
Introdução ao Processamento de Dados

Introdução

**Instituto de Matemática e Estatística
(IME)
Universidade do Estado do Rio de Janeiro
(UERJ)**

Semestre 2016.1

Introdução ao Processamento de Dados

Objetivos

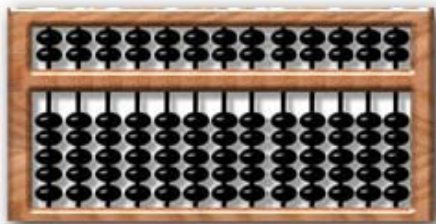
- **Compreender os conceitos fundamentais de processamento de dados num computador.**
- **Compreender e construir algoritmos.**
- **Saber fazer uso correto de técnicas de programação estruturada para a construção de programas.**
- **Conhecer as primitivas de programação oferecidas pela linguagem Python.**
- **Construir programas em Python.**

Introdução ao Processamento de Dados

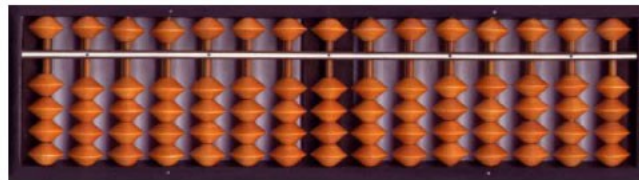
- Introdução ao conceito de algoritmo.
- Variáveis
- Constantes
- Tipos básicos de dados
- Atribuições
- Comandos de entrada e saída (E/S)
- Estruturas de controle de fluxo
- Estruturas de repetição
- Funções e procedimentos
- Vetores, strings, matrizes, dicionários
- Arquivos
- **OBS: A ementa detalhada será fornecida na segunda semana de março**

História

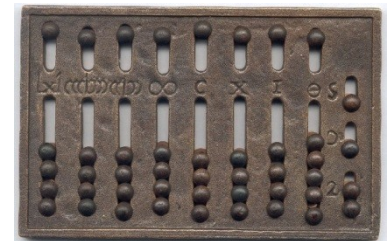
- Muitos aspectos da vida moderna não seriam possíveis se computadores.
- Aplicações: indústria, comércio, agricultura, comércio, cultura, ciência, educação, jogos, etc.
- Necessidade humana pela computação muito antiga: limitações humanas em relação aos cálculos mentais.
- Exemplo: ábaco (~300 A.C.)



Suanpan (chinês)



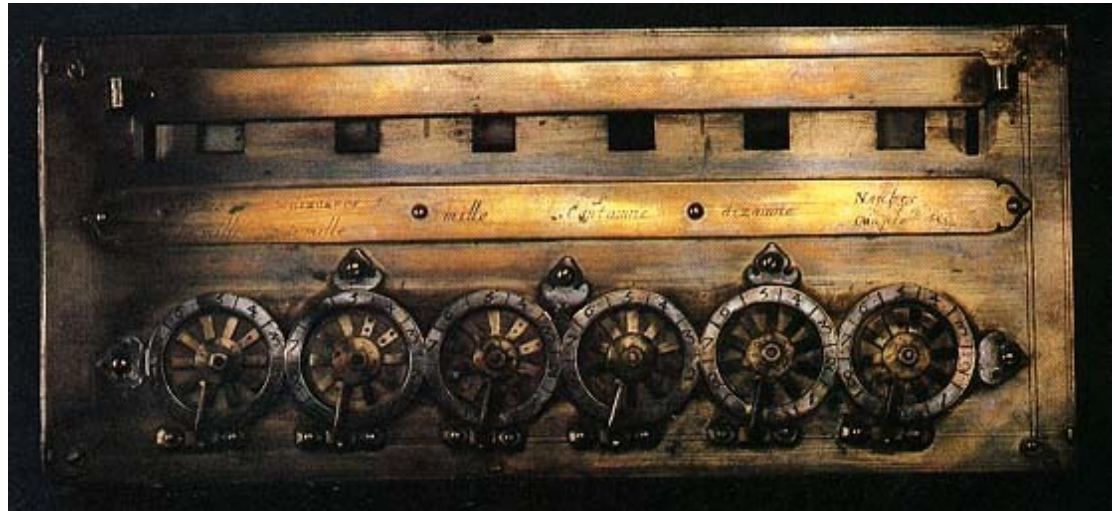
Soroban (japonês)



Abacus (romano)

História

- Precusores: máquinas mecânicas.
- Pascaline (soma e subtração), inventada por Blaise Pascal (1623-1662)



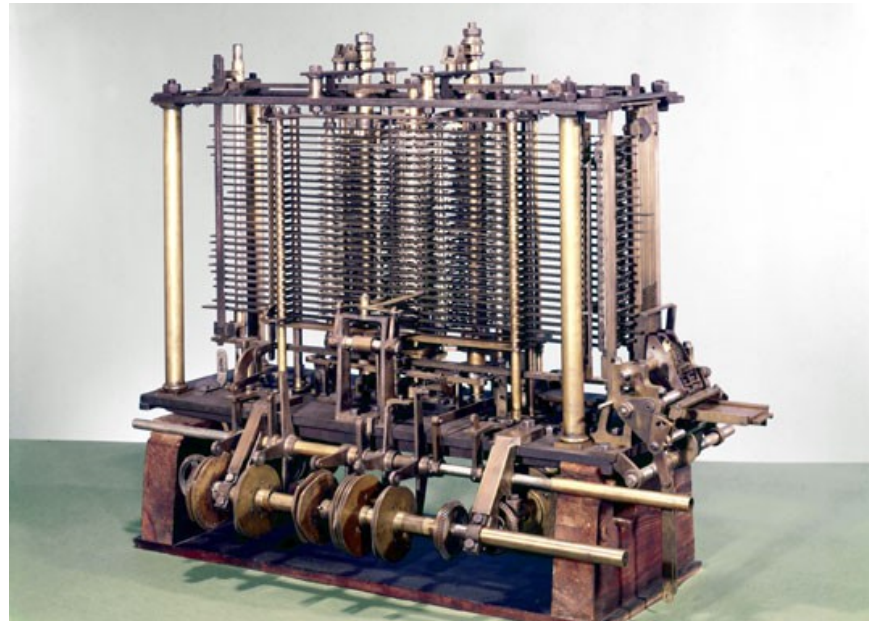
História

- Precusores: máquinas mecânicas.
- Máquina de Leibniz (quatro operações), Gottfried Wilhelm Leibniz (1646-1716)



