

# Algoritmos e Estruturas de Dados I

---

Apresentação

**Prof. Lucas Astore**

**Prof. Cristiano Rodrigues**

# Plano de Ensino

Algoritmos e Estruturas de Dados I

---

# Algoritmos e Estruturas de Dados I

## **Ementa**

- Representação e armazenamento de dados.
- Manipulação e movimentação de dados em memória principal e secundária.
- Abstração de dados.
- Estruturas e abstração de controle.
- Modularização, encapsulamento e herança.
- Recursividade.
- Documentação e testes.
- Implementação em linguagem de programação.
- Contagem de operações.

---

# Algoritmos e Estruturas de Dados I

---


# Algoritmos e Estruturas de Dados I



Sequência finita de instruções  
que levam à solução de um  
problema em um número finito de  
passos.

---

# Algoritmos e Estruturas de Dados I



Sequência finita de instruções que levam à solução de um problema em um número finito de passos.

Podemos dizer que é a forma como os dados são agregados ou organizados. Uma mesma estrutura de dados pode ser eficiente em algumas operações e ineficiente em outras.

Exemplos: arrays, listas, pilhas, filas, dicionário, grafos...

---

# Algoritmos e Estruturas de Dados I

## Objetivos

- Desenvolver habilidades em expressar soluções algorítmicas e mediante uso de estruturas de dados.
- Desenvolver habilidades em discernir e escolher representações de informações.
- Desenvolver habilidades em operar diferentes representações de dados.
- Exercitar o emprego de técnicas elementares para a construção e a documentação de programas.
- Introduzir o uso de uma linguagem de programação.
- Desenvolver a expressão de soluções algorítmicas em uma linguagem de programação.

# Metodologia

Algoritmos e Estruturas de Dados I



---

# Programa e Formato

- Cronograma
- Aulas teóricas – sala de aula
  - 2as e 4as feiras, das 8h50 às 10h30
- Aulas práticas - laboratório
  - G1 - 4as feiras

# Avaliações

1a Prova - 25 pontos – 02/04/2025  
2a Prova - 25 pontos – 28/05/2025  
3a Prova - 25 pontos – 16/06/2025

Trabalhos e listas de exercícios - 10 pontos

Trabalho interdisciplinar (LIEC) – 5 pontos

Maratona de Programação - 5 pontos

Avaliação de Desempenho Acadêmico (ADA) - 5 pontos

Reavaliação: O aluno que não atingir 60 pontos durante o semestre deverá submeter-se à reavaliação no valor de 25 pontos, que substituirá a menor nota das provas.

# História dos Computadores

Do Ábaco (2000aC) à máquina analítica de Charles Babbage (1831)

