INE5231 Computação Científica I

Prof. A. G. Silva

07 de março de 2017

Disciplina

INE5231 – Computação Científica I

- **Turma:** 02203A
- **Professor:** Alexandre Gonçalves Silva
  - https://www.inf.ufsc.br/~alexandre.goncalves.silva/
  - alexandre.goncalves.silva@ufsc.br
  - Sala INE-506
- **Carga horária:** 54 horas-aula • Teóricas: 54 • Práticas: 0
- **Cursos:** Engenharia Eletrônica (235); Engenharia Mecânica (203)
- **Requisitos:** *Não há*
- **Período:** 1º semestre de 2017
- **Horários:**
  - 3ª 09h10 (3 aulas) – CTC204/LIICT8

Ementa

Ementa: Noções de sistemas de computação. Formulação de algoritmos e sua representação. Noções sobre linguagem de programação e programas. Implementação prática de algoritmos em uma linguagem de programação. Descrição de algumas aplicações típicas.

Objetivos

- **Geral:** • Analisar problemas e elaborar algoritmos para sua solução de forma clara e precisa usando uma linguagem de programação e implementá-los.

- **Específicos:** • Dominar a atividade de análise detalhada de problemas. • Compreender o processo de Engenharia de Software, aplicando os passos de definição do problema, análise de requisitos, modelagem, implementação e testes. • Modelar algoritmos em notação algorítmica adequada. • Implementar algoritmos em uma linguagem de nível intermediário. • Dominar ambientes de desenvolvimento de programas na resolução de problemas matemáticos

Conteúdo programático – Parte I

- **O COMPUTADOR - [3 horas-aula]:** • Arquitetura de Computadores • Linguagens de Programação • Programa Conversores.
- **REPRESENTAÇÃO DE ALGORITMOS - [3 horas-aula]:** • Conceito de Algoritmo • Pseudo-Código para Representar Algoritmos • Tipos de Dados • Variáveis • Tipos de Variáveis • Desenvolvimento Modularizado de Algoritmos
- **LINGUAGENS DE PROGRAMAÇÃO ESTRUTURADAS [3 horas-aula]:** • Paradigma Estruturado e Estruturas de Controle • Comandos de Atribuição • Variáveis, Constantes, Funções e Procedimentos • Comandos de Entrada/Saída: Teclado, Vídeo e Arquivos • Compilação,Interpretação,Execução,Depuração de Programas
- **INTRODUÇÃO À PROGRAMAÇÃO EM C [6 horas-aula] :** • Estrutura do Programa em C • Sintaxe e comandos • Compilação, Teste e Debugação de Programas.

Conteúdo programático – Parte II

- **PROGRAMAS ENVOLVENDO PROCESSOS DE REPETIÇÃO E SELEÇÃO [6 horas-aula]:** • Estruturas de Seleção • Estruturas de Repetição
- **VARIÁVEIS ESTRUTURADAS UNIDIMENSIONAIS HOMOGÊNEAS [9 horas-aula]:** • O Tipo Array e seus similares • Problemas vetoriais no Rn e sua resolução computacional
- **VARIÁVEIS ESTRUTURADAS MULTIDIMENSIONAIS HOMOGÊNEAS [6 horas-aula]:** • O Tipo Matriz e seus similares • Problemas da Álgebra Linear e sua resolução computacional • Resolução computacional de sistemas de equações lineares

Conteúdo programático – Parte III

- **VARIÁVEIS ESTRUTURADAS HETEROGÊNEAS [6 horas-aula]:** • O Tipo Registro ou Estrutura • Campos, Tipos Derivados Estruturados e Referência de Campos • O Tipo Abstrato de Dados e Programação Modular • Armazenamento de dados em arquivos externos
- **SUBDIVISÃO DE PROBLEMAS E SUBPROGRAMAÇÃO [6 horas-aula]:** • Técnicas para Subdivisão de Problemas • Subprogramas sem retorno de valor: Procedimentos • Subprogramas com retorno de valor: Funções • Passagem de parâmetros por valor e por referência • Desenvolvimento e uso de bibliotecas de funções matemáticas
- **PROGRAMAÇÃO UTILIZANDO UMA LINGUAGEM DE COMPUTAÇÃO TÉCNICA NUMÉRICA [6 horas-aula]:** • Conceitos de Ambientes de Simulação Matemática • Resolução Numérica de Integrais e Representação Gráfica

Metodologia e avaliação

Metodologia:

- Uma parte das aulas será expositiva, utilizando o quadro e o projetor multimídia, para a apresentação do conteúdo da disciplina. Outra parte será prática, com problemas sendo resolvidos em sala de aula, no ambiente de programação do Moodle e no laboratório, de modo a exercitar o aprendizado. Uma lista de problemas será proposta semanalmente, sendo um exercício selecionado para submissão e avaliação.

Avaliação:

- A avaliação constará de três provas (P_1, P_2 e P_3) e aproximadamente n (aproximadamente 18) exercícios (E_i). A média final (MF) é:

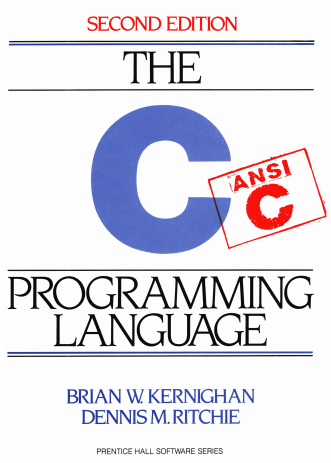
$$MF = 0,3P_1 + 0,3P_2 + 0,3P_3 + 0,1\ \frac{\sum_{i=1}^{n} E_i}{n}$$

- Datas previstas para as avaliações:
  - P_1 - 18abr
  - P_2 - 23mai
  - P_3 - 20jun
  - REC - 04jul (apenas para MF entre 3,0 e 5,5)

Bibliografia

Básica:

- KERNIGHAN, Brian W.; RITCHIE, Dennis M. C, a linguagem de programação. 4. ed. Porto Alegre: EDISA; Rio de Janeiro: Campus, 1988. 208p ISBN 8570014104



- TENENBAUM, Aaron M; LANGSAM, Yedidyah; AUGENSTEIN, Moshe. Estruturas de dados usando C. São Paulo (SP): Pearson Makron Books, 1995. 884p. ISBN 8534603480
- Tanenbaum. Organização estruturada de computadores, 3a Edição, Rio de Janeiro: PHB, 1995.
- SCHILDT, Herbert. C, completo e total. 3. ed. rev. e atual. São Paulo (SP): Pearson Education do Brasil, 2006. xx,827p.
- OLIVEIRA, U. Programando em C, vol. I – fundamentos. Editora Ciência Moderna, 2008, 743p.

Bibliografia (cont...)

Complementar:

- KERNIGHAN, Brian W.; PLAUGER, P. J. The elements of programming style. 2nd ed. New York: Yourdon: McGraw-Hill, c1978. 168p ISBN 0070342075
- BORATTI, I.C. e OLIVEIRA, A B. Introdução a Programação – Algoritmos. Visual Books Florianópolis -2007.
- PRESS, W. H, et all. Numerical Recipes in C, The art of Scientific Computing, 2nd Ed. Cambridge University Press, 2002. Disponível em: http://www.nrbook.com/a/bookcpdf.php
- Acton, F. S. Numerical Methods that Work. (Rev. 1970, Harper & Row edition), The Math. Assoc. Am., 1990.

Introdução

- Informática
  - "Ciência do tratamento automático das informações";
    - ★ que permitam aprimorar e automatizar tarefas em qualquer área de atuação da sociedade
    - ★ engloba toda atividade relacionada ao desenvolvimento e uso dos computadores

Computador

- Definição de "**computador**"
  - que computa; calculador, calculista;
  - máquina destinada ao processamento de dados; dispositivo capaz de efetuar uma tarefa a partir de instruções.

Histórico e Evolução dos Computadores

- Necessidade de contar (por exemplo, número de ovelhas): utilização dos dedos (digitus – lat. dedo) e pedras (calculus – lat. pedra).

- **Ábaco** (lat. abacus, e esta veio do grego abakos, forma genitiva abax (tábua de cálculos) como primeiro instrumento de cálculo (Mesopotâmia, depois China) a mais de 5000 anos.

