



Instituto Federal do Ceará - IFCE

Campus Aracati

Disciplina: Arquitetura de Computadores

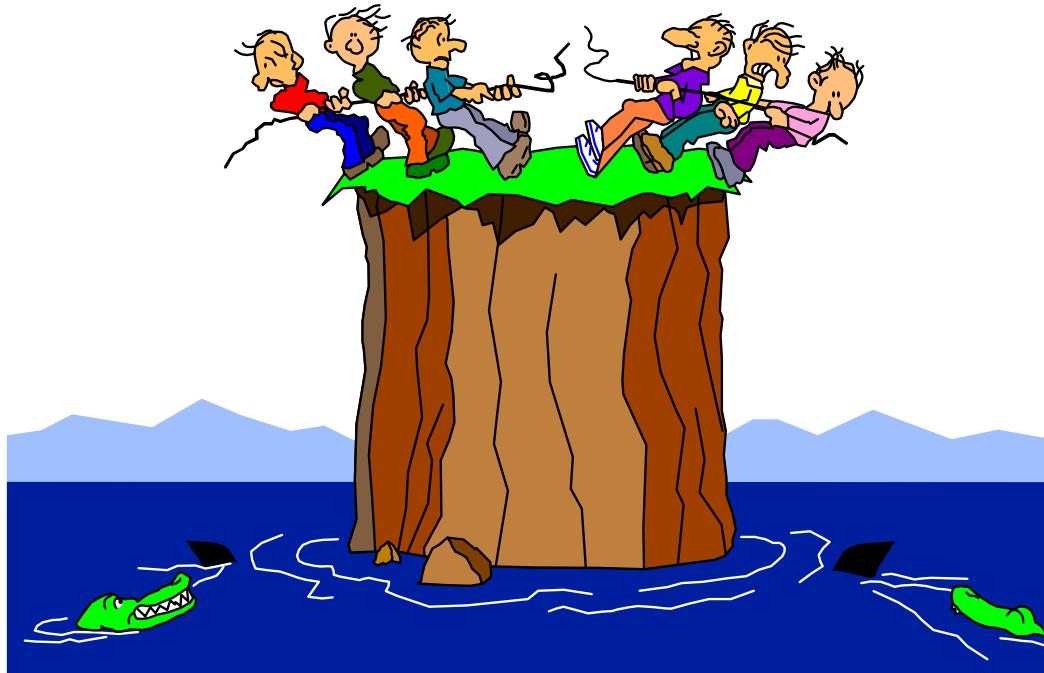
*Chico Anysio*  
Óleo sobre tela 60x40 cm

# PROGRAMA

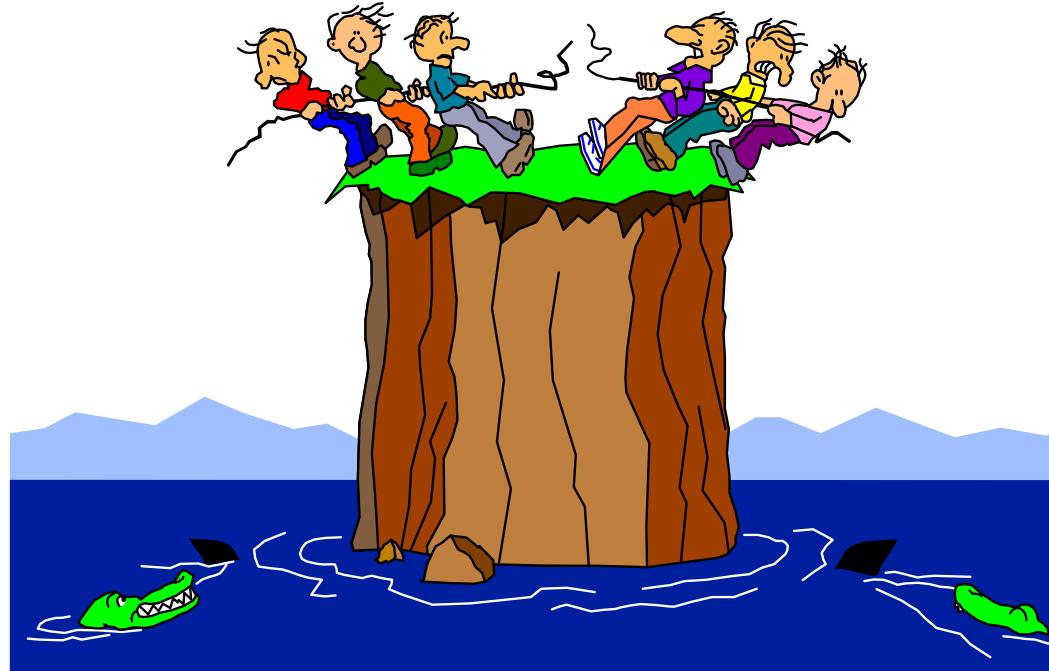
1. Modelo Von Neumann
2. Funcionamento de um computador.
3. Circuitos Combinacionais e Sequenciais.
4. Representação de dados.
5. Arquitetura clássica de um computador.
6. Memória e seus mecanismos de gerenciamento;
7. Evolução das Arquiteturas: x86, x64, etc.
8. Arquiteturas com processadores modernos.
9. Arquiteturas RISC, CISC, Pipeline.
10. Arquiteturas Paralelas e não Convencionais.