# Geração Zero - Mecanismo

---

- Ábaco (2000aC) - Soma, subtração, multiplicação e divisão

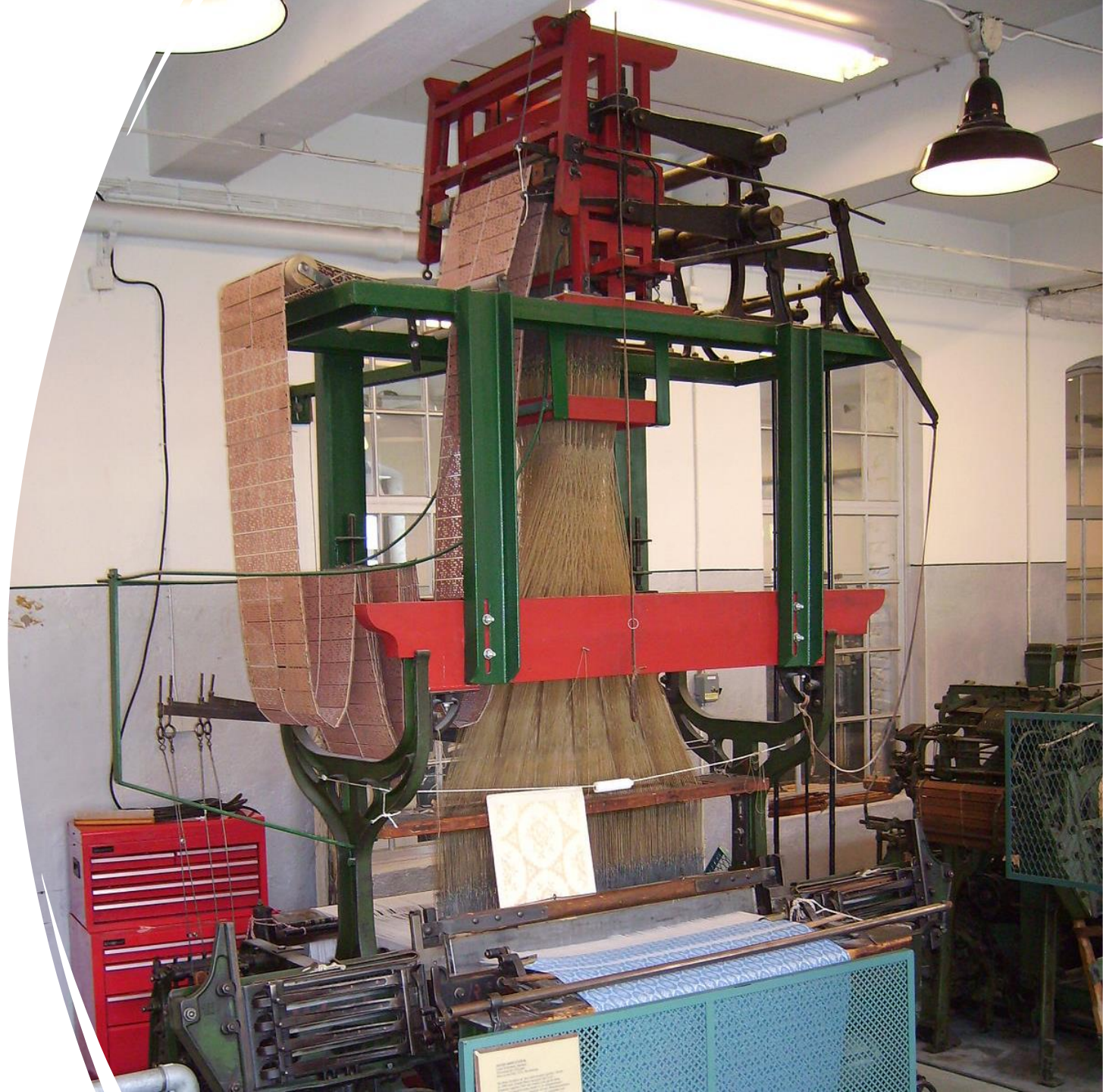


- 
- Pascaline (1642) – Soma e subtração com 8 algarismos



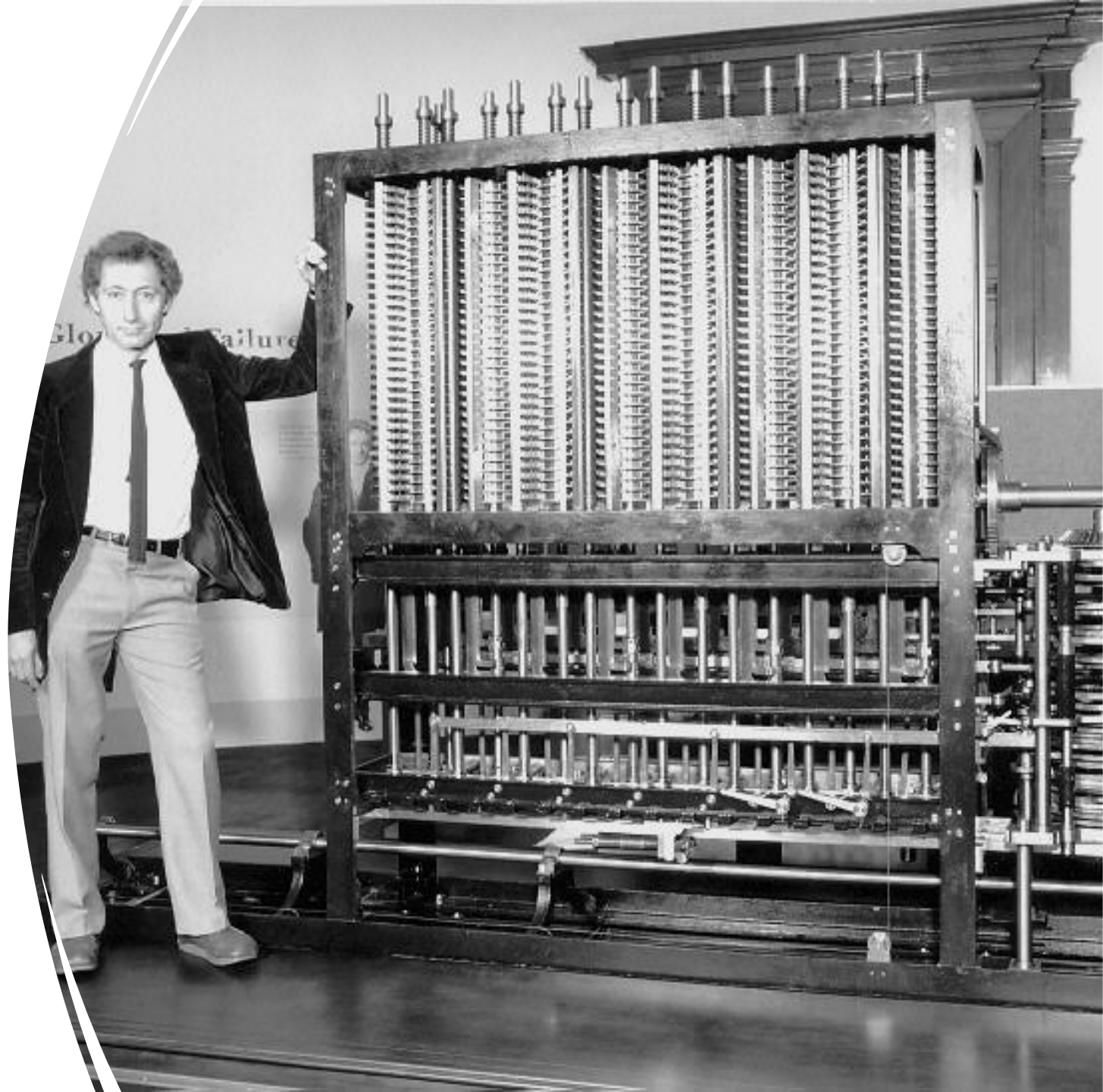


- 
- Tear de Jacquard (1801) – máquina programável para padrões de tecelagem
  - Padrões em cartões perfurados





- 
- Charles Babbage (1831) –  
Calculador Analítico:  
máquina analítica com quatro  
componentes – memória,  
processador, entrada e saída



# Primeira Geração

---

- 1940 - 1956
- Válvulas termiônicas
- Grandes dimensões
- Alto consumo de energia
- Dados inseridos por cartões perfurados