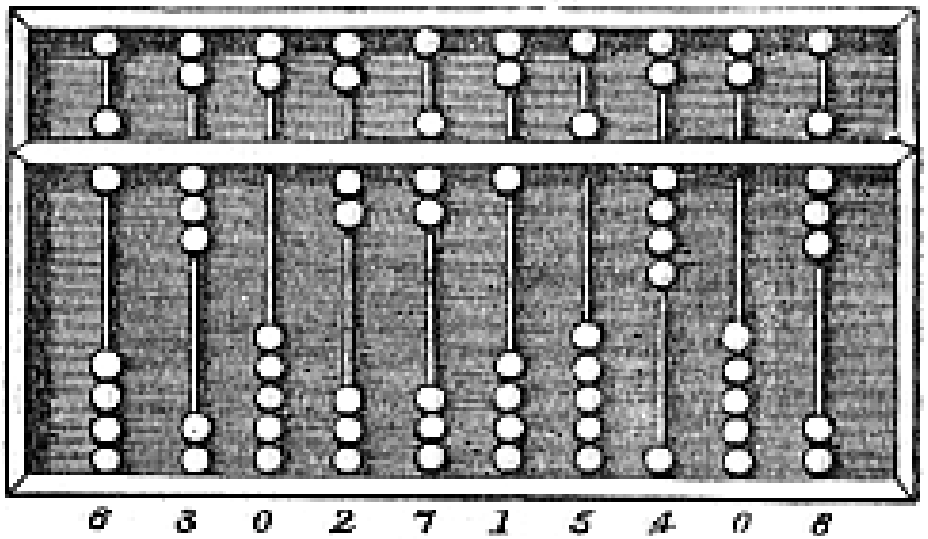


Figure: Representação do número 6302715408.

Histórico e Evolução dos Computadores

- Máquinas de calcular:
  - **Schickard (1623)**
    - ★ Primeiro a construir uma máquina de calcular mecânica
    - ★ Engrenagens para representar números e realizar cálculos
    - ★ Capaz de realizar as 4 operações básicas com números de seis dígitos e indicar um overflow por meio do toque de um sino
  - **Pascal (1642)**
    - ★ Máquina com 6 rodas dentadas, cada uma contendo algarismos de 0 a 9
    - ★ Permitia somar até 3 parcelas de cada vez, desde que o total não ultrapassasse 999 999
    - ★ A multiplicação era feita a partir de somas
    - ★ Vida útil de quase 200 anos, sendo aperfeiçoada por diversos inventores

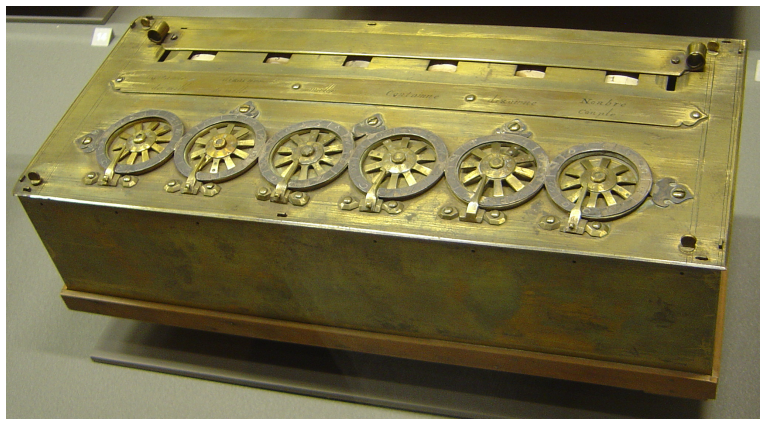


Figure: Máquina de Pascal ou Pascalina.

Histórico e Evolução dos Computadores

- Máquinas de calcular:
  - **Leibniz (1671)**
    - ★ Um dos formuladores do cálculo integral.
    - ★ Projetou a primeira máquina de multiplicação e divisão (além de soma e subtração).
    - ★ Equivalente às calculadoras de bolso que efetuam as quatro operações.
    - ★ Leibniz também descreveu o sistema binário.

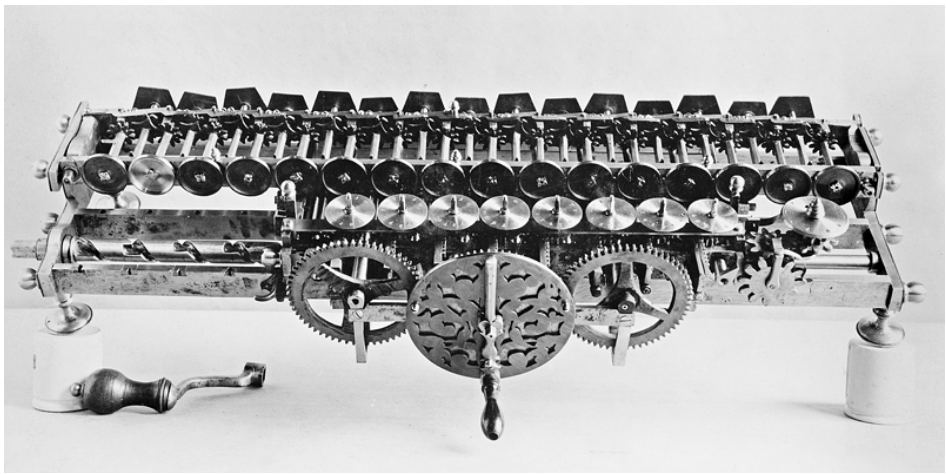


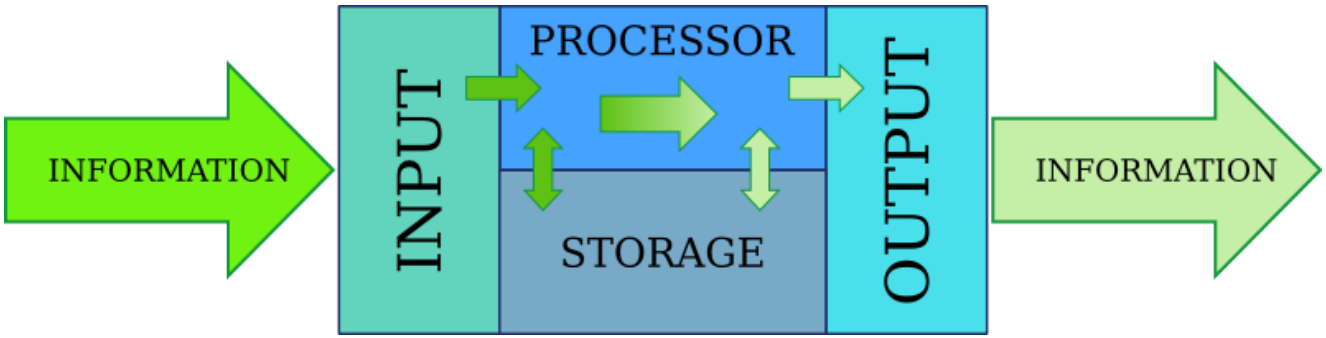
Figure: Máquina original de Leibniz.

Histórico e Evolução dos Computadores

- Calculadora $\times$ Computador
  - Normalmente, uma calculadora realiza um cálculo por vez
  - Computadores são programáveis
    - ★ Possuem suporte a um conjunto de instruções
    - ★ Instruções lidas da memória realizam tarefas
    - ★ Vários cálculos "ao mesmo tempo"

Histórico e Evolução dos Computadores

- Partes principais de um sistema computacional
  - Entrada
  - Processador
  - Memória
  - Saída



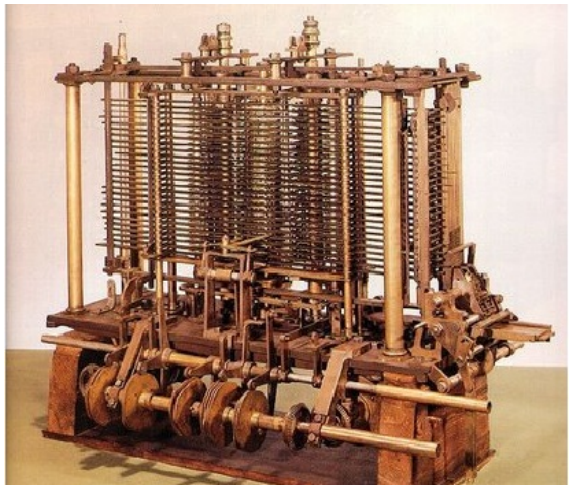
Histórico e Evolução dos Computadores

- Primeiros computadores liam as instruções de cartões perfurados
  - **Jacquard** (1801)
    - ★ Construiu um tear inteiramente automatizado
    - ★ Programado por uma série de cartões perfurados, cada um deles controlando um único movimento da lançadeira

Figure: Mecanismo de Jacquard.