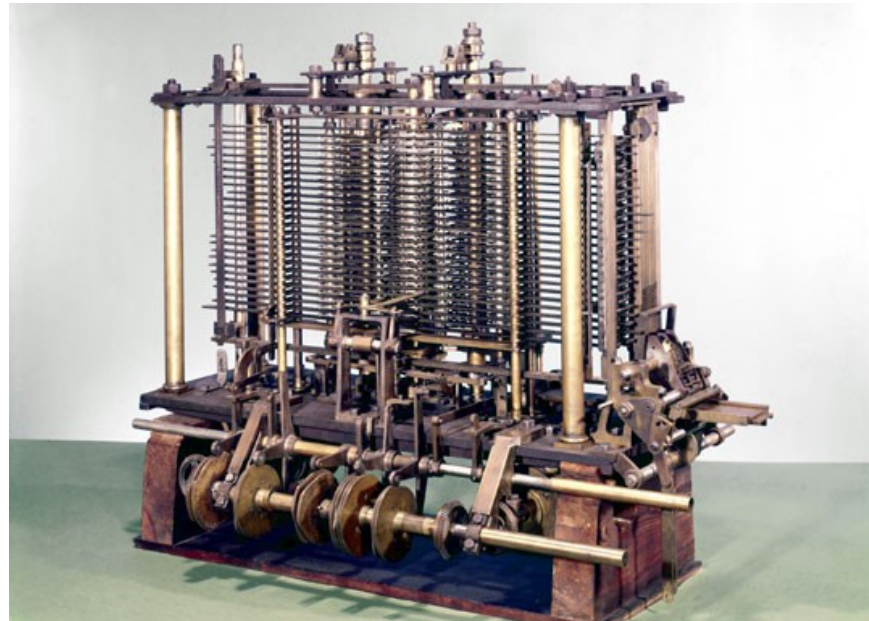
História

- **Precusores: máquinas mecânicas.**
- **Motor Analítico (cartões perfurados), Charles Babbage (1791-1871).**
- **Uso geral – programável, mas só foi construído em 1910.**



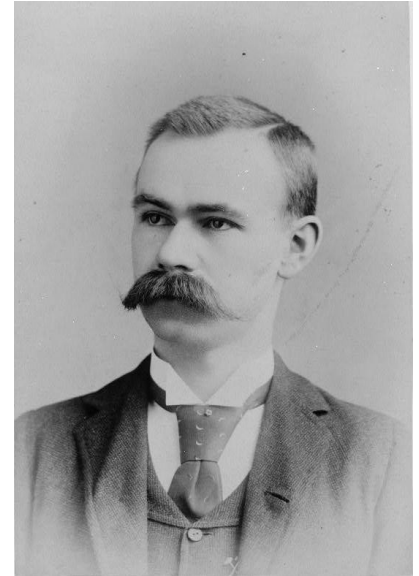
História

- **Precusores: máquinas mecânicas.**
- **Motor Analítico (cartões perfurados), Charles Babbage (1791-1871).**
- **Ada Byron (1815-1852): primeiro programador.**



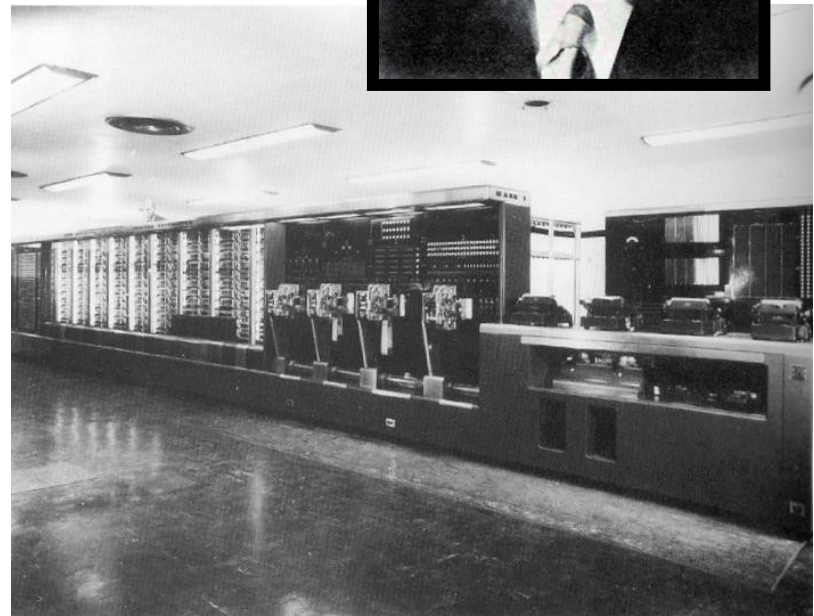
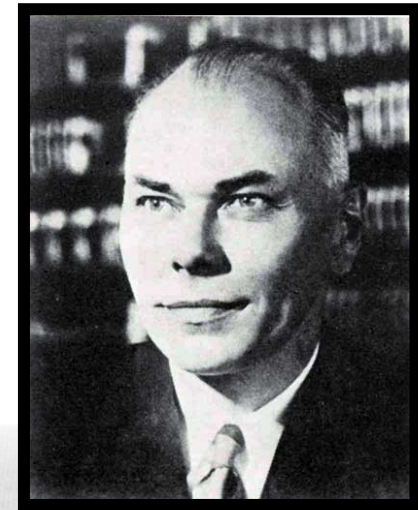
História