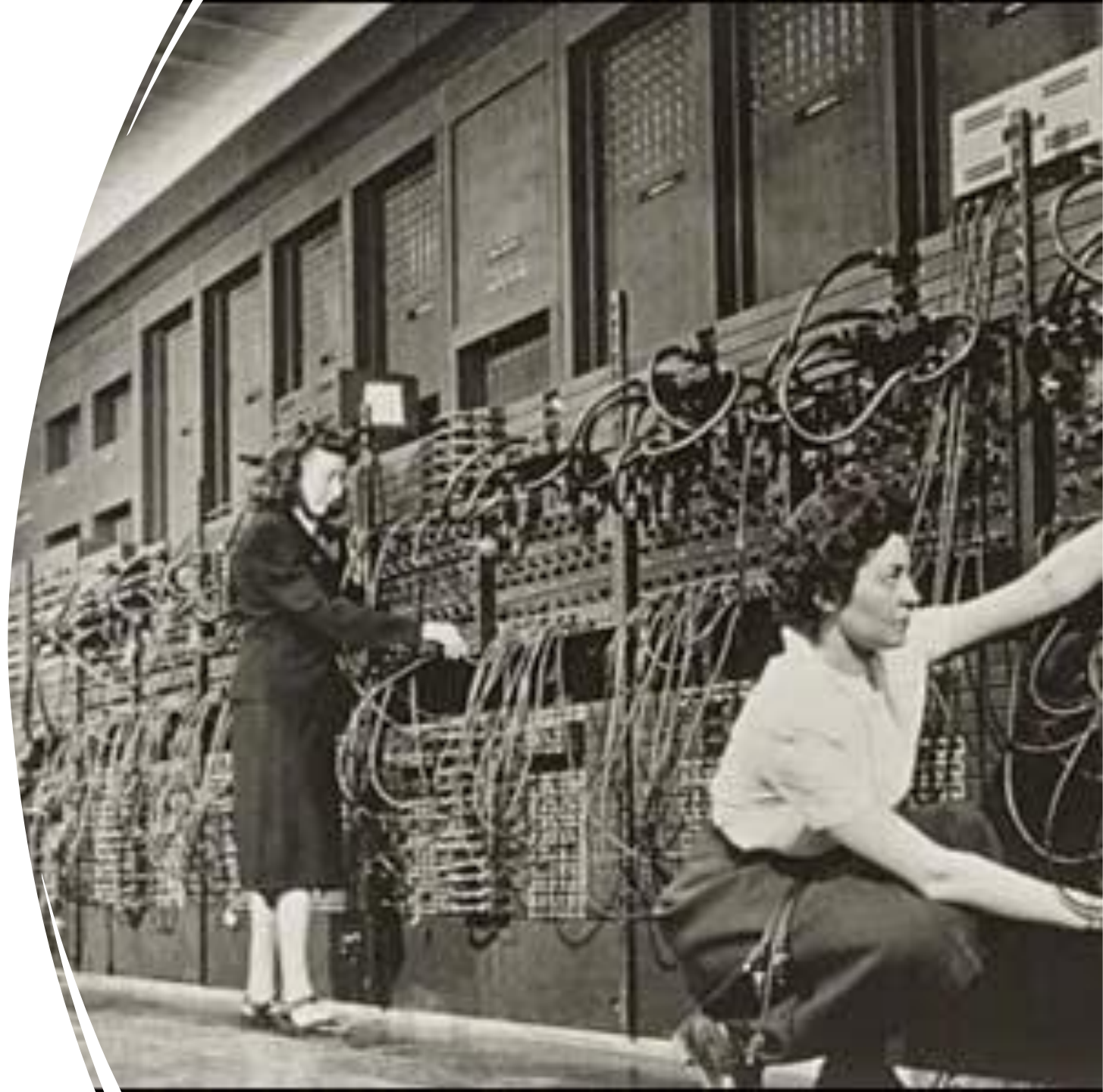




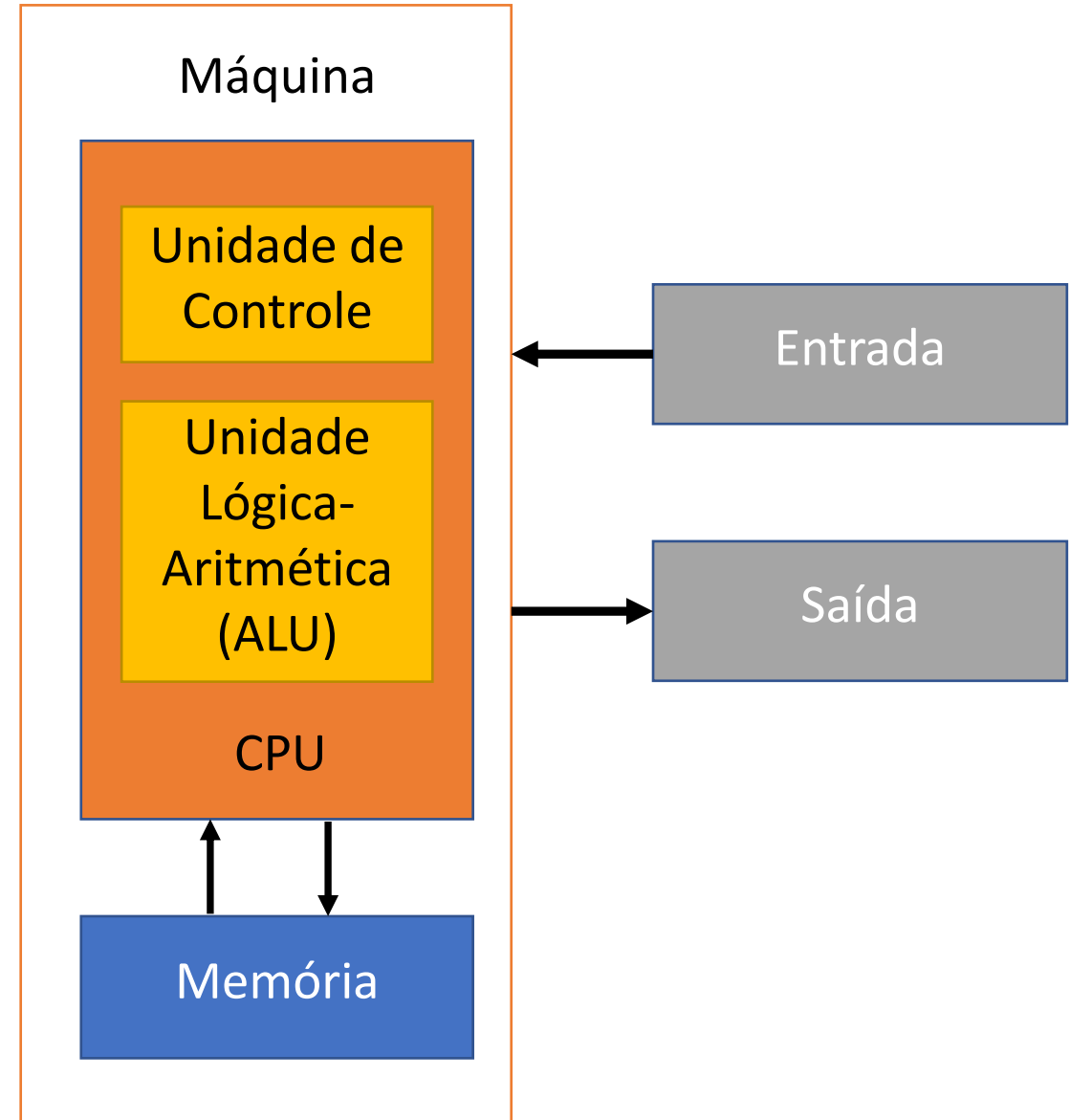
- 
- COLOSSUS (1943 – 1945)
  - Considerado o primeiro computador programável
  - Eletromecânico - programação feita por interruptores e plugues (e não um programa armazenado)



- 
- ENIAC (1946)
  - Criado para cálculos balísticos (II Guerra Mundial)
  - Dados armazenados em cartões perfurados

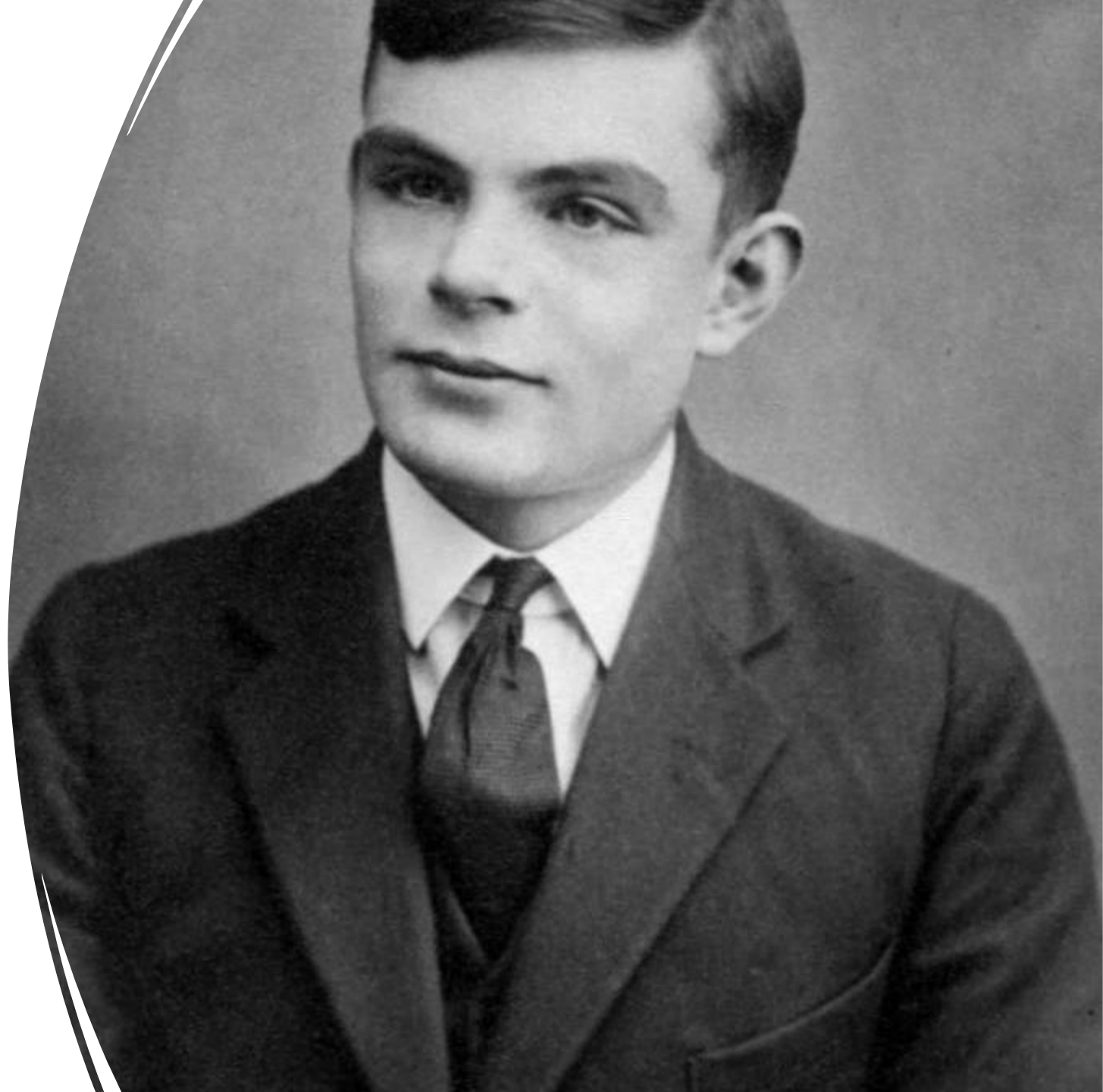


- 
- ISA (1946) – John von Neumann
  - ISA - Arquitetura de Conjunto de Instruções (do inglês, *Instruction Set Architecture*)





- 
- Alan Turing: “Poderia uma máquina pensar?” (1950)