# Modelo de Von Neumann

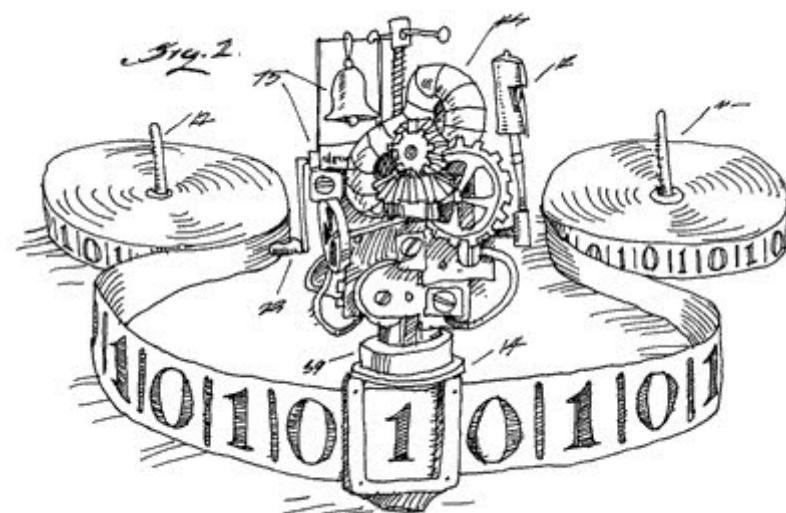
## Aula 1



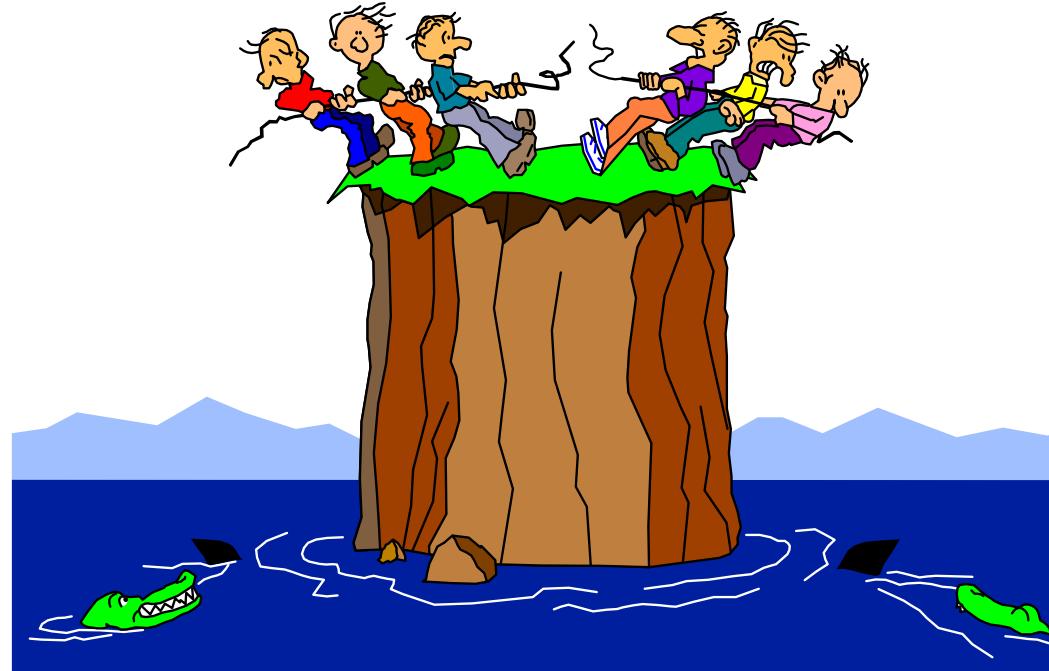