- **Precusores: máquinas eletromecânicas.**
- **Máquina de Hollerith (cartões perfurados), Hermann Hollerith (1860-1929).**
- **Cálculo de estatísticas para Censo Demográfico.**
- **Fundou a Tabulating Machine Company (1890), depois chamada de International Business Machines Corporation (IBM).**



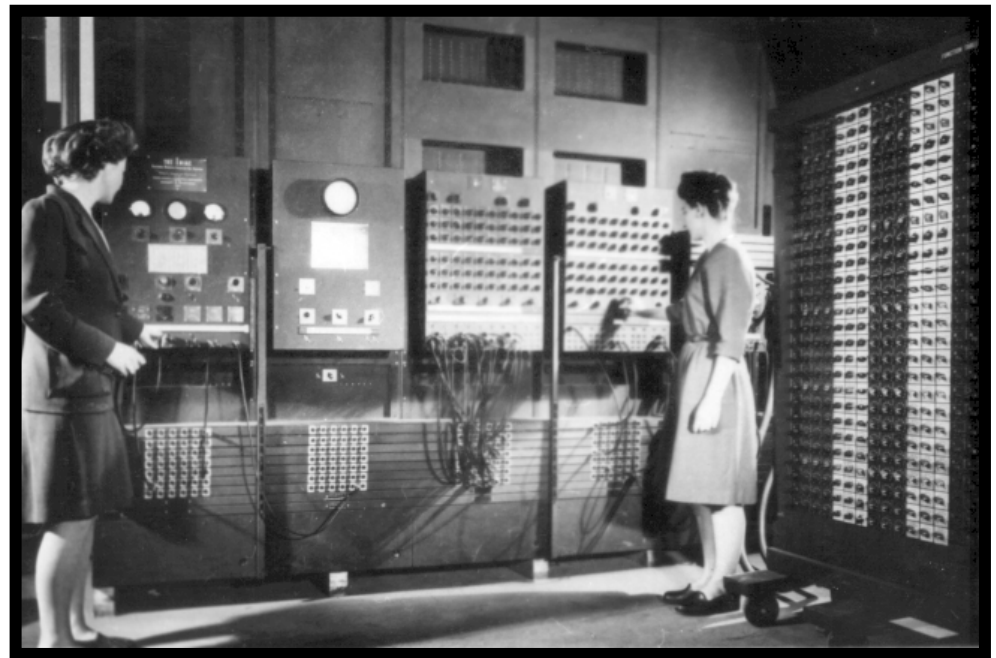
História

- **Precusores: máquinas eletromecânicas.**
- **Mark 1, Howard Aiken (1900-1973)**
- **Cálculos balísticos: primeiro computador programável nos EUA (1944)**
- **Programação: fitas perfuradas.**
- **5 toneladas, 800 km de fios, 2,5x15m.**



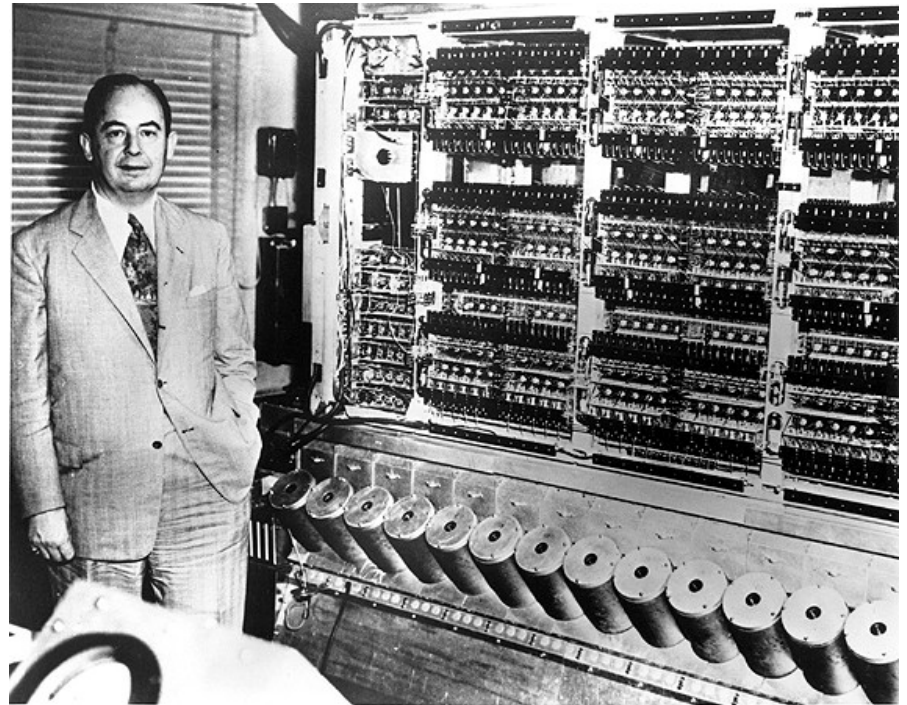
História

- **Primeiro computador eletrônico programável: Electronic Numerical Integrator and Calculator (ENIAC) - 1946.**
- **John William Mauchly (1907 -1980) e John Adam Presper Eckert Jr. (1919 -1995)**
- **Processamento decimal (não binário).**
- **18.000 válvulas, 30 tons, 1350 metros quadrados.**
- **Programável através de chaves/botões.**