Histórico e Evolução dos Computadores

- "Máquina analítica" de **Babbage** (1834)
  - Máquina mecânica à vapor que utilizava base 10
  - Programação sequencial de operações
  - Anteviu 4 componentes que até hoje são a base do funcionamento de um computador: unidades de entrada, de saída, de memória e de computação
  - Programável em linguagem de montagem simples (software)
  - Primeiros algoritmos escritos por Ada Lovelace
  - As máquinas de Babbage nunca foram construídas



Histórico e Evolução dos Computadores

- Samuel **Morse** (1837)
  - Desenvolvimento de um sistema telegráfico com uso de energia elétrica para transmitir sinais à distância.

- George **Boole** (1854)
  - Concepção dos fundamentos lógicos para a criação de programas: lógica matemática/álgebra booleana.
  - Forma de armazenamento e processamento de dados utilizando relações binárias.

- Herman **Hollerith** (1890)
  - Cartões de Jacquard para **dados** (não apenas instruções) + conceito de impulsos elétricos para transmissão de dados.
  - Tabulador rápido para o processamento de estatísticas.
  - Sistema reconhecido no recenseamento americano de 1890 (redução de 8 para 3 anos).
  - Companhia de Hollerith se tornou a **IBM** (1924).

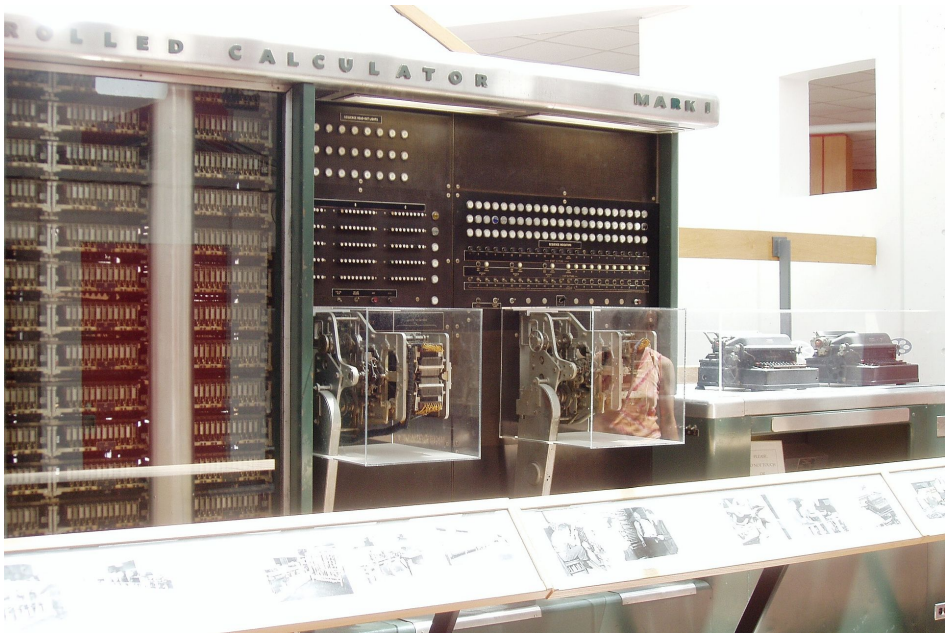
Histórico e Evolução dos Computadores

- Segunda Guerra Mundial aumentou a demanda por máquinas de calcular
  - Cálculo de trajetória de mísseis
  - Quebra de comunicações cifradas

- Zuse (1941)
  - Primeiro computador eletromecânico, constituído de **relés** – efetuava cálculos em binário e exibia os resultados em fita perfurada.
  - Destruído em ataque aliado a Berlim. Réplica no Museu Teconológico:

Histórico e Evolução dos Computadores

- Harvard Mark I (1944)
  - Foi o primeiro computador eletromecânico automático de larga escala
  - Construído em 1944 num projeto da Universidade de Harvard em conjunto com a IBM
  - 15 toneladas
  - 15 anos em serviço
  - Produziu tabelas com fins militares e científicos



As gerações dos computadores

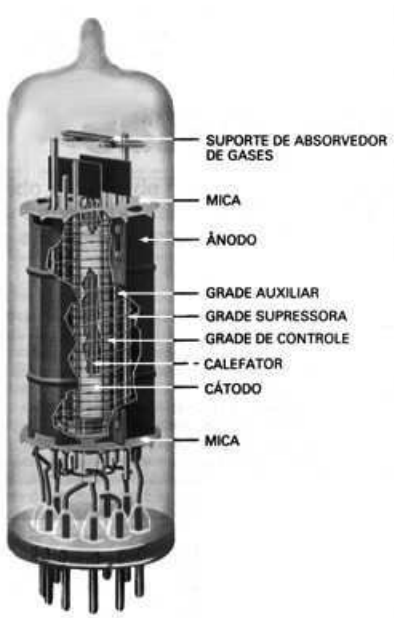
- Os computadores são máquinas capazes de realizar vários cálculos automaticamente, além de possuir dispositivos de armazenamento e de entrada e saída.

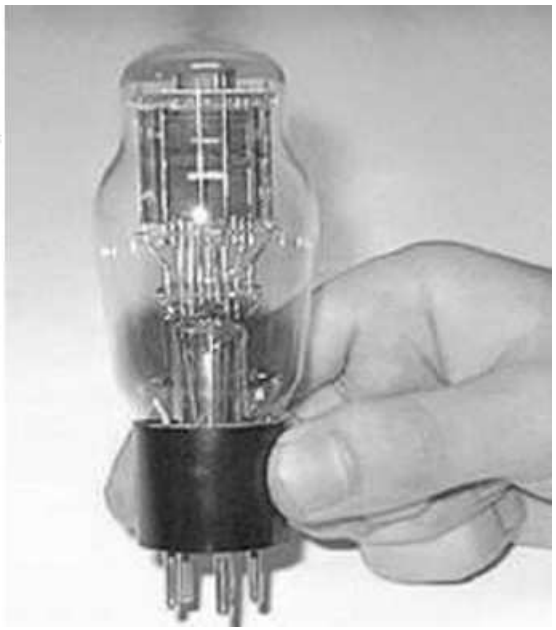
- A evolução dos computadores recebe uma classificação usual em gerações em função das época/tecnologias empregadas.

Primeira geração (1946-1954)

- A primeira geração dos computadores é marcada pela utilização de **válvulas**.
- A válvula é um tubo de vidro, similar a uma lâmpada fechada sem ar em seu interior, ou seja, um ambiente fechado a vácuo, e contendo eletrodos, cuja finalidade é controlar o fluxo de elétrons.
- As válvulas aqueciam bastante e costumavam queimar com facilidade.

Primeira geração (1946-1954)



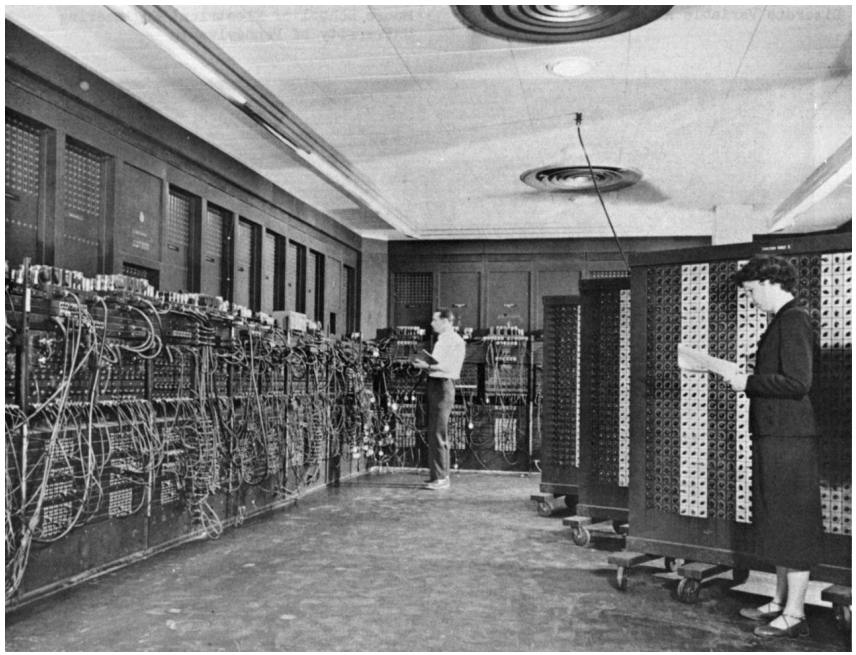


Primeira geração (1946-1954)