# Segunda Geração

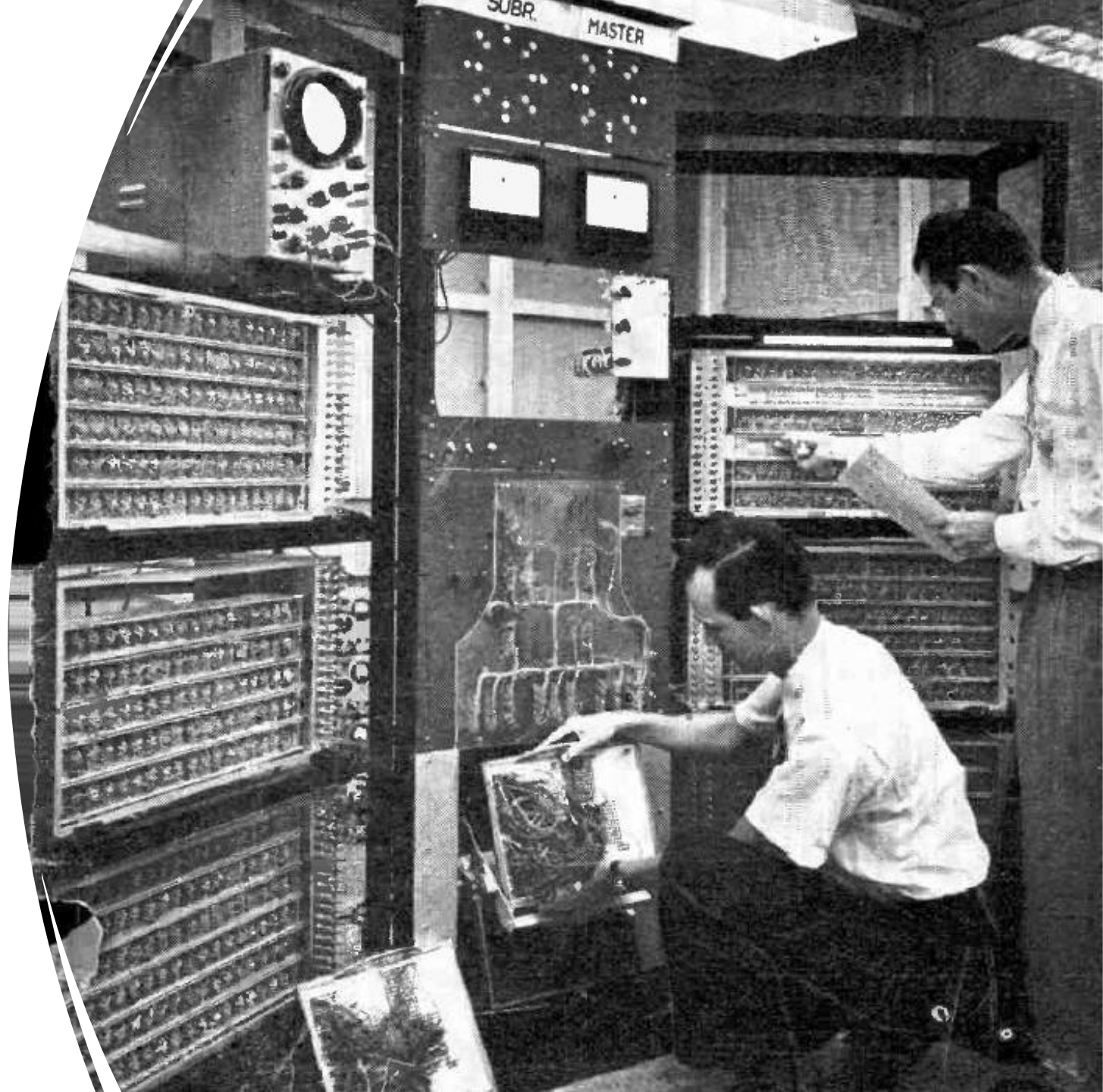
---

- Diodos e transistores (1955 a 1964)
- Menores dimensões
- Peso de até 750 kg





- 
- TRADIC (1955)
  - Primeiro computador transistorizado, com 800 transistores



- 
- PDP-1 (1959)
  - Percurssor dos minicomputadores
  - Precisava de apenas um operador
  - Preço acessível (U\$120K)
  - Monitor CRT



- 
- CDC 6600 (1964)
  - Primeiro computador com processamento paralelo

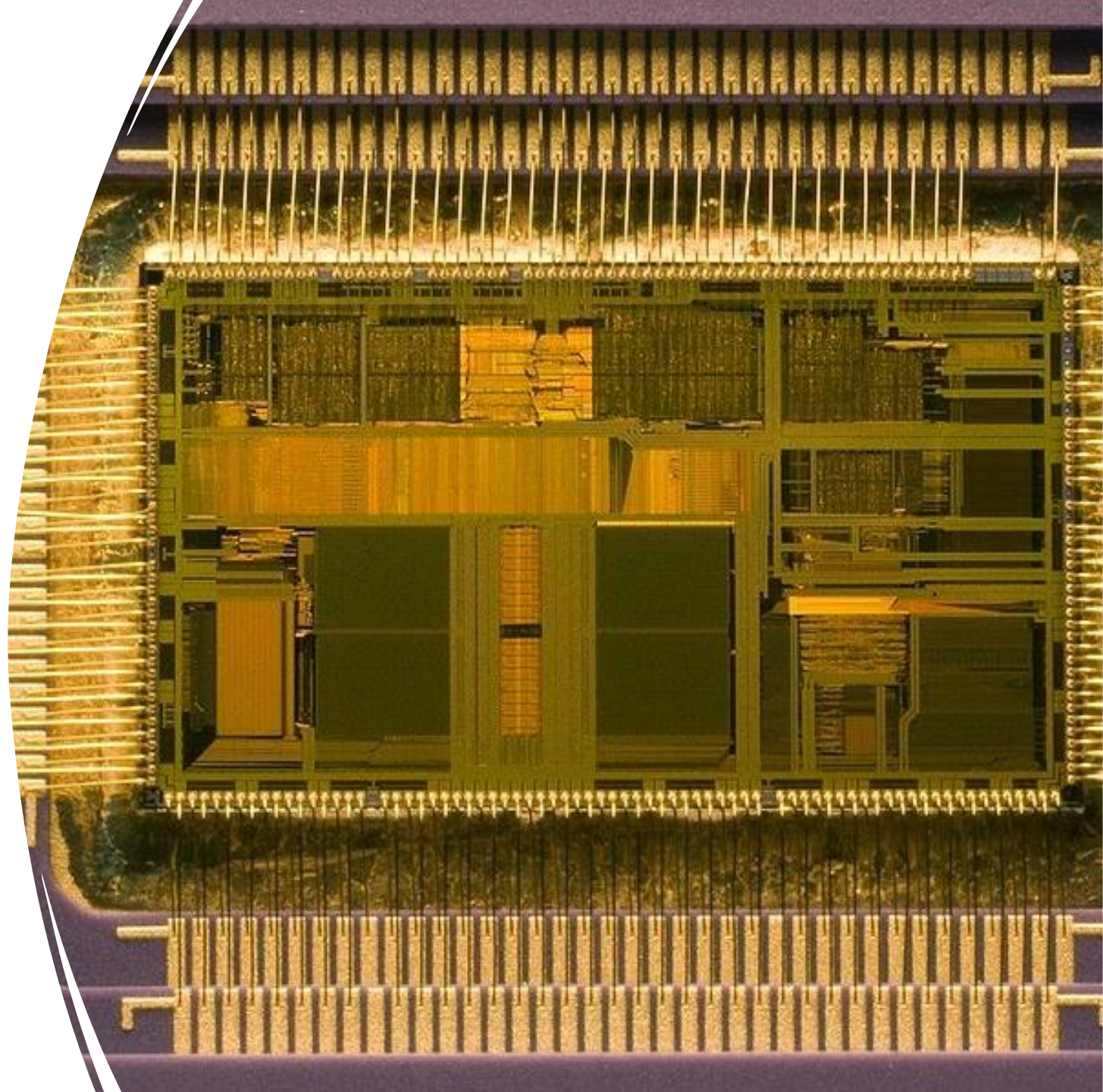




# Terceira Geração

---

- Circuitos integrados (1965 a 1971)
- Criação dos sistemas operacionais
- Entrada de dados por dispositivos periféricos





- IBM System 360 (1964)
- Família de computadores utilizando CIs com variação da velocidade do clock entre os computadores



- 
- Primeiro mouse (1964) - Douglas Engelbart (1964)



# Quarta Geração

---

- Microprocessadores (1971 – presente)
- Maior Integração
- Armazenamento, memória RAM e ROM.
- Entrada de dados por dispositivos periféricos
- Miniaturização (A partir de 1980)