História

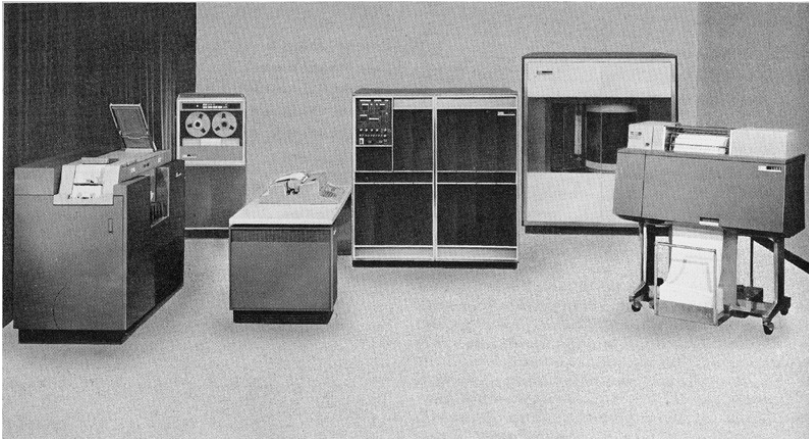
- **Electronic Discrete Variable Automatic Computer (EDVAC) - 1949.**
- **John William Mauchly (1907 -1980), John Adam Presper Eckert Jr. (1919 -1995) e Janos von Neumann (1903-1957).**
- **6.000 válvulas, 12.000 diodos, 7,8 tons, 45,5 m2.**
- **Aritmética binária.**
- **Programável através de cartões perfurados.**
- **Precursor da Arquitetura de von Neumann: dados e programas armazenados na memória do computador.**



História

- Segunda geração de computadores eletrônicos: transistores (1948).

IBM 1401 (1959)



IBM 7094 (1962)



História

- Segunda geração de computadores eletrônicos: transistores (1948).

DEC PDP-1 (1961)



DEC PDP-8 (1965)



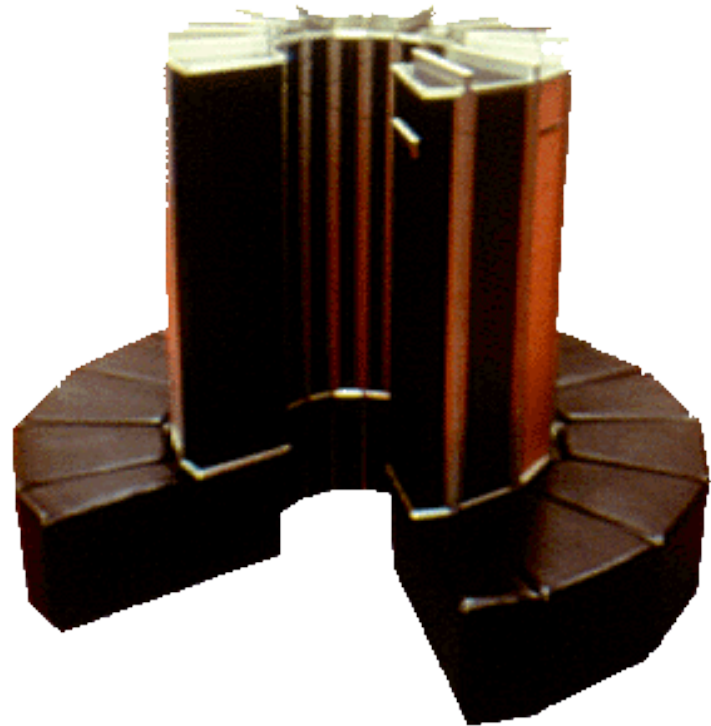
História

- Segunda geração de computadores eletrônicos: transistores (1948).

CDC 6600 (1964)



Cray-1 (1976)



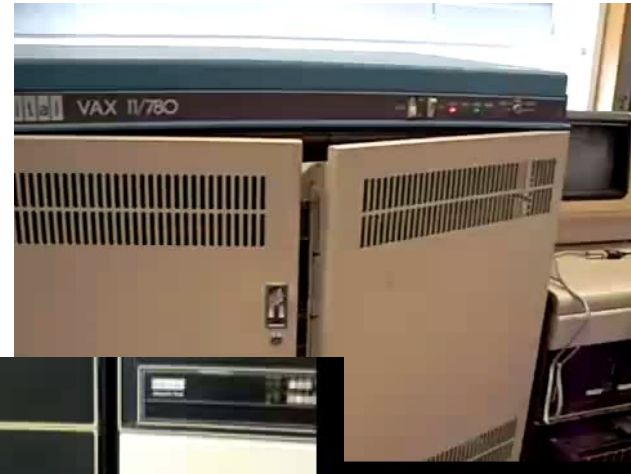
História

- Terceira geração: circuitos integrados (1958).

IBM 360 (1964)



VAX (1976)



DEC PDP-11 (1976)