# 1.1 Alan Turing

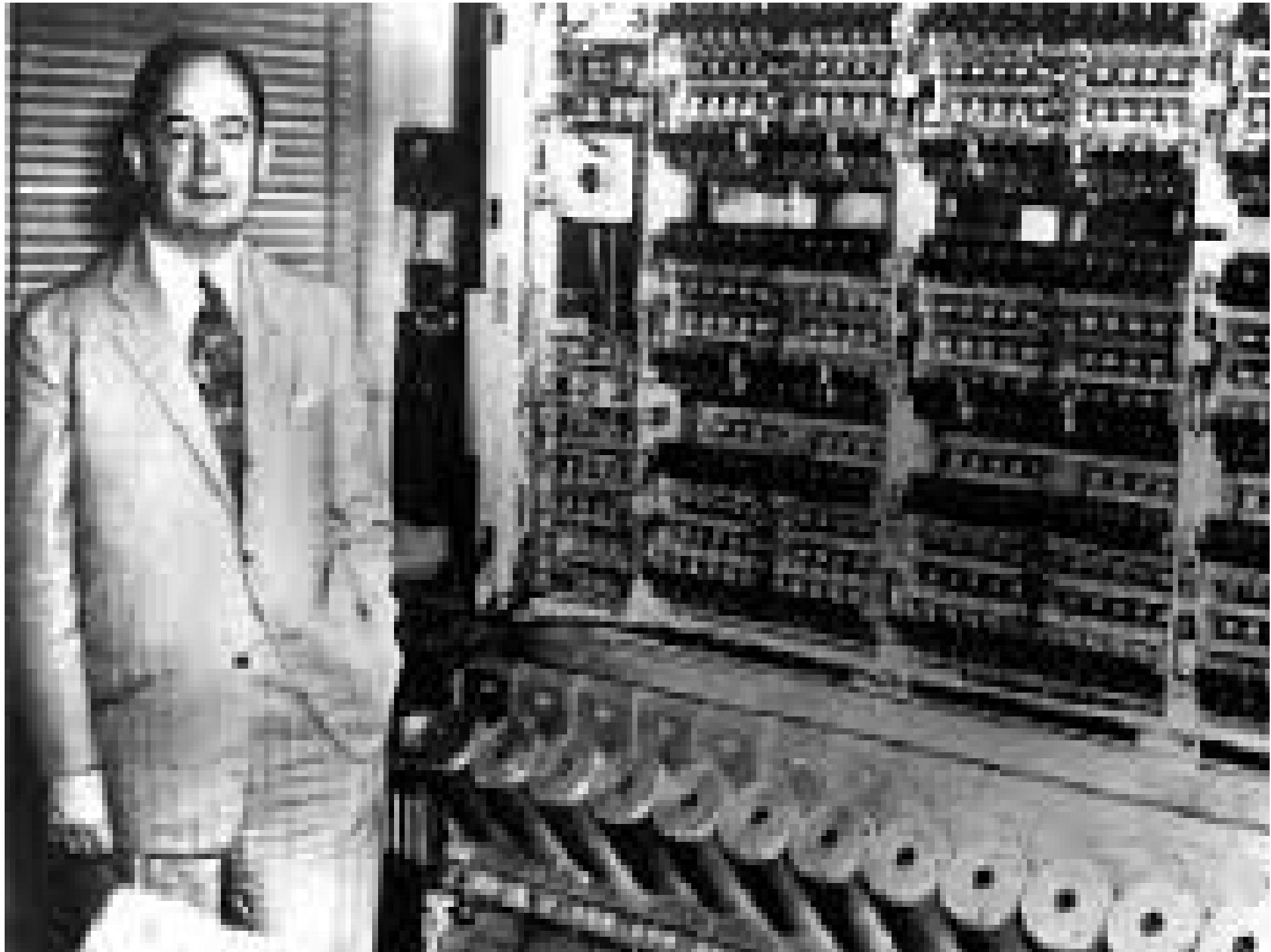


# 1.1 Alan Turing



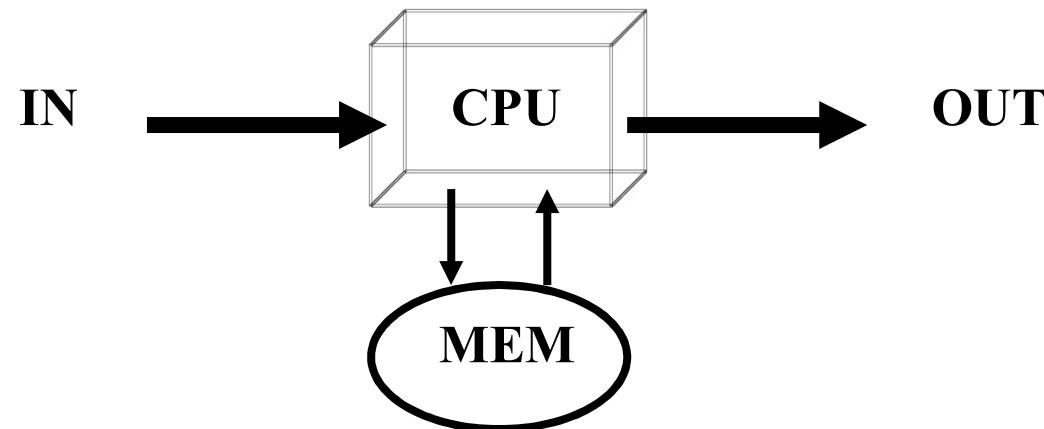