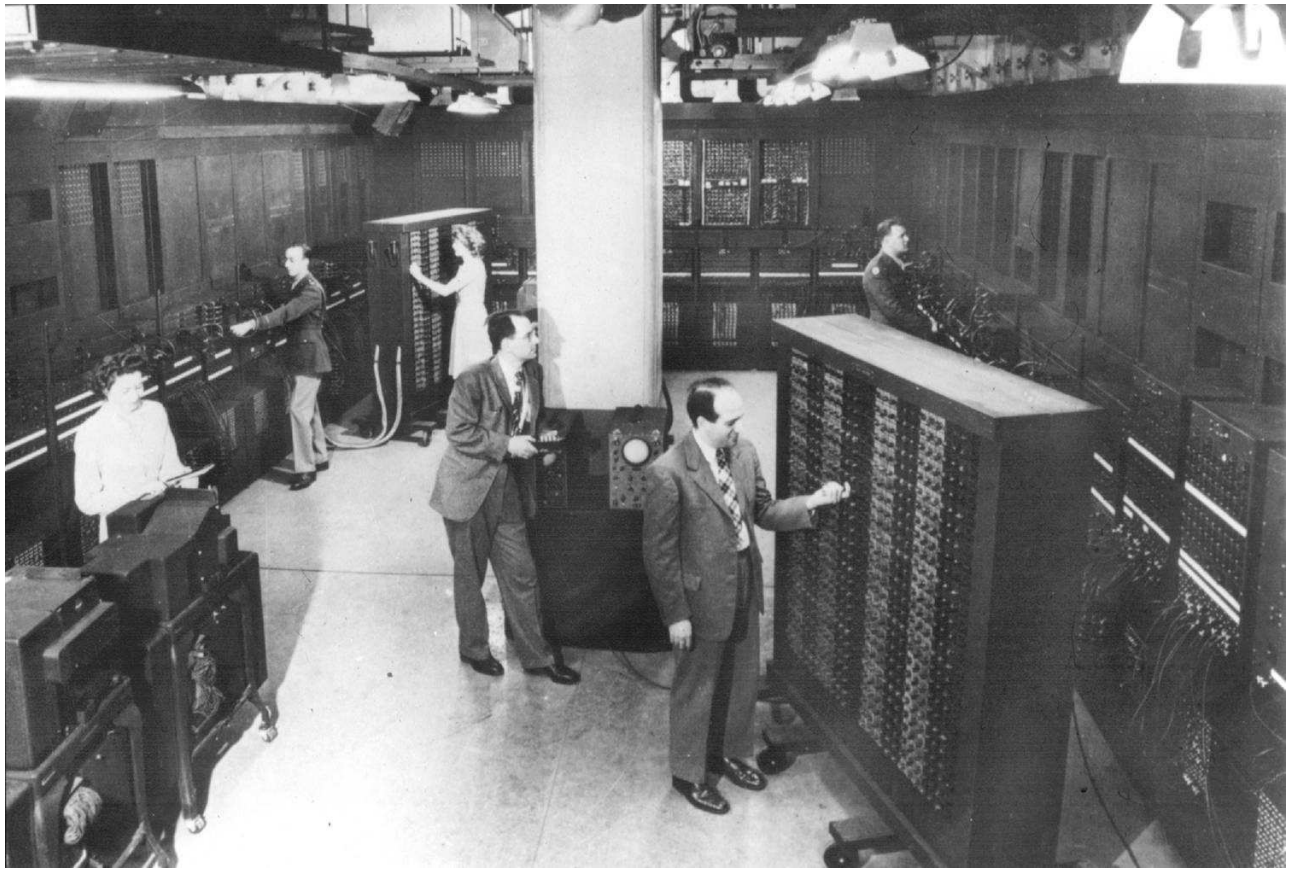
- Programação realizada diretamente na linguagem de máquina. Despendia muito tempo.
- O armazenamento dos dados era realizado em cartões perfurados, que depois passaram a ser feitos em fita magnética.
- Um dos representantes desta geração é o ENIAC. Ele possuía 17.468 válvulas, pesava 30 toneladas, tinha 180m² de área construída, sua velocidade era da ordem de 100 kHz e possuia apenas 200 bits de memória RAM.
- Nenhum dos computadores da primeira geração possuíam aplicação comercial, eram utilizados para fins balísticos, predição climática, cálculos de energia atômica e outros fins científicos.

Histórico e Evolução dos Computadores

- ENIAC (1946) – *Electronic Numerical Integrator and Calculator*
  - Primeiro computador digital eletrônico
  - 18 metros de comprimento por 2,5 metros de largura
  - 18 mil válvulas; 30 toneladas; capacidade para reter em memória 74 números de 23 algarismos
  - 5000 adições ou 300 multiplicações por segundo; consumo de 200 kilowatts; processava 5 mil adições, 357 multiplicações ou 38 divisões por segundo



Primeira geração (1946-1954)

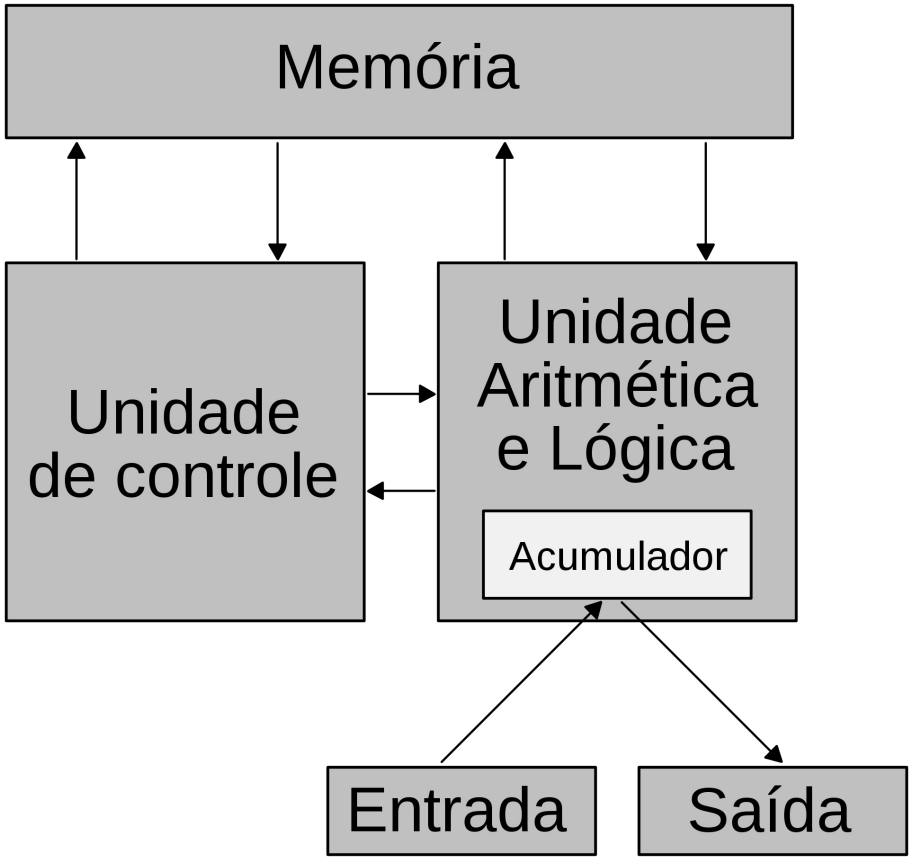


ENIAC, representante da primeira geração dos computadores.