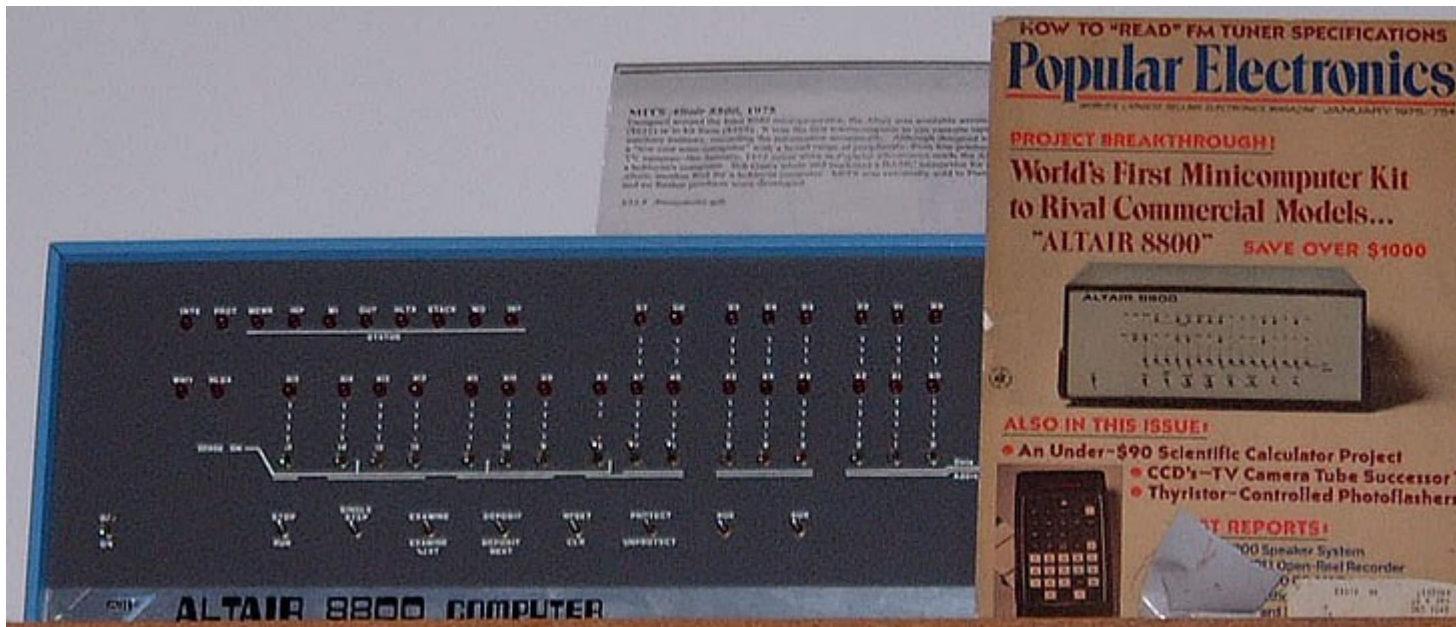
Introdução ao Processamento de
Dados

Unidade 1: Introdução

História

- Presente: era dos computadores pessoais.

Altair 8800 (1975)



História

- **Presente: era dos computadores pessoais.**

Apple I (1976)



Rare original 'Apple-1' computer sold by Steve Jobs goes on sale for £150,000



Apple II (1979)



História

- **Presente: era dos computadores pessoais.**

IBM PC (1981)

the first IBM-PC



Macintosh (1984)



História

- **Presente:** era dos computadores pessoais.

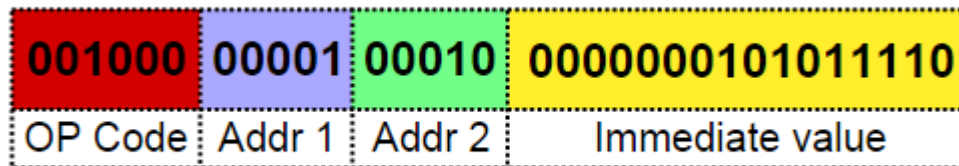
iPad (2010)



Linguagens de Programação

- **Código de máquina:** conjuntos de instruções que um processador (CPU) sabe executar, representada por sequências de bits.

MIPS32 Add Immediate Instruction



Equivalent mnemonic: **addi** \$r1, \$r2, 350

- **Linguagem de montagem (Assembly):** linguagem de baixo-nível, representação textual de códigos de máquina.
- **Linguagens de alto-nível:** facilitam a programação, são traduzidas por programas especiais para códigos de máquina.

Linguagens de Programação

- Primeira linguagem de alto-nível: Plankalkül, Konrad Zuse (1945).
- Autocode (1952), para o Mark 1.
- Fortran, pela IBM (metade dos anos 1950s).
- Lisp (1958), LISt Processing, linguagem funcional.
- FLOW-MATIC (1959), precursor do COBOL (1960).
- APL (metade dos anos 1960s).



Grace Hopper (1906-1992)
inventora do FLOW-MATIC

Linguagens de Programação

- **ALGOL (metade dos anos 1960s), precursora de diversas outras linguagens: Pascal, C, Simula.**
- **Simula (metade dos anos 1960s) e Smalltalk (metade dos anos 1970s), primeiras linguagens orientadas a objetos.**
- **C (entre 1969 e 1973).**
- **Prolog (1978), linguagem de programação em lógica matemática.**
- **C++ (começo dos 1980s).**
- **Pascal (1970), criada por Niklaus Wirth, que também criou outras linguagens como o Modula-2.**
- **Muitas outras vieram depois: Perl, Java, Python, Ruby, Groovy, PHP, C#, Lua, etc.**

