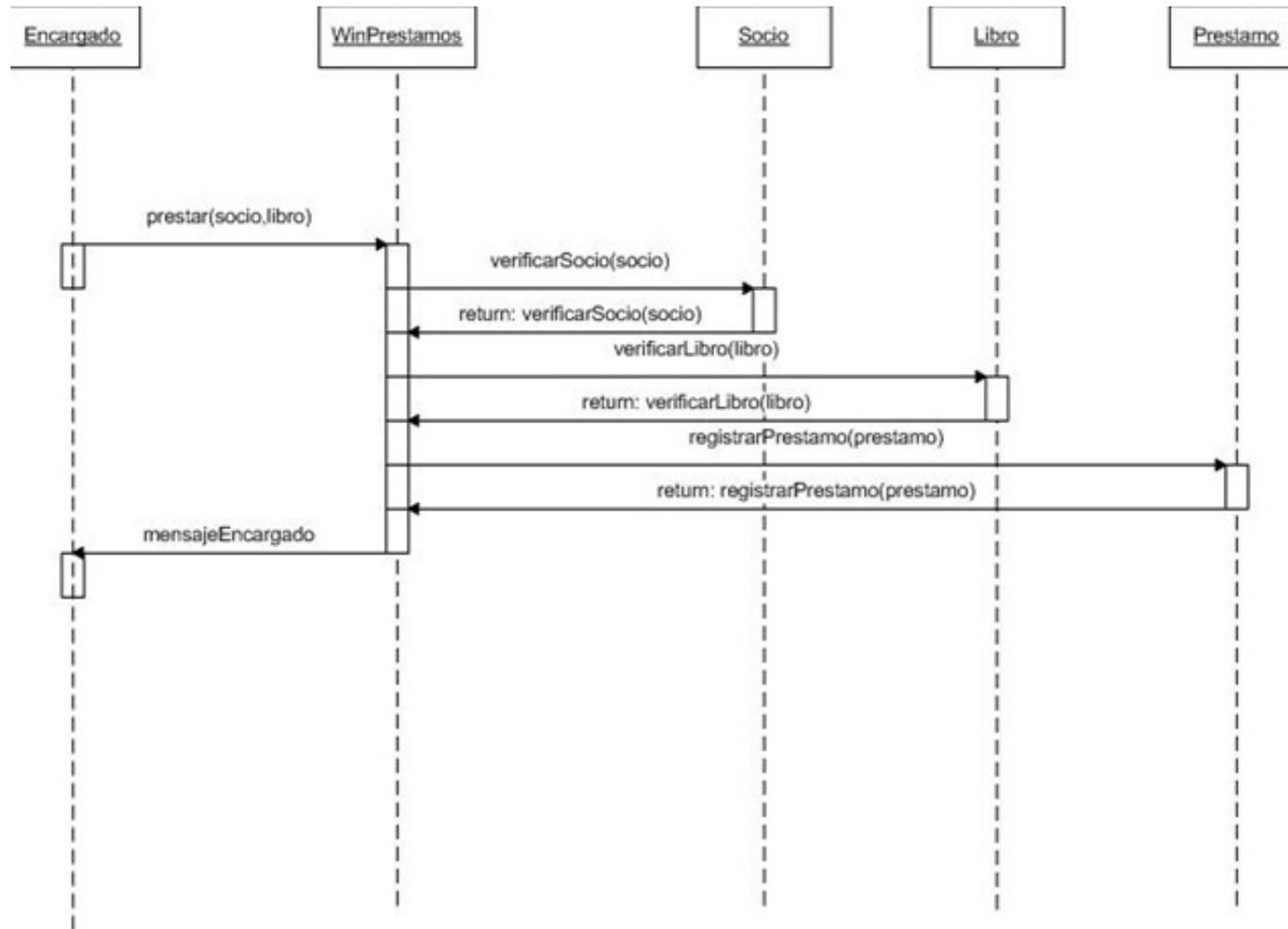




Engenharia de Software



Engenharia de Software



Software de Computador

- Durante a evolução dos computadores, os cientistas notaram que havia uma série de instruções comuns a todos os programas
 - Ex.: entrada e saída
- É mais eficiente escrevê-las somente uma vez para todos os programas

Software de Computador

- Surgiu o conceito de **Sistema Operacional**
- Inicialmente trabalhava como um gerenciador
 - Ele facilitava o acesso aos componentes do computador
- Atualmente ele faz muito mais
 - Ex.: gerencia os programas em execução

Referências Bibliográficas

- Fundamentos da Ciência da Computação
 - Behrouz Forouzan e Firouz Mosharraf
- Ciência da Computação: uma visão abrangente
 - J. Gleen Brookshear

Introdução à Ciência da Computação: aula introdutória

Prof. Dr. Danilo Medeiros Eler
danilo.eler@unesp.br