Primeira geração (1946-1954)

- Arquitetura de Von Neumann
  - Arquitetura que seria seguida por todas as gerações de computadores
  - Conceito de programa armazenado: a memória armazenaria tanto as instruções a serem executadas quanto os dados a serem processados
  - Facilidade de modificar instruções e também possibilitou que uma mesma representação armazenasse dados e instruções

Primeira geração (1946-1954)



Histórico e Evolução dos Computadores

- Manchester Mark1 (1948)
  - Primeiro computador a funcionar com um programa armazenado (de acordo com o modelo de Von Newman).
- UNIVAC I (Universal Automatic Computer) (1951)
  - Primeiro computador de uso geral que obteve sucesso comercial.
  - Desenvolvido por Eckert e Mauchy (os mesmos do ENIAC).
  - 5000 válvulas.
  - 1905 operações por segundo.
  - Entrada e saída de dados em fita magnética.
  - Primeiro a contar com unidades de equipamentos periféricos independentes (por exemplo, impressoras)
  - O UNIVAC foi um dos primeiros computadores do Brasil, adquirido pelo IBGE em 1961 por US$ 2.976.350,00, incluídos acessórios e periféricos, para processar dados do censo.

Primeira geração (1946-1954)

Univac 1

Histórico e Evolução dos Computadores

- Mainframes
  - IBM 704, de 1954 (abaixo).
  - Primeiro em produção em larga escala com suporte a ponto flutuante
  - 4 mil multiplicações ou divisões por segundo (o dobro para adições e subtrações)
  - O termo *mainframe* passou a se referir à linha System/360, da IBM, de 1964

Segunda geração (1955-1964)

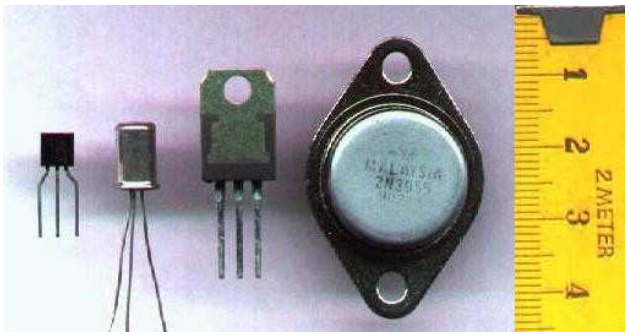
- A segunda geração de computadores foi marcada pela substituição da válvula pelo **transistor**.

- O transistor revolucionou a eletrônica em geral e os computadores em especial.
  - Um transistor é um dispositivo semi-condutor, isto é, conduz corrente elétrica de acordo com a tensão aplicada
  - Pode ser utilizado como uma chave, assim como o relé e a válvula

- Eles eram muito menores do que as válvulas a vácuo e tinham outras vantagens:
  - Não exigiam tempo de pré-aquecimento
  - Consumiam menos energia
  - Geravam menos calor
  - Eram mais rápidos e confiáveis

Segunda geração (1955-1964)

Réplica do primeiro transistor



Segunda geração (1955-1964)

- O tamanho dos computadores diminuiu consideravelmente.
- Desenvolvimento da linguagem assembly (simbólica) em substituição à linguagem de máquina.
- A linguagem assembly possibilita a utilização de *mnemônicos* para representar as instruções de máquina. Exemplo:

```
10110000 01100001
```

agora poderia ser escrito como

```
MOV AL, 61h
```

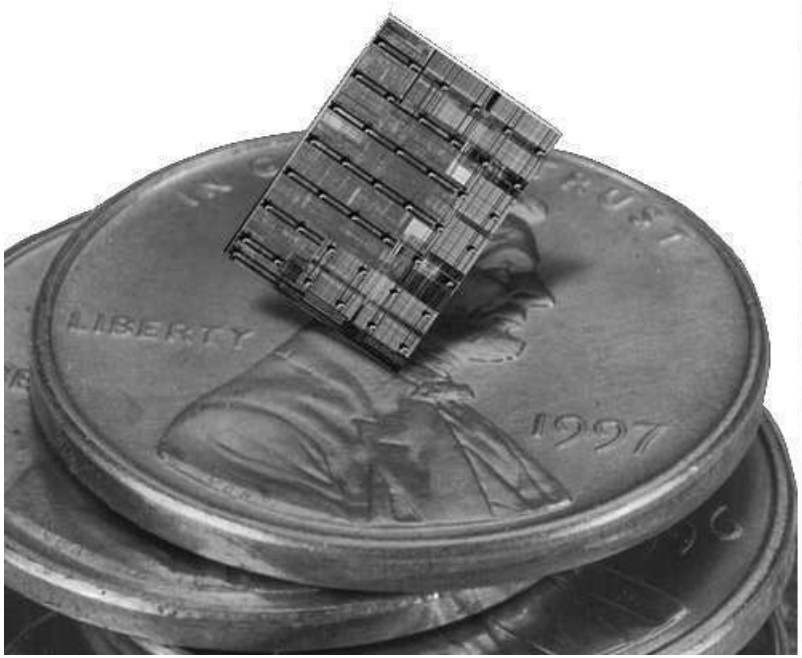
Segunda geração (1955-1964)

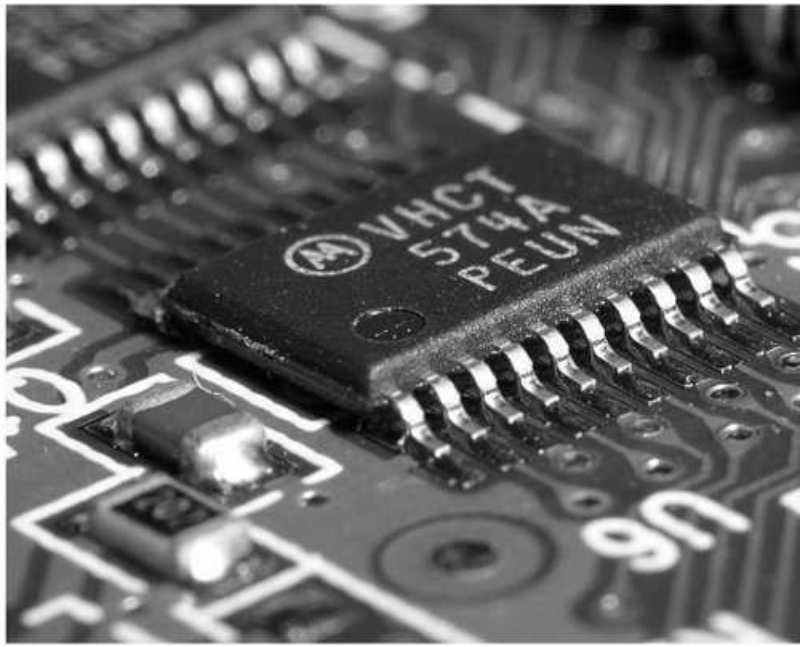
TX-0: primeiro computador transistorizado da história (MIT, 1957)

Terceira geração (1964-1971)

- Marcada pela utilização dos **circuitos integrados**, feitos de silício (ou outro material semicondutor), também conhecidos como **microchips**
- Construídos integrando um grande número de transistores
  - Robustez a interferências elétricas
  - Baixo consumo
  - Equipamentos menores e mais baratos
- O processo de fabricação que possibilitava a construção de vários circuitos simultaneamente, facilitando a produção em massa (algo como o advento da imprensa que revolucionou a produção de livros)

Terceira geração (1964-1971)





Terceira geração (1964-1971)