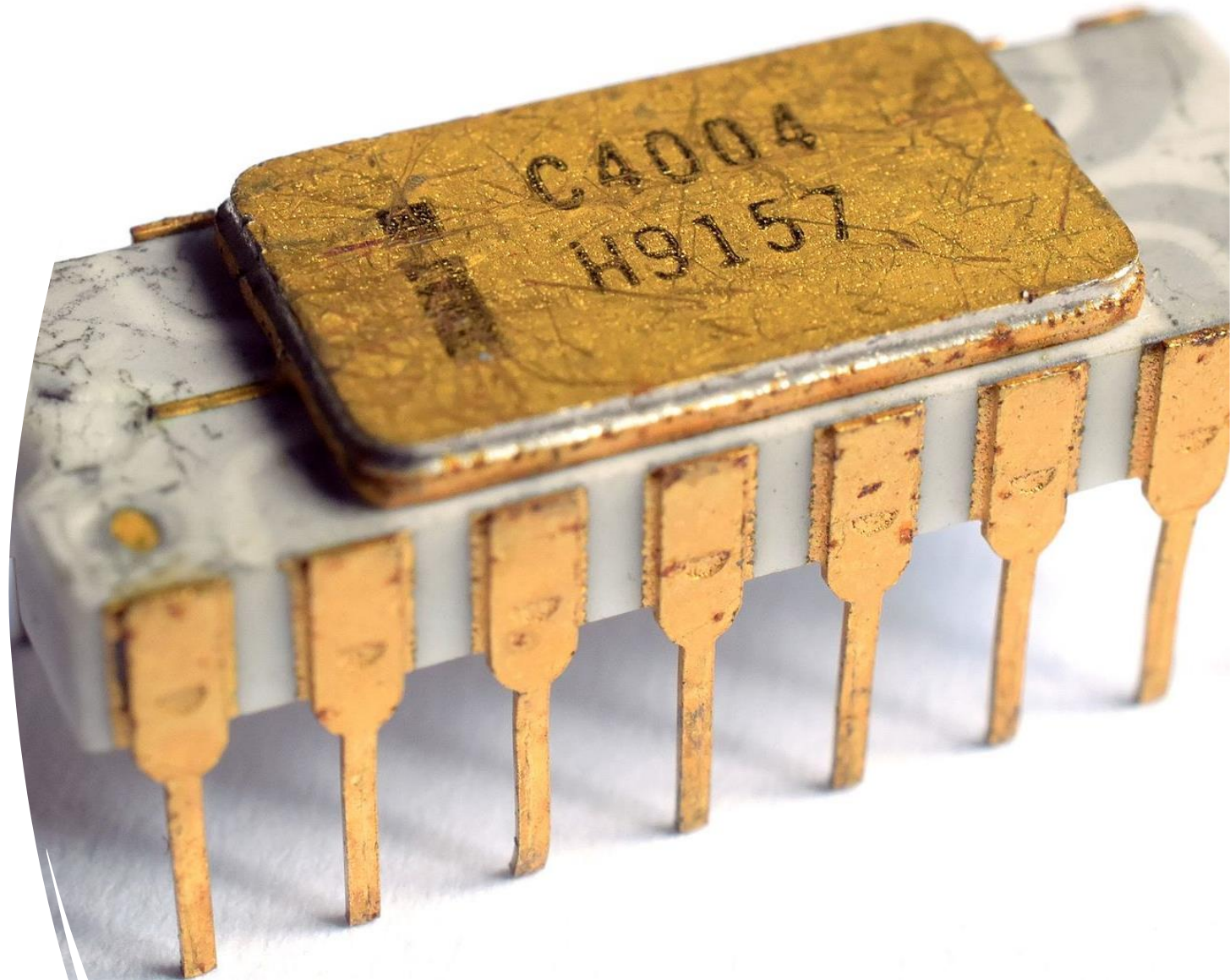




- IBM Personal Computer (1980)
  - processador Intel 8088 com 4.77MHz de clock e MS-DOS



- 
- Intel 4004 (1971)
  - Primeiro microprocessador com 2250 transistores





- 
- Apple I (1976)
  - Processador 6800 da Motorola



- 
- Osborne I (1981) – primeiro computador portátil, 11Kg, 64KB de memória, display de 5 polegadas





- 
- Macintosh (1984) – primeiro com drive de mouse e interface gráfica



- 
- IBM PS/2 (1987) – primeiro com mouse da IBM e drive 3 ½ (disquete)