Sistema Computacional

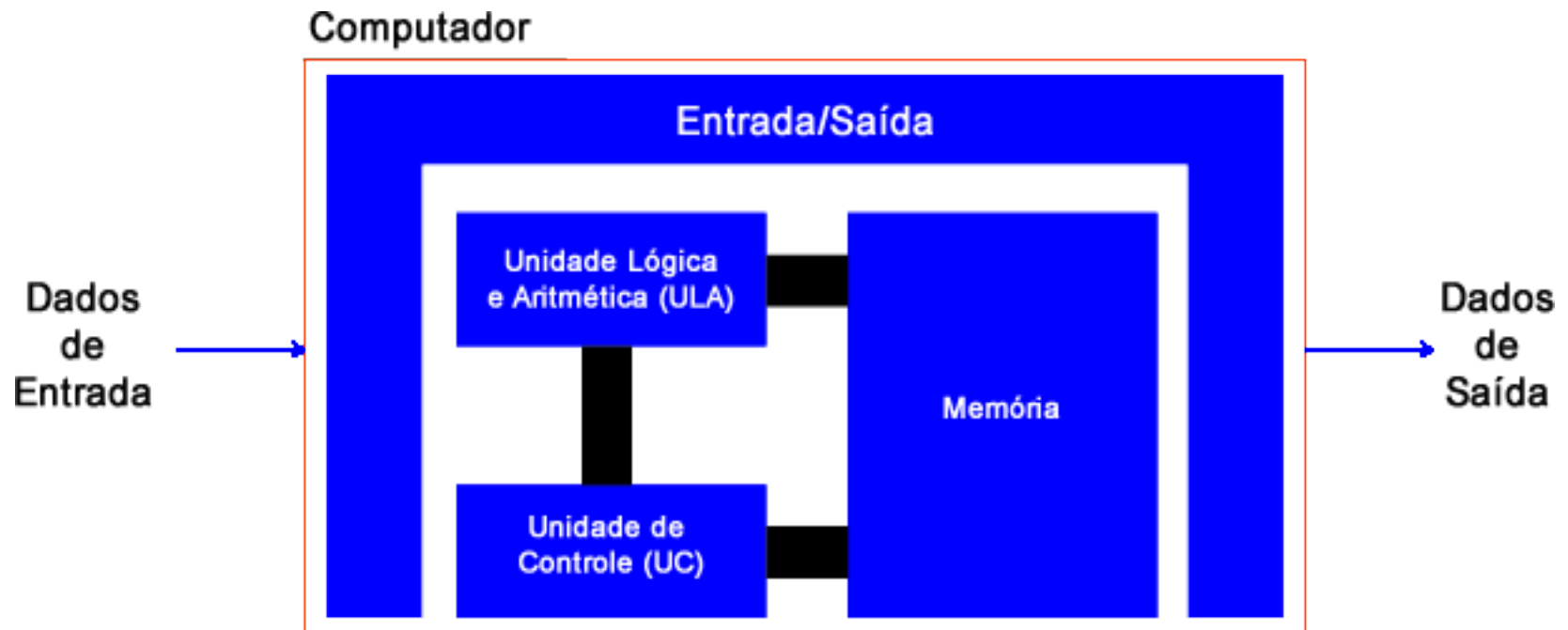
- **Componentes de um Sistema Computacional: Hardware e Software.**
- **Hardware: partes físicas (computador).**
- **Software: conjunto de instruções previamente gravados (programas).**
- **Um não funciona sem o outro.**

Arquitetura de von Neumann

- A principal característica da arquitetura de von Neumann é o conceito de programa.
- Programar os primeiros computadores significava modificar os sistema de fios, ligar ou desligar um conjunto de comutadores.
- Eram necessários operadores ou engenheiros especializados.
- No modelo de Von Neumann, os programas são armazenados na memória do computador e não somente os dados.

Arquitetura de von Neumann

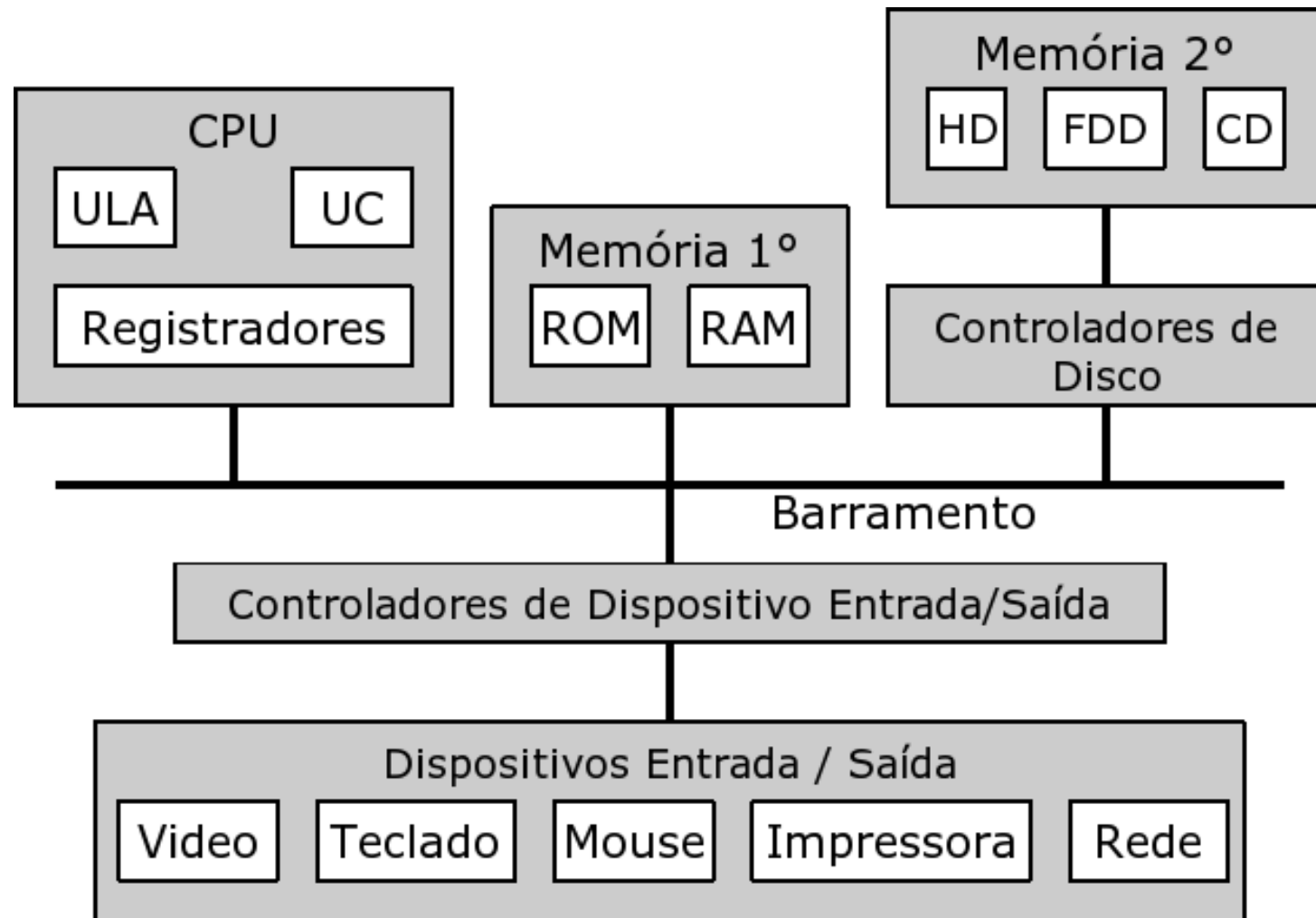
- Hardware do computador composto por quatro subsistemas principais.
- Memória, Unidade Lógica e Aritmética, Unidade de Controle e Entrada/Saída