## 1.2 Modelo de Von Neumann





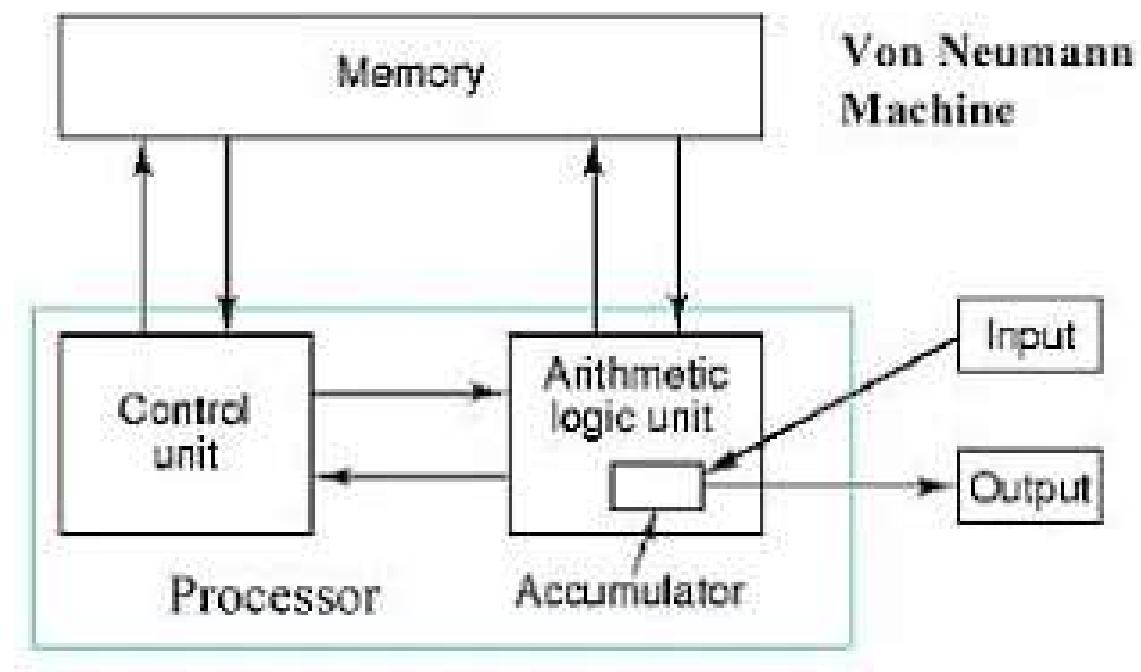
# Modelo de Von Neumann

- CARACTERÍSTICA PRINCIPAL