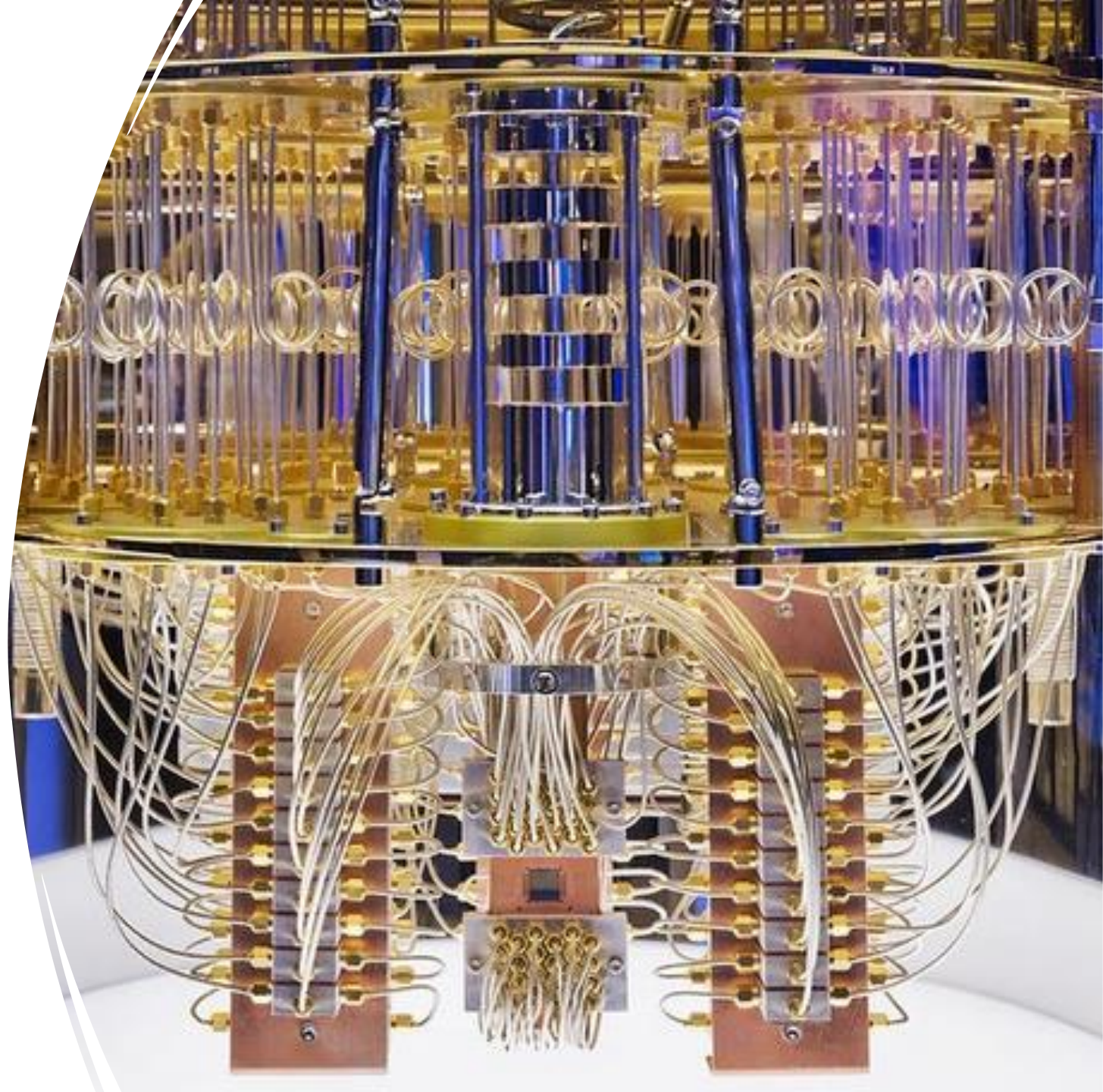




# Quinta Geração

---

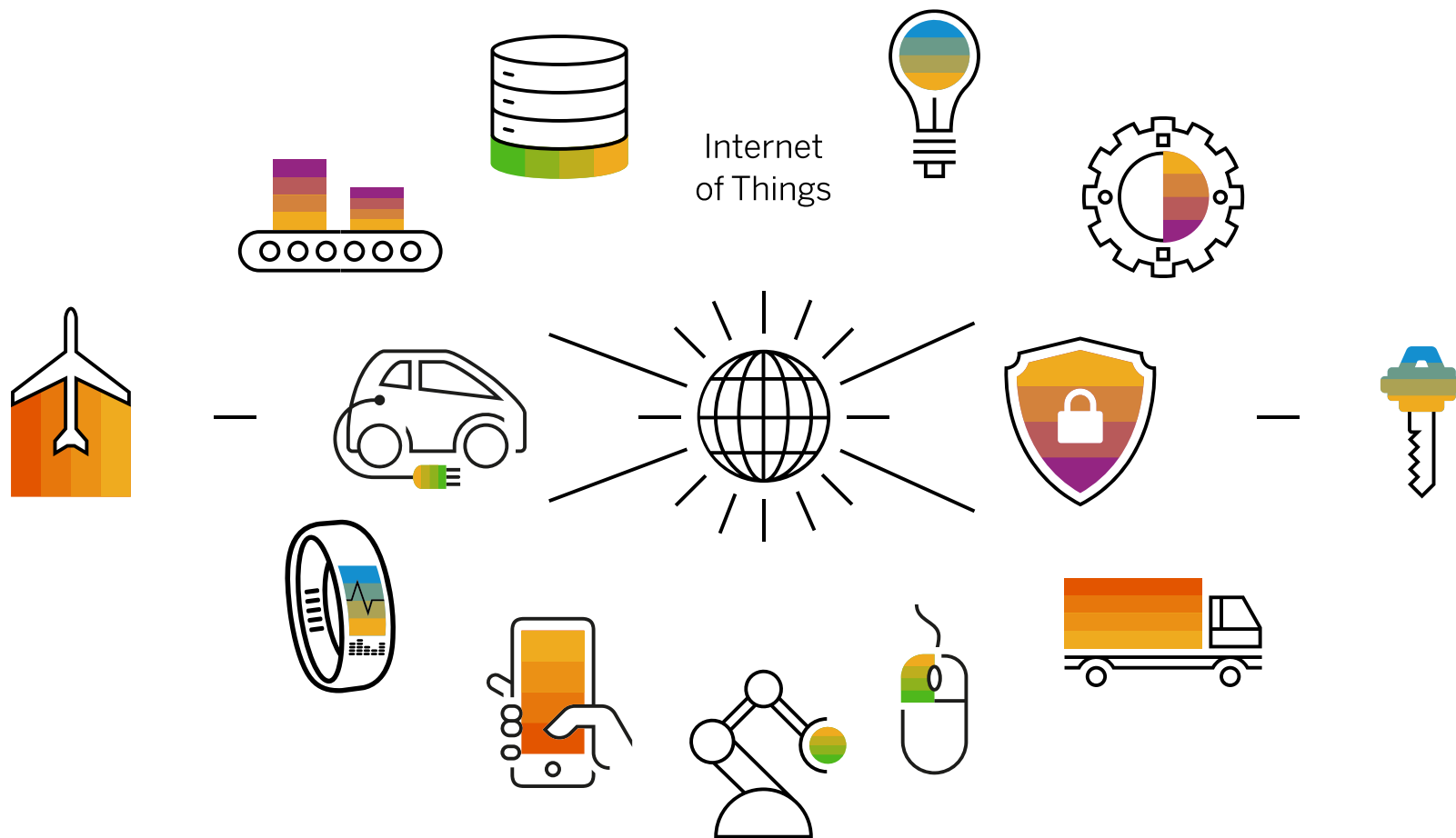
- Computação quântica e nanotecnologia
- Portáteis, leves
- Maior velocidade
- Maior memória
- Reconhecimento de linguagem natural
- Reconhecimento facial e de voz