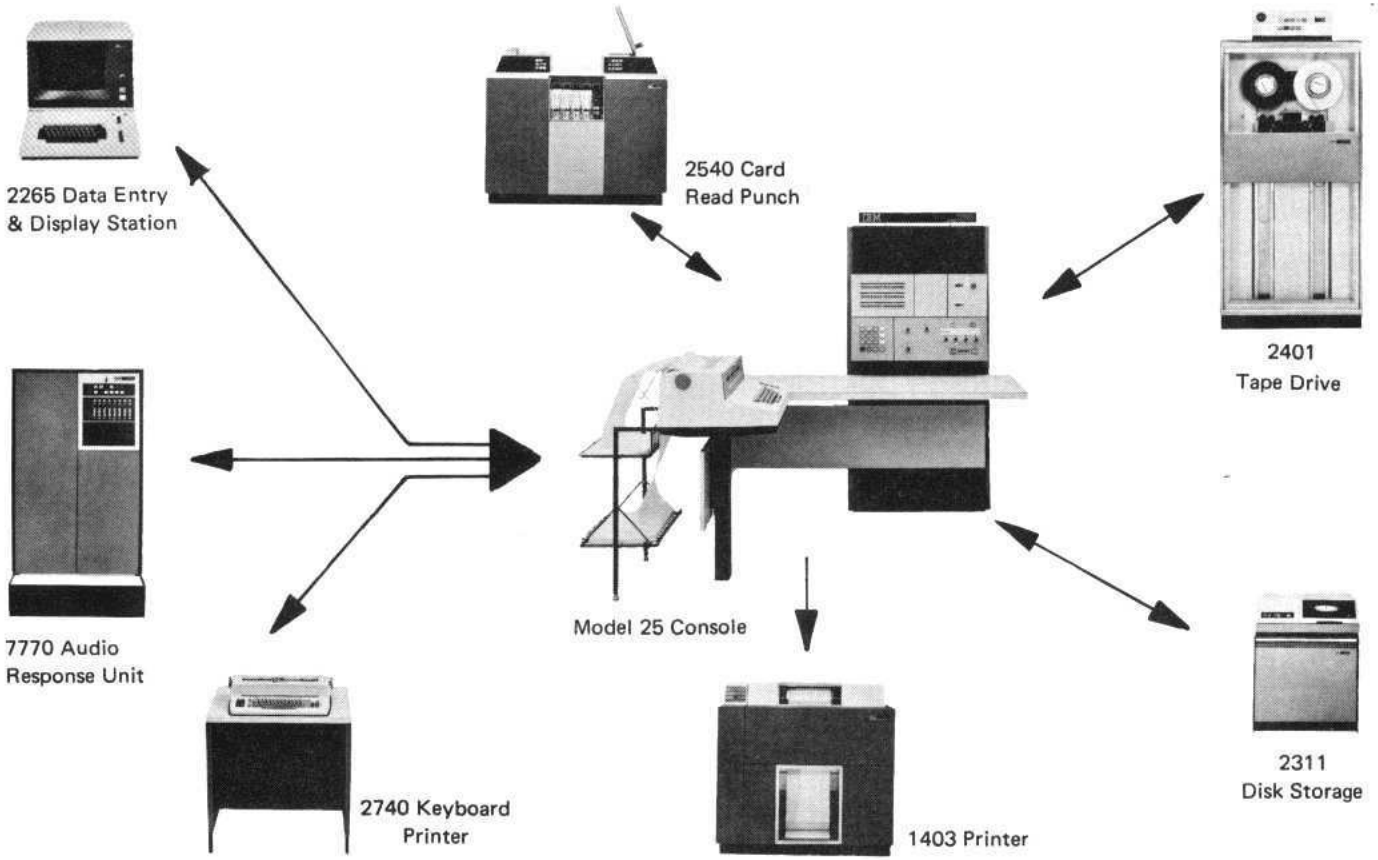


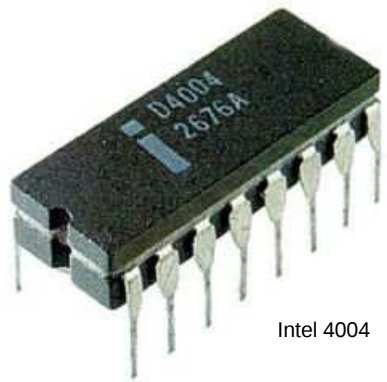
Figure 16. Machine-to-machine communication

Histórico e Evolução dos Computadores

- Minicomputadores
  - PDP 7, da DEC (1965)
  - Tinha um relógio de 0,5 MHz

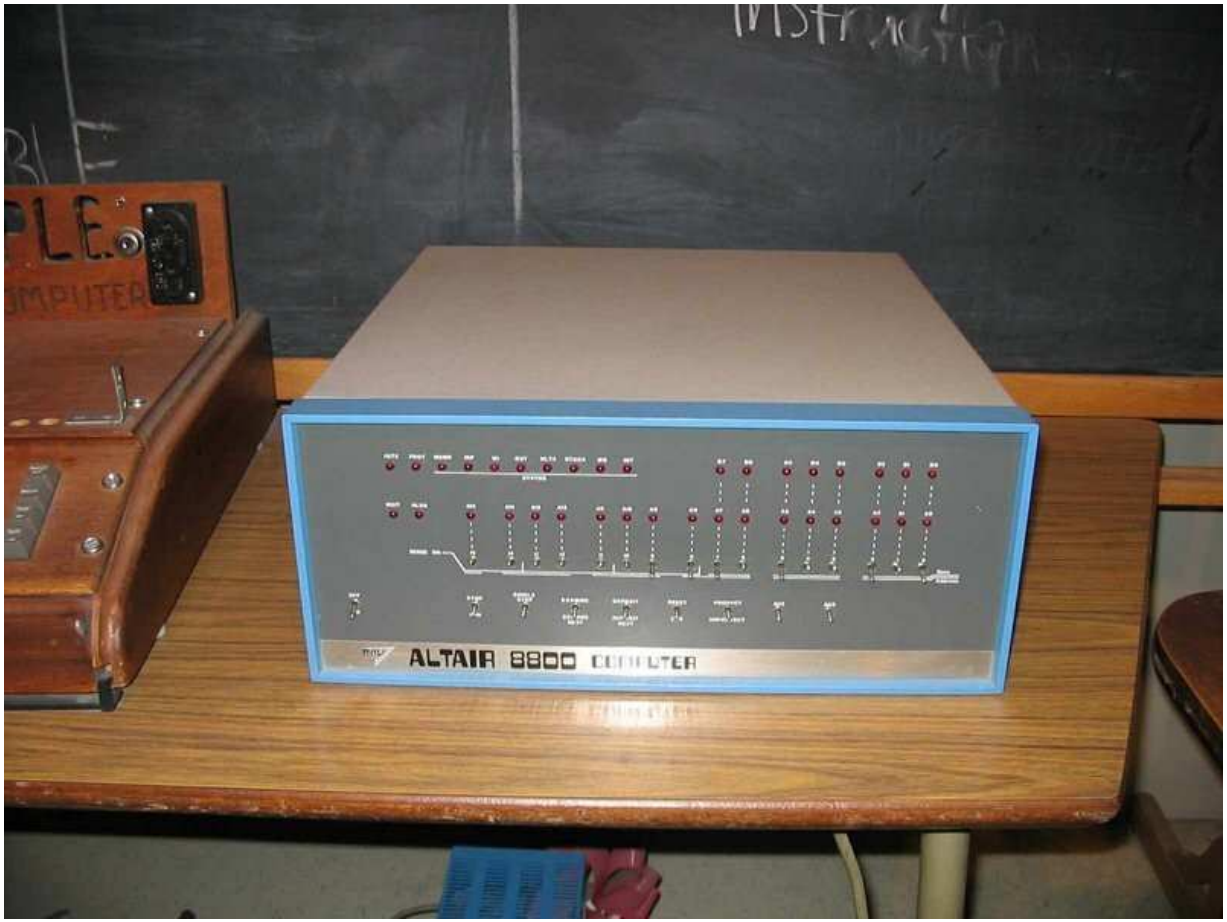
Quarta geração (1971-1991)

- Surgimento dos **microprocessadores** (chip com unidade de controle, unidade lógica-aritmética e uma memória interna com funcionalidades básicas de um computador)
- Computadores mais confiáveis, mais rápidos, menores e com maior capacidade de armazenamento
- Sistemas operacionais como MS-DOS, UNIX,
- Apple's Macintosh foram desenvolvidos
- Discos rígidos eram utilizados como memória secundária
- Impressoras matriciais, e os teclados com os layouts atuais foram criados nesta época



Intel 4004

Quarta geração (1971-1991)



Altair 8800, projetado em 1975, baseado na CPU Intel 8080

Quarta geração (1971-1991)

Apple I (1976)

Quarta geração (1971-1991)

Apple II (1976)

Quarta geração (1971-1991)

Quinta geração (1991-dias atuais)