Arquitetura de von Neumann

- **Memória**
 - Área de armazenamento de programas e dados durante o processamento
- **Unidade Lógica e Aritmética (ULA)**
 - Responsável pelas operações de lógica e cálculo sobre os dados
- **Unidade de Controle (UC, CPU)**
 - Controla as operações da memória, a ULA e o subsistema de entrada/saída
- **Entrada/Saída (E/S)**
 - Recebe dados de fora do computador e envia os resultados para o mundo externo

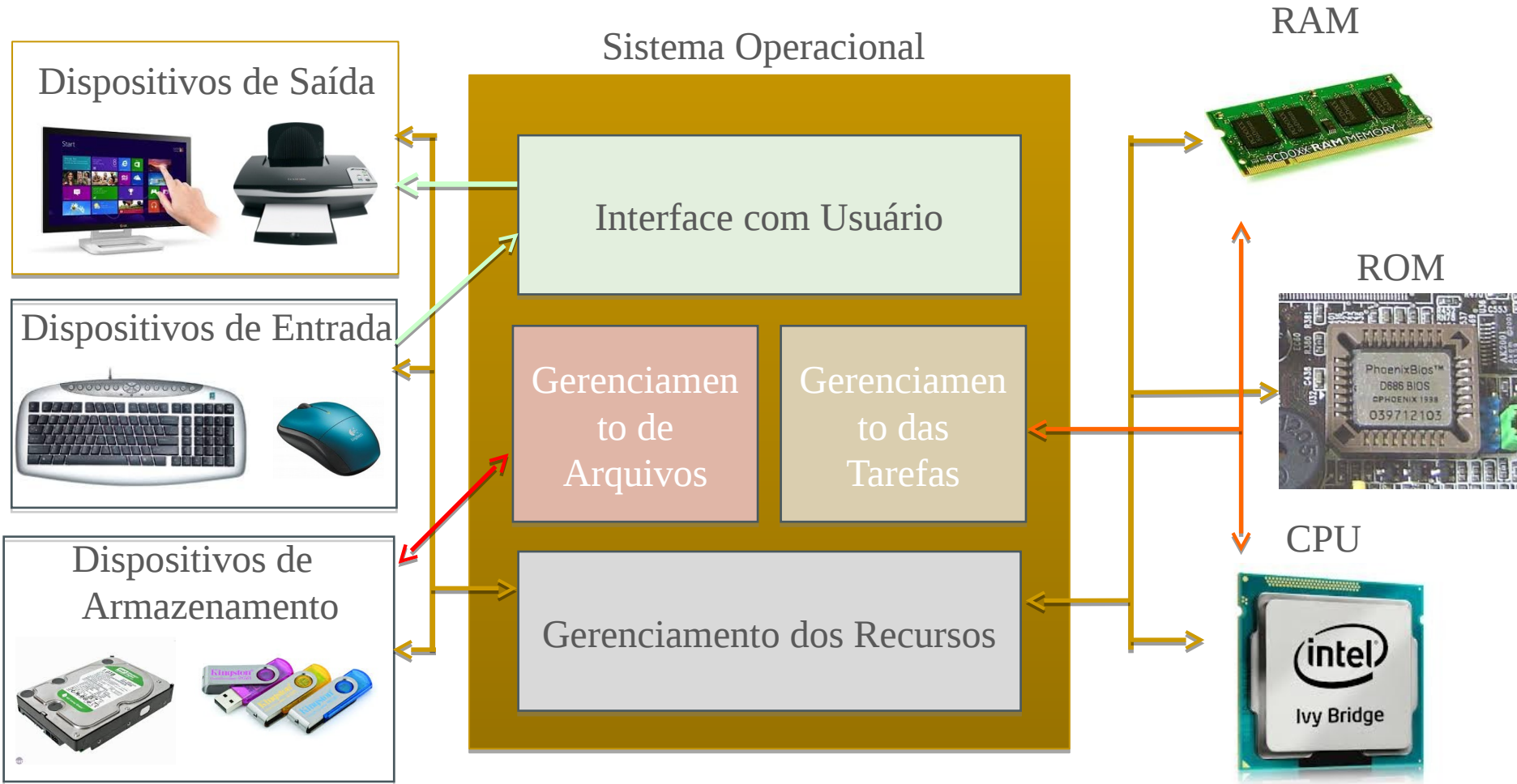
Componentes de Hardware



Software

- **Software Básico**
 - **Sistemas Operacionais**
 - **Compiladores**
 - **Interpretadores**
 - **Sistemas de Redes**
- **Software Aplicativo**
 - **Editores de Texto**
 - **Planilhas**
 - **Jogos**
 - **Etc.**

Sistema Operacional



Sistema Operacional

Alguns sistemas operacionais:

- **Windows**
- **Linux**
- **Unix**
- **Solaris**
- **Mac OS**
- **iOS**
- **Android**
- **Windows Phone**

Sistema Operacional

Sistemas que traduzem as linguagens de programação para código de máquina.