- 
- Smartphones



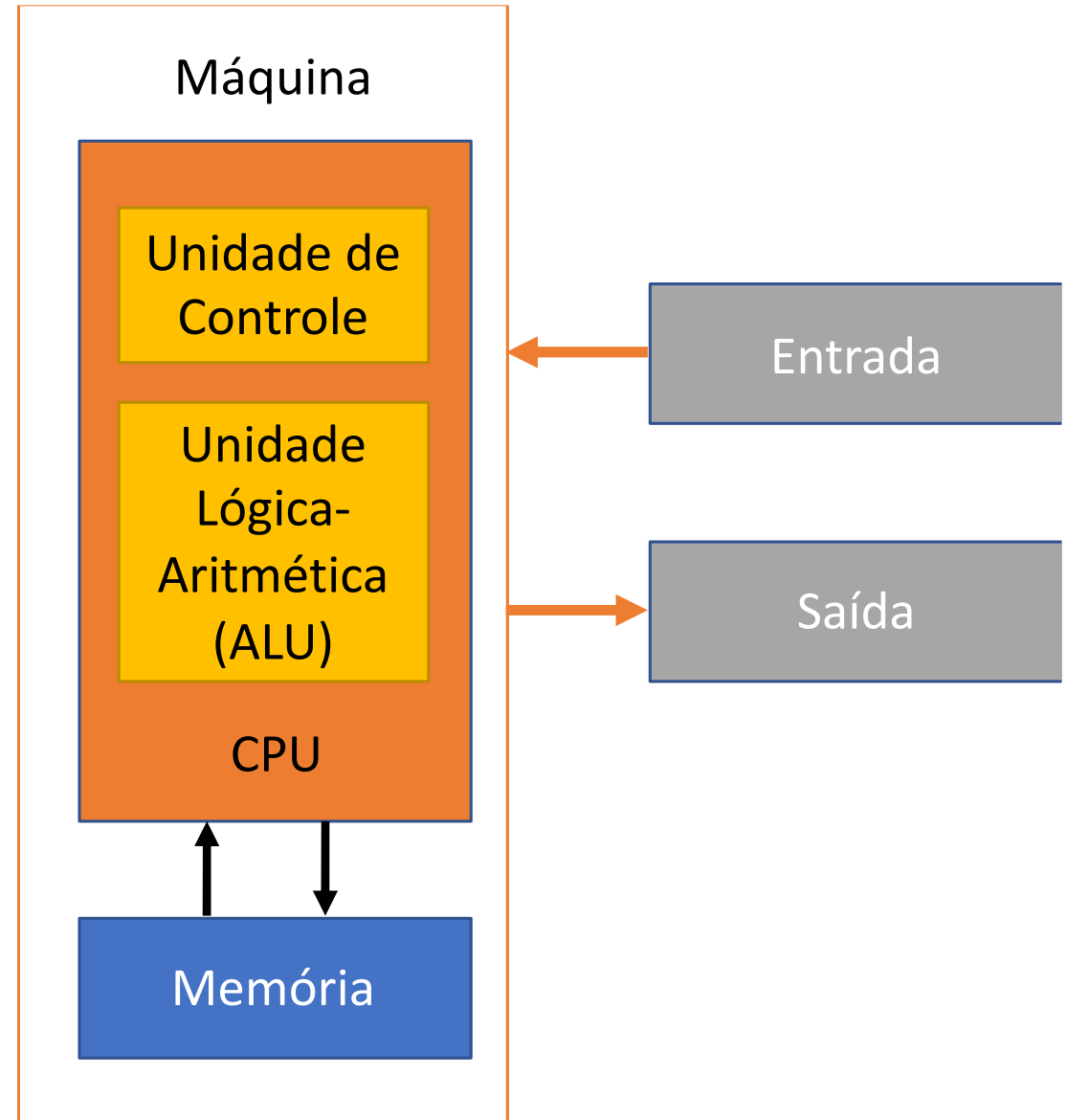
# O mundo hoje



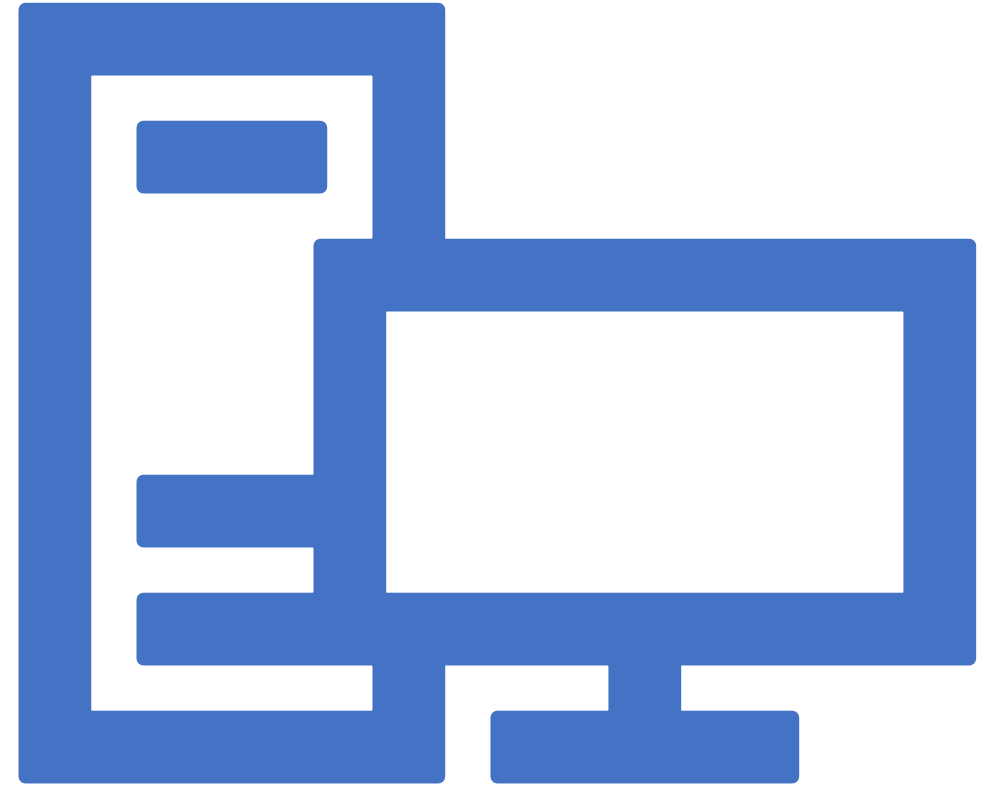
# Metodologia de Desenvolvimento de Algoritmos



- 
- Computador tem como finalidade receber, manipular e armazenar dados
  - Processamento de dados – receber dados de um dispositivo de entrada (teclado, mouse, joystick, ...), operar com base nesses dados para gerar resultado direcionado para um dispositivo de saída (monitor, projetor, impressora, ....)



- 
- Computador possui hardware (partes físicas) e software (programas) que operam de modo integrado
  - Linguagem de programação – sintaxe e regras definidas para a estruturação de um programa/software



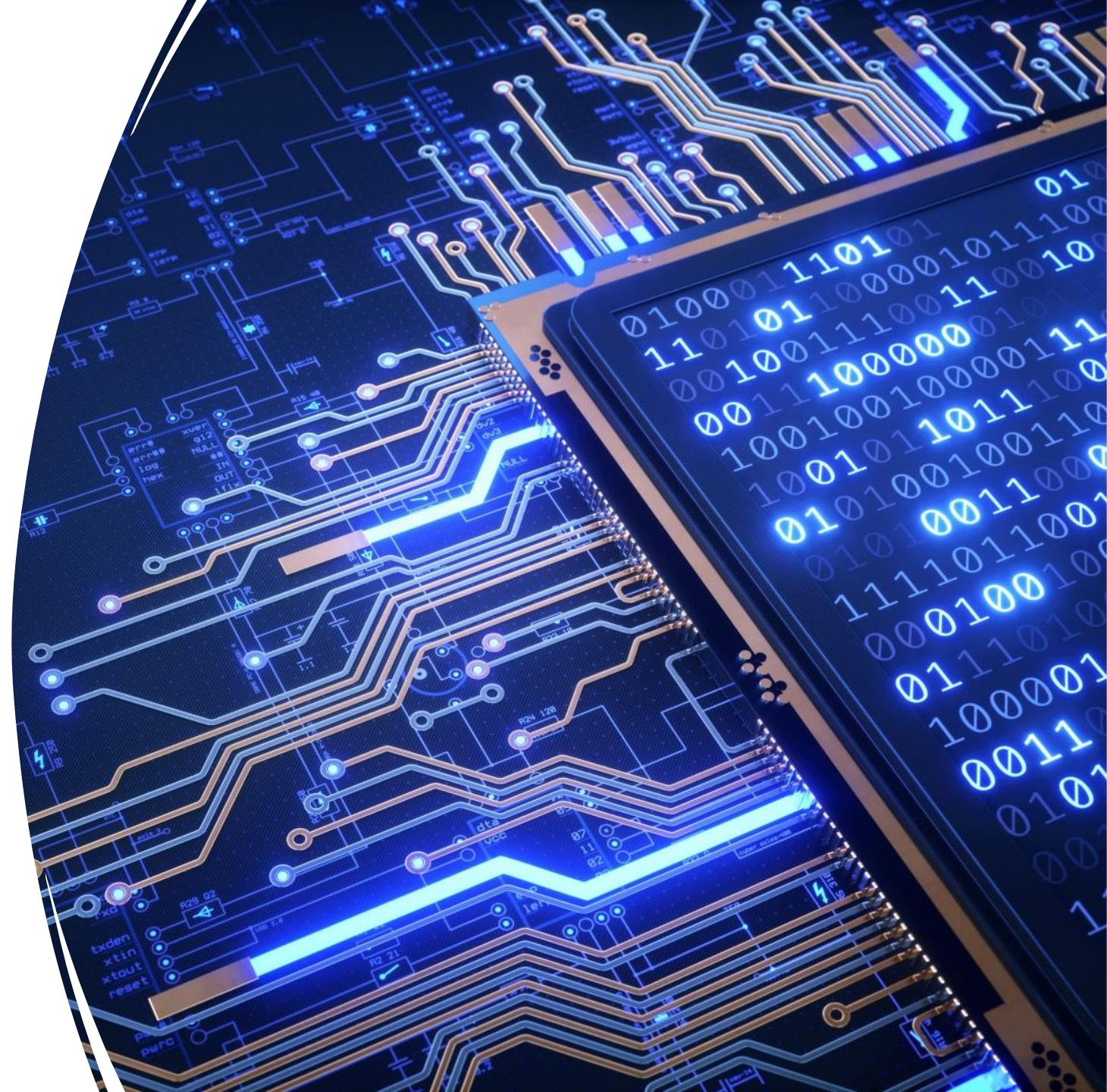


- 
- **Programa (ou Código/Software)**  
– conjunto de comandos que definem o que deve ser feito pelo computador.
  - **Linguagem de Programação** – conjunto de regras que define como escrever o programa. Ex.: C, C# (ou C Sharp), Java, Python.





- 
- **Editor do Código** – programa utilizado para escrever o programa. Ex.: Bloco de notas, editor de Código do Visual Studio, etc.
  - **Compilador** – programa que traduz o código escrito segundo a convenção estabelecida pela Linguagem de Programação em Linguagem de Máquina (ou Código Binário), compreendida pelo computador.