  - Conceito de Programa Armazenado



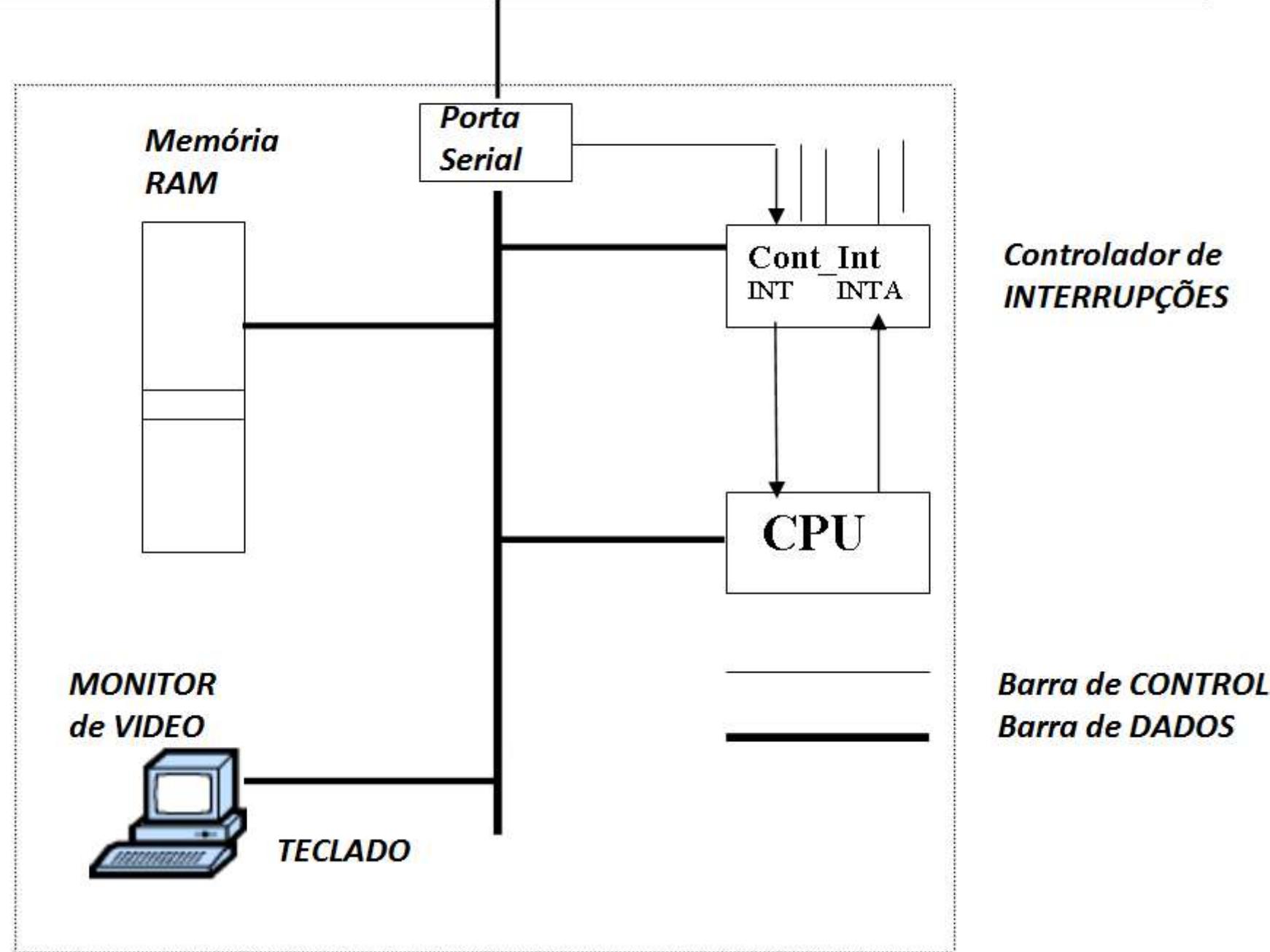
# Modelo de Von Neumann

- Conceito de Programa Armazenado