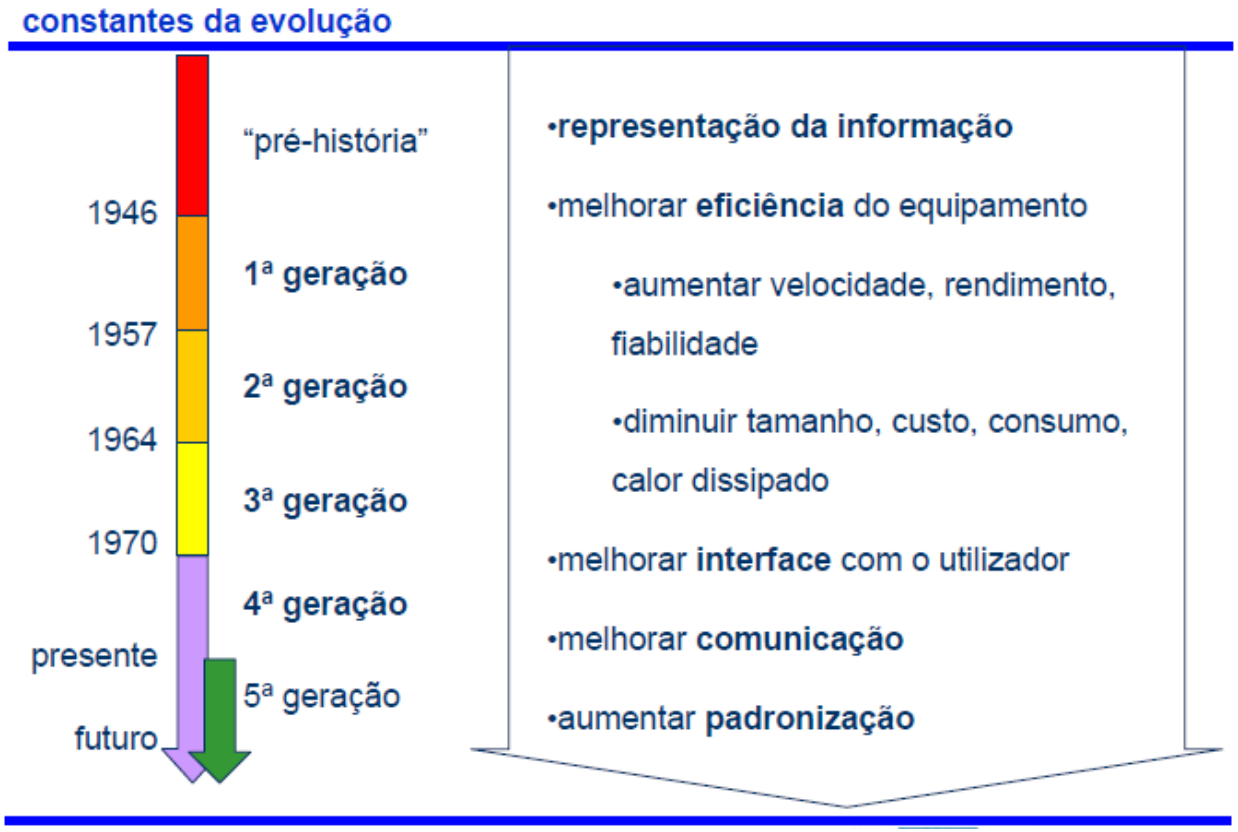
- Processadores com milhões de transistores
- Surgiram as arquiteturas de 64 bits
- Processadores que utilizam tecnologias RISC e CISC
- Discos rígidos com capacidade superior a 600GB
- Pen-drives com mais de 1GB de memória e utilização de disco ótico com mais de 50GB de armazenamento
- Marcada pela inteligência artificial e por sua conectividade

Quinta geração (1991-dias atuais)

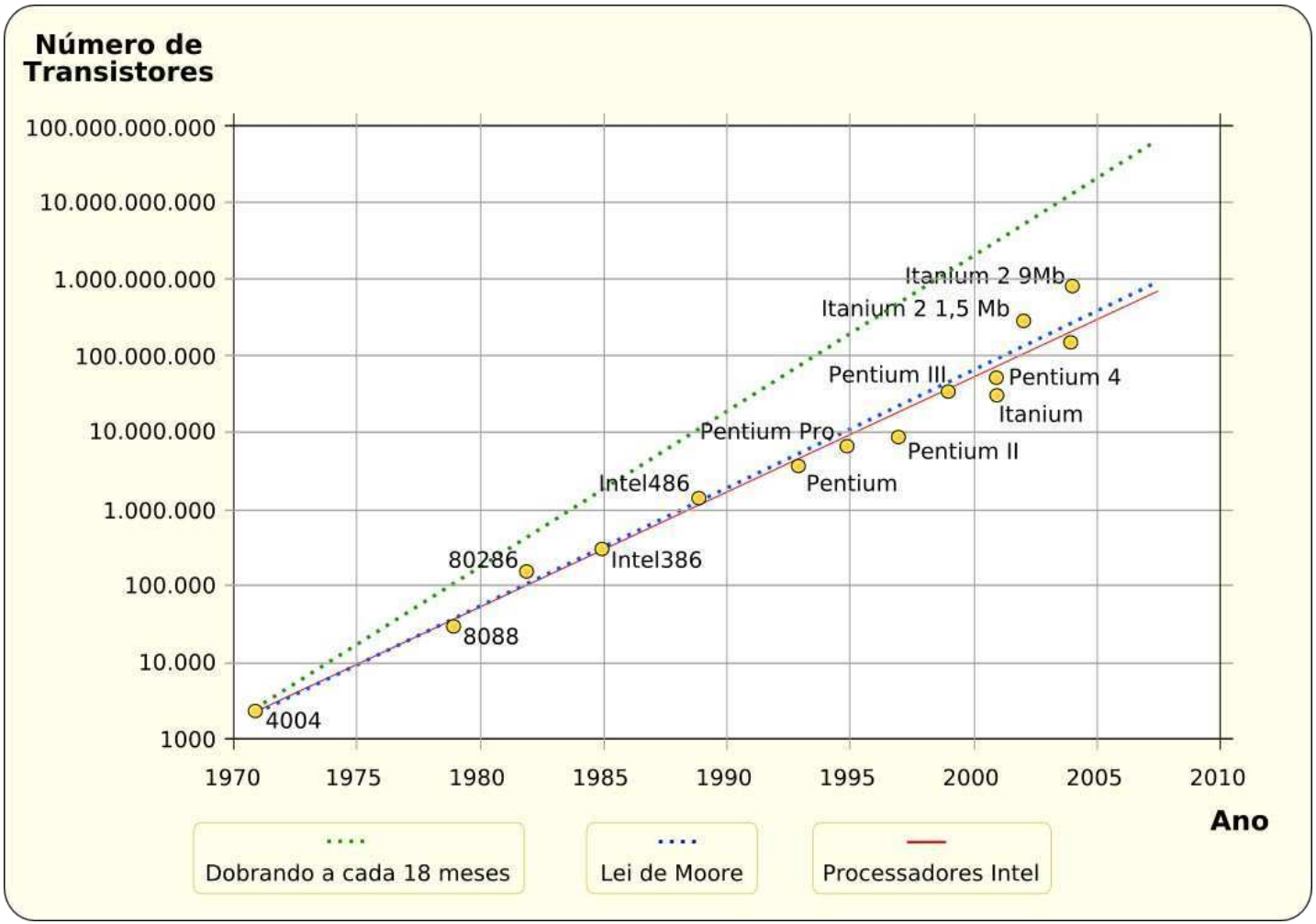
Circuitos integrados

- Categorizados de acordo com a quantidade de integração que eles possuem
  - LSI (*Large Scale Integration* – 100 transistores): computadores da terceira geração
  - VLSI (*Very Large Scale Integration* – 1.000 transistores): computadores da quarta geração
  - ULSI (*Ultra-Large Scale Integration* – milhões de transistores): computadores da quinta geração (1990)
    - Intel Pentium Pro (1996) possuía mais de 6.000.000 de elementos concentrados em poucos centímetros quadrados.
    - AMD Phenom II X4 – 258 mm^2, 758 milhões de transistores de 45 nm
    - Intel Core i7 – 263 mm^2, 731 milhões de transistores de 45 nm

Evolução


constantes da evolução
"pré-história"
1946
1ª geração
1957
2ª geração
1964
3ª geração
1970
4ª geração
presente
5ª geração
futuro
•representação da informação
•melhorar eficiência do equipamento
•aumentar velocidade, rendimento, fiabilidade
•diminuir tamanho, custo, consumo, calor dissipado
•melhorar interface com o utilizador
•melhorar comunicação
•aumentar padronização