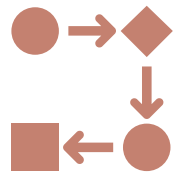
- 
- **IDE (Integrated Development Environment)** – integração dos elementos anteriores em um único ambiente. Ex. Code Blocks, Eclipse, replit.com.





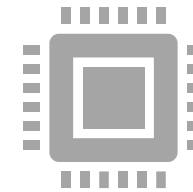
## Análise

Com base no enunciado do problema a ser resolvido, identifica-se as entradas, as saídas e o processamento necessário



## Algoritmo

Sequência de passos para resolver o problema. Podem existir diversas sequências possíveis. Esses podem ser apresentados por meio de narrativas, fluxogramas, pseudocódigo (ou português estruturado)



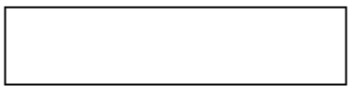

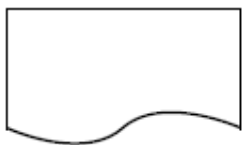
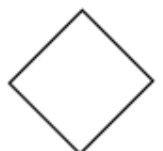


## Codificação

Representação do algoritmo em comandos da linguagem de programação



## Conjunto de símbolos utilizados no fluxograma.

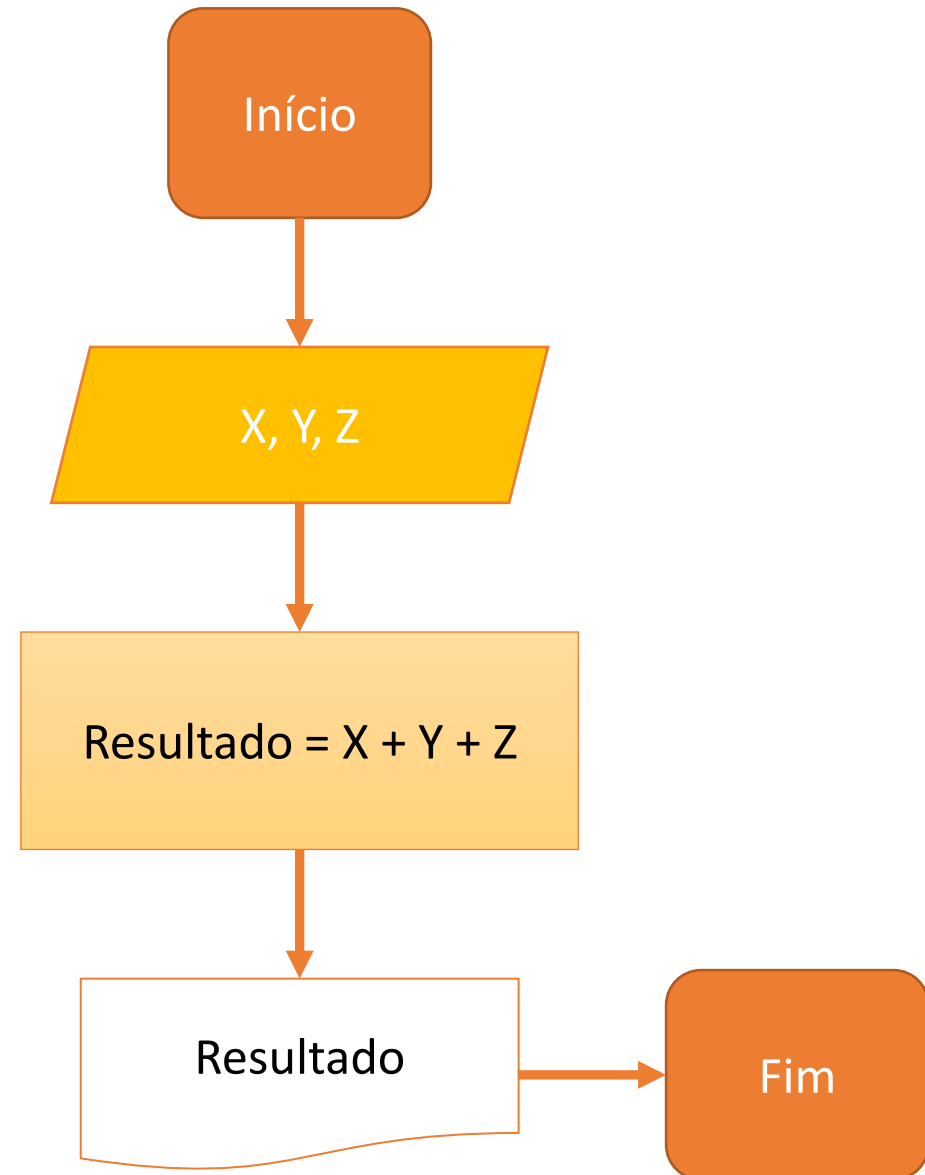
	Símbolo utilizado para indicar o início e o fim do algoritmo.
	Símbolo que permite indicar o sentido do fluxo de dados. Serve exclusivamente para conectar os símbolos ou blocos existentes.
	Símbolo utilizado para indicar cálculos e atribuições de valores.
	Símbolo utilizado para representar a entrada de dados.
	Símbolo utilizado para representar a saída de dados.
	Símbolo utilizado para indicar que deve ser tomada uma decisão, apontando a possibilidade de desvios.

$$\text{Resultado} \leftarrow X + Y + Z$$

**Problema** – somar 3 números quaisquer e exibir o resultado

**Algoritmo com descrição narrativa**

- Passo 1 – solicitar e memorizar cada um dos 3 números
- Passo 2 – realizar a operação de soma dos números e memorizar o resultado
- Passo 3 – exibir o resultado



---

**Resultado  $\leftarrow X + Y + Z$**

## **Pseudocódigo**

INÍCIO\_ALGORITMO

DECLARE X, Y, Z, resultado NUMÉRICO

ESCREVA “Digite o primeiro número”

LEIA X

ESCREVA “Digite o segundo número”

LEIA Y

ESCREVA “Digite o terceiro número”

LEIA Z

resultado  $\leftarrow X + Y + Z$

ESCREVA “A soma deles é ”, resultado

FIM\_ALGORITMO

**Resultado  $\leftarrow X + Y + Z$**

## Pseudocódigo

INÍCIO\_ALGORITMO

```
DECLARE X, Y, Z, resultado NUMÉRICO
ESCREVA "Digite o primeiro número"
LEIA X
ESCREVA "Digite o segundo número"
LEIA Y
ESCREVA "Digite o terceiro número"
LEIA Z
resultado  $\leftarrow X + Y + Z$ 
ESCREVA "A soma deles é ", resultado
```

FIM\_ALGORITMO

## Código em C

```
#include <stdio.h>

int main(void)
{
    int X, Y, Z, resultado;

    printf("Digite o primeiro número: ");
    scanf("%d",&X);
    printf("Digite o segundo número: ");
    scanf("%d",&Y);
    printf("Digite o terceiro número: ");
    scanf("%d",&Z);

    resultado = X + Y + Z;

    printf("A soma deles é %d\n",resultado);

    return 0;
}
```



# Paradigmas de Programação

Forma de organização do Código pelo desenvolvedor

**Estruturado (ou Procedural)** – solução pode ser dividida em funções, utilizando estruturas sequenciais, condicionais e de repetição

**Orientado por Objetos** – solução dividida em objetos que trocam mensagens entre si. Objetos são estruturas contendo estado (dados) e comportamento (lógica)

modificadorAcesso tipoRetorno  
nomeFuncao (parâmetros)

```
{  
    .....  
}
```

class NomeClasse

```
{  
    ....atributos.....  
    ....métodos.....  
}
```

# Comentários

Forma de realizar anotações no Código pelo desenvolvedor que serão ignoradas pelo compilador

Uma linha – será considerado comentário tudo o que estiver após o // até o final da linha

Mais de uma linha – será considerado comentário tudo o que estiver entre /\* e \*/

```
int quantidade; // indica quantidade de portas
```

```
/*
```

```
    Essa variável foi criada apenas para a versão 1,  
    contemplando a quantidade de portas existentes  
    no carro
```

```
*/
```