- BUSCA / DECODIFICA / EXECUTA



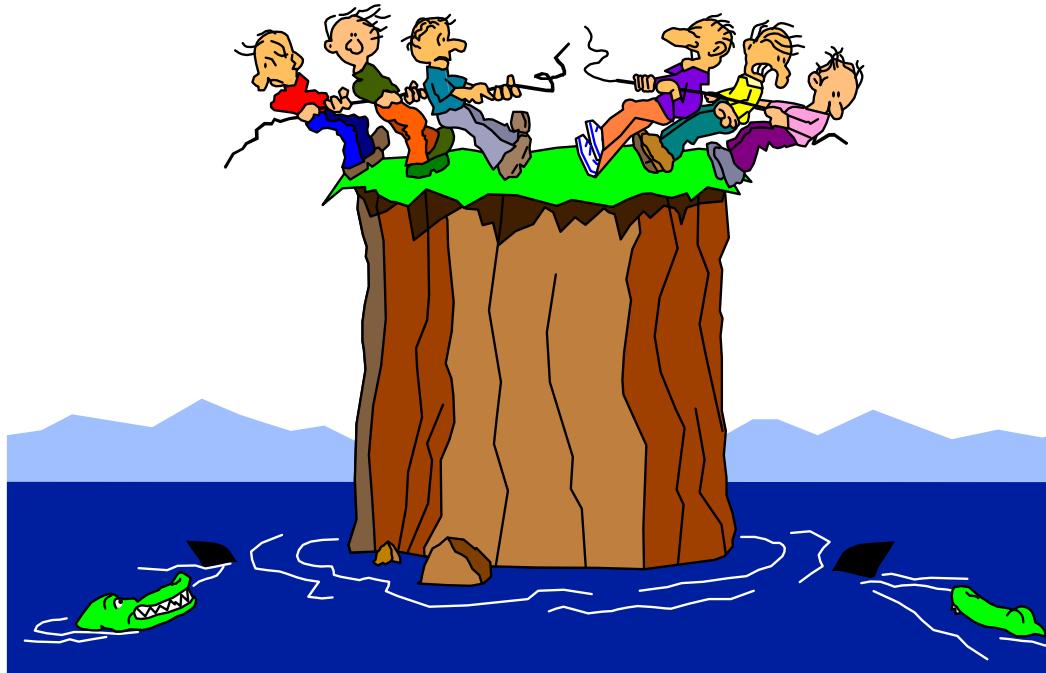
«Caractère : ‘a’ »

‘0 1 0 0 0 0 0 1’



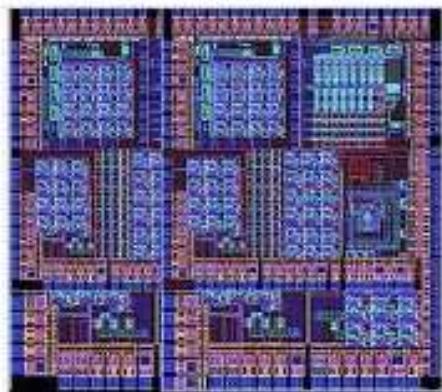
# Funcionamento do Computador

## Aula 2



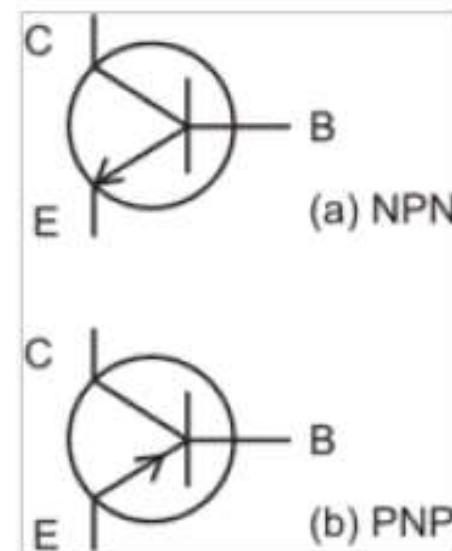
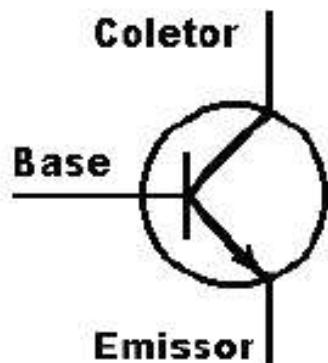
# Funcionamento dos circuitos lógicos

## O Transistor e o Circuito Integrado

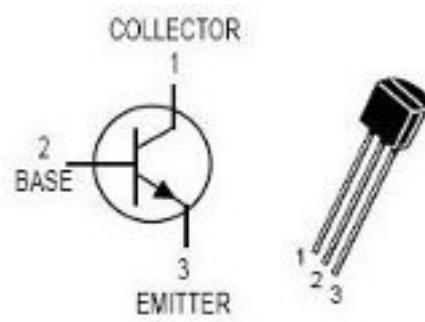


# Funcionamento dos circuitos lógicos

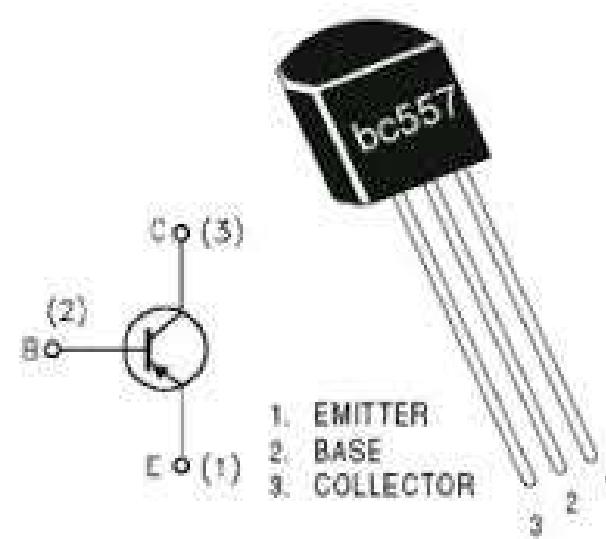
## O Transistor



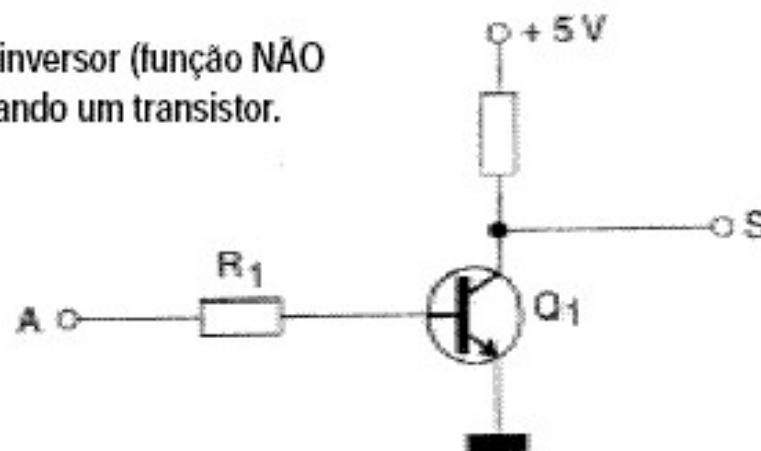
O transistor foi desenvolvido em 1948 nos laboratórios da Bell Telephone, nos EUA, pelos físicos Walter Brattain, John Bardeen e William Shockley, que receberam o Prêmio Nobel de 1956 pela descoberta.



BC548



**Figura 1 - Um inverter (função NÃO ou NOT) usando um transistor.**

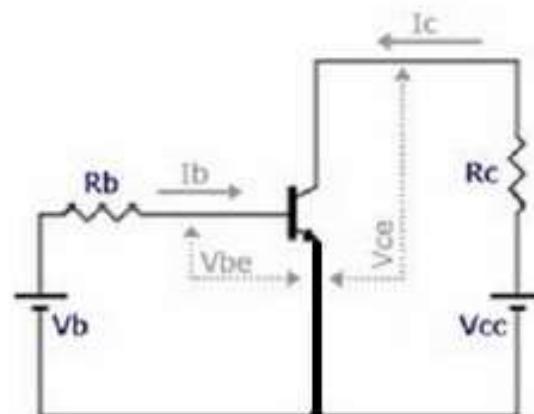
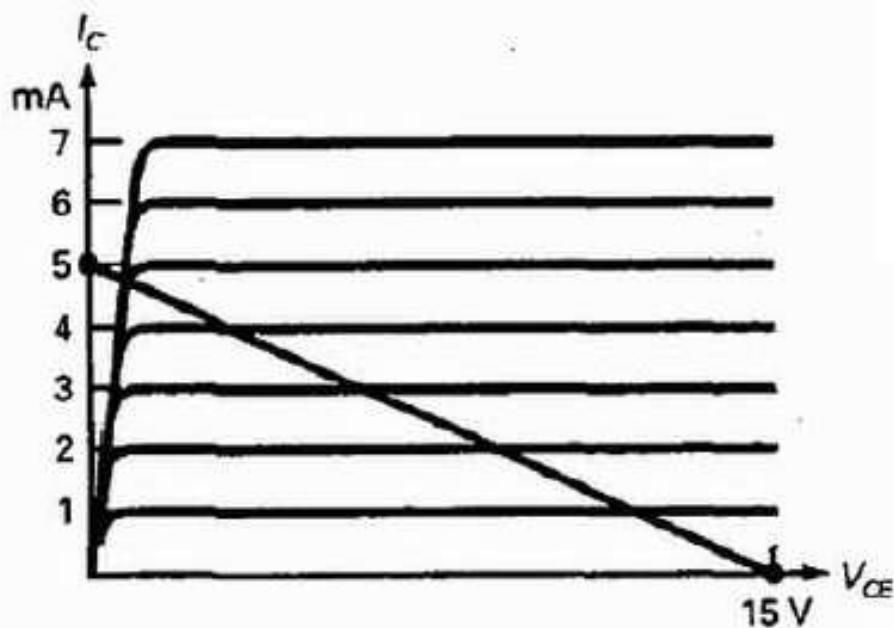


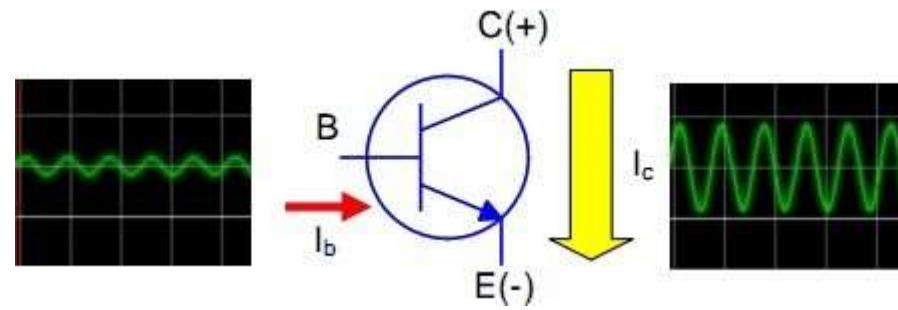