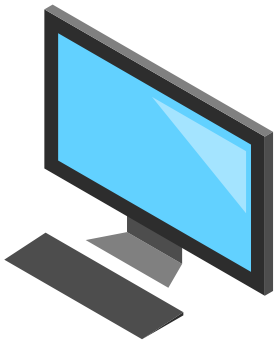
Lei de Moore



Lei de Moore: número de transistores dos chips teria um aumento de 100%, pelo mesmo custo, a cada período de 18 meses

Histórico e Evolução dos Computadores

- Evolução tecnológica
- Memórias maiores e mais baratas
- Processadores menores e mais rápidos




Histórico e Evolução dos Computadores

- Leitura
  - Capítulo 1 do Livro *Introdução à Computação* de Gilberto Farias: http://producao.virtual.ufpb.br/books/camyle/introducao-a-computacao-livro/livro/livro.chunked/
  - Museu Virtual Informática http://museuvirtualutfpr.blogspot.com.br/
  - Livro do Prof. Raul Sidnei Wazlawick – História da Computação (1ª ed.)
- Filmes sugeridos
  - Sobre o início: *O Jogo da Imitação* (2014), *Estrela Além do Tempo (Hidden Figures)* (2016)
  - Sobre a atualidade: *Piratas do Vale do Silício* (1999)
  - Sobre o futuro (recente): *Ela* (2013)
  - Outros: http://olhardigital.uol.com.br/noticia/11-filmes-sobre-tecnologia-que-merecem-sua-atencao/43621

Linguagens

- Dificuldade em montar programa diretamente no conjunto de instruções do processador.
- Assembly foi criado para facilitar a montagem do programa (assembler = montador).
- Linguagem de alto nível $\rightarrow$ código objeto $\rightarrow$ montador.
- Solução: Compiladores!

Assembly

rótulo: mnemônico argumento1, argumento2, argumento3 ...

- O rótulo é um identificador seguido de dois pontos.
- O mnemônico é uma palavra reservada para o código da instrução do processador.
- Os operadores "argumento" são opcionais e podem ser em número de 0 a 3, dependendo do código do processador.
- Exemplo:
  - `carr_reg:  MOV  AL, 61h  ⇒  10110000 01100001`

FORTRAN

- FORmula TRANslator – 1954-1958
- Procedural e imperativa
- Criada pela IBM (John Backus)
- Dedicada à resolução de equações e fórmulas matemáticas
- FORTRAN II: laços, funções, sub-rotinas e a primitiva do comando FOR
- Sugestão de filme: Estrelas Além do Tempo (*Hidden Figures*), 2016

FORTRAN (exemplos)

- FORTRAN90

```
nfatorial = PRODUCT((/(i, i=1,n)/))
```

- FORTRAN77

```
FUNCTION FAT(N)
      INTEGER N,I,FAT
      FACT=1
      DO 10 I=1,N
   10 FAT=FAT*I
      END
```

LISP

- LISt Processor – 1958-1960
- Funcional
- Criada por McCarthy
- Desenvolvida para processamento de listas
- Puramente recursiva e não iterativa
- Não diferencia código e dados

LISP (exemplo)

```
;; Programa fatorial em Lisp
(defun fatorial (n)
    (if (<= n 1)
        1 (* n (fatorial (- n 1)))
    )
)
```

ALGOL

- ALGOrithmic Language – 1958-1968
- Procedural, criada em 58 como IAL (International Algorithmic Language)
- Criada por comite de especialistas em computação
- Primeira linguagem autônoma, independente de arquitetura (portável)
- Introduziu a declaração em blocos e variáveis locais, arrays dinâmicos, := para atribuição, laços, IF...THEN...ELSE, FOR, SWITCH, WHILE ALGOL 68 define cast de tipos e UNION.

ALGOL (exemplo)

```
integer procedure Fatorial(m); integer m;
begin
    integer F;
    F := if m=1 then 1 else m*Fatorial(m-1);
    Fatorial := F
end
```

COBOL

- COmmon Business Oriented Language – 1959
- Linguagem orientada para negócios e processamento de banco de dados comerciais
- Criado pelo Departamento de Defesa Norte-Americano sob direção de Grace Murray Hopper

COBOL (exemplo)

```
IDENTIFICATION DIVISION.
FUNCTION-ID. fatorial.

DATA DIVISION.
LOCAL-STORAGE SECTION.
01  i       PIC 9(10).

LINKAGE SECTION.
01  n       PIC 9(10).
01  ret     PIC 9(10).

PROCEDURE DIVISION USING BY VALUE n RETURNING ret.
    MOVE 1 TO ret

    PERFORM VARYING i FROM 2 BY 1 UNTIL n < i
        MULTIPLY i BY ret
    END-PERFORM

    GOBACK.
```

BASIC

- Beginers All-purposes Symbolic Instruction Code – 1963-1964
- Procedural
- Criada por John Kemeny e Thomas Kurtz (não pelo Bill Gates como citam algumas fontes).
- Originalmente, código de linhas numeradas e subrotinas chamadas por linha (GOTO e GOSUB)

BASIC (exemplo)

```
10 LET x=5: GO SUB 1000: PRINT "5! = ";r
999 STOP
1000 REM ************
1001 REM * FATORIAL *
1002 REM ************
1010 LET r=1
1020 IF x<2 THEN RETURN
1030 FOR i=2 TO x: LET r=r*i: NEXT i
1040 RETURN
```

C

- Imperativa, procedural, de propósito geral – 1969-1973
- Da linhagem do ALGOL, desenvolvida originalmente por Denis Ritchie.
- Destinada à programação de sistemas Unix, a partir do BCPL (*Basic Combined Programming Language*, 1965) e B (contração de BCPL, 1967) desenvolvidas pela Bell Labs.
- Padronizada em 1973 (ANSI C).
- Conceito de blocos, bibliotecas (headers) de funções, array, pointers e casting de tipos.
- É a linguagem mais utilizada até hoje.

C (exemplo)

```
int fatorial(int n) {
    int result = 1;
    for (int i = 1; i <= n; ++i)
        result *= i;
    return result;
}
```

HASKELL

- Nome em homenagem a Haskell Curry – 1990-1998
- Funcional (diretamente derivada da ML)
- Criada pela Universidade de Glasgow
- Linguagem puramente funcional e baseada em cálculo lambda tipado

HASKELL (exemplo)

```
-- Fatorial em Haskell
module Main () where

main = print (fatorial 20)

fatorial 0 = 1
fatorial n = n * fatorial (n-1)
```

Python

- Multiparadigma (orientada a objetos, imperativa, funcional) – 1991
- Nome em homenagem ao grupo humorístico e não à cobra
- Criada por Guido van Rossum
- Interpretada
- Tipagem dinâmica

Python (exemplo)

```
def factorial(n):
    result = 1
    for i in range(1, n+1):
        result *= i
    return result
```

Java

- Green Project (convergência entre computadores e eletrodomésticos) da Sun Microsystems – 1991
- *7 (StarSeven) - controle remoto com interface toutchscreen (mascote Duke) – 1992
- Oak (carvalho) como nova especificação por James Gosling – vídeo por demanda e programas interativos (muito cedo para época)
- Java consistiu numa adaptação do Oak para a internet – 1995
  - Projetada para se mover por redes de dispositivos heterogêneos
  - Aplicações executadas nos navegadores (Applets Java)
- Orientada a objetos, portabilidade, recursos de rede, segurança
- Sintaxe similar a C/C++, Unicode nativo
- Vasto conjunto de bibliotecas (API)
- Facilidade para programas distribuídos e multitarefas
- Desalocação de memória automática por processo de coletor de lixo

Java (exemplo)

```
public static long fatorial(final int n) {
    if (n < 0) {
        System.err.println("No negative numbers");
        return 0;
    }
    long ans = 1;
    for (int i = 1; i <= n; i++) {
        ans *= i;
    }
    return ans;
}
```

Décadas 1970-2000

- Várias linguagens derivadas das anteriores
  - **Imperativas:** Forth - Modula 2 - Perl - PHP - Javascript - C# ...
  - **Orientada a objetos:** Smalltalk - ADA - C++ - Eiffel - Ruby ...
  - **Declarativas:** Prolog - SQL - HTML - UML - XML - Scheme - ML - Miranda - Haskell - O'Haskell (OOP) - Clean - CAML - OCAML (OOP) - UML ...

- Objetivo de todas: gerar código em mais baixo nível para a instrução da máquina.

Genealogia das linguagens