- **Compiladores: verificam e decodificam todas as instruções do programa fonte, gerando um código executável.**
- **Exemplos de linguagens compiladas: Fortran, Pascal, C, Pascal, C++ e Delphi.**
- **Interpretadores: verifica, decodifica e executa instrução a instrução.**
- **Exemplos de linguagens interpretadas: Java, HTML, Javascript, ASP, Perl, C#, Python e PHP.**

Compiladores e Interpretadores

Sistemas que traduzem as linguagens de programação para código de máquina.

- **Compiladores:** verificam e decodificam todas as instruções do programa fonte, gerando um código executável.
- **Exemplos de linguagens compiladas:** Fortran, Pascal, C, Pascal, C++ e Delphi.
- **Interpretadores:** verifica, decodifica e executa instrução a instrução.
- **Exemplos de linguagens interpretadas:** Java, HTML, Javascript, ASP, Perl, C#, Python e PHP.

Processamento

- **Funcionamento básico de um computador.**
- **Lê instruções (pedacinhos de um programa) e as executa sequencialmente.**
- **Exemplos de instruções: LER, ESCREVER e SOMAR**
 - **LER: lê um valor (de um dispositivo de E/S) e o armazena na memória. Destrói o valor que existia antes na memória.**
 - **ESCREVER: escreve um valor (em um dispositivo de E/S). Não destrói o valor que existia na memória.**
 - **SOMAR: soma dois valor e armazena o resultado na memória.**

Processamento

Instruções (programa)	Execução	Resultado
Início	A = 2	5
Ler um valor para A	B = 3	
Ler um valor para B	C = 2 + 3	
Somar os conteúdos de A e B e colocar em C		
Escrever o conteúdo de C		
Fim		

Bi

T A B L E 86						M E M O I R E S D E L' A C A D E M I E R O Y A L E
D E S						bres entiers au-deffous du double du
N O M B R E S.						plus haut degré. Car icy, c'est com-
.	me si on disoit, par exemple, que III
.	.	.	.	I	1	ou 7 est la somme de quatre, de deux
.	.	.	I	C	2	Et que IIIOI ou 13 est la somme de huit, quatre
.	.	I I			3	& un. Cette propriété sert aux Essayeurs pour
.	I O O				4	peler toutes sortes de masses avec peu de poids,
.	I O I				5	& pourroit servir dans les monnoyes pour don-
.	I I O				6	ner plusieurs valeurs avec peu de pieces.
.	I I I				7	Cette expression des Nombres étant établie , sert à
I O O O					8	faire tres-facilement toutes sortes d'operations.
I O O I					9	
I O I O					10	
I O I I					11	
I I O O					12	
I I O I					13	
I I I O					14	
I I I I					15	

Leibniz, 1705

Sistema de Numeração

- Estamos familiarizados com a base 10 (decimal), já os computadores atuais trabalham exclusivamente com a base 2 (binário).
- É comum o uso de bases numéricas derivadas de 2 ao se utilizar computadores em baixo nível.
- Durante o processamento números são convertidos para a base binária, cálculo (processamento) é feito e resultado convertido para outra base (decimal).
- Em muitas situações a digitação de códigos binários é complicada/longa para o programador.
- Existem outros códigos que facilitam a digitação: base 8 (octal), base 16 (hexadecimal).

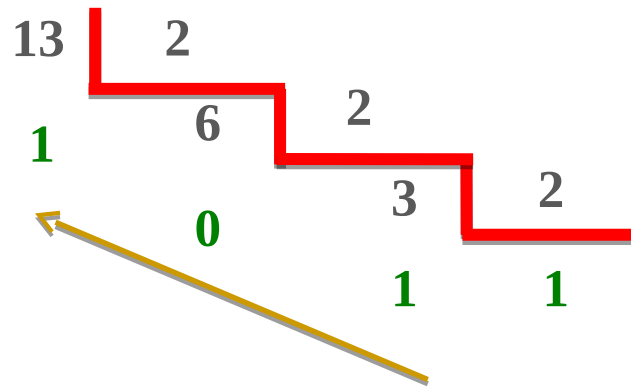
Sistema de Numeração

■ Exemplos:

10 (decimal)	2 (binário)	8 (octal)	16 (hexadecimal)
0	0	0	0
3	11	3	3
10	1010	12	A
15	1111	17	F
301	100101101	455	12D
1379	10101100011	2543	563
42685	1010011010111101	123275	A6BD

Sistema de Numeração

- Transformação do decimal 13 em binário:



- Transformação do binário 1101 em decimal:

$$1 \times 2^3 + 1 \times 2^2 + 0 \times 2^1 + 1 \times 2^0 = 13$$

Sistema de Numeração

- **Transformação de binário em octal (tabela de conversão):**

Octal	0	1	2	3	4	5	6	7
Binário	000	001	010	011	100	101	110	111

- **Para converter $(472)_8$ em binário, substituir: 4 por 100; 7 por 111; 2 por 010.**
- **Desta forma, obtemos o número binário 100111010.**

Sistema de Numeração

- **Transformação de binário em hexadecimal (tabela de conversão):**

Hexadecimal	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
Binário	0000	0001	0010	0011	0100	0101	0110	0111	1000	1001	1010	1011	1100	1101	1110	1111

- **Para converter $(3A6)_{16}$ em binário, substituir: 3 por 011 ; A por 1010;
6 por 0110.**
- **Desta forma, obtemos o número binário 1110100110.**

Sistemas de Numeração

**Transformação de binário em octal
(tabela de conversão):**

Dígito octal	0	1	2	3	4	5	6	7
Binário equivalente	000	001	010	011	100	101	110	111

**Para converter $(472)_8$ em binário, substituir:
4 por 100; 7 por 111; 2 por 010.**

Desta forma, obtemos o número binário 100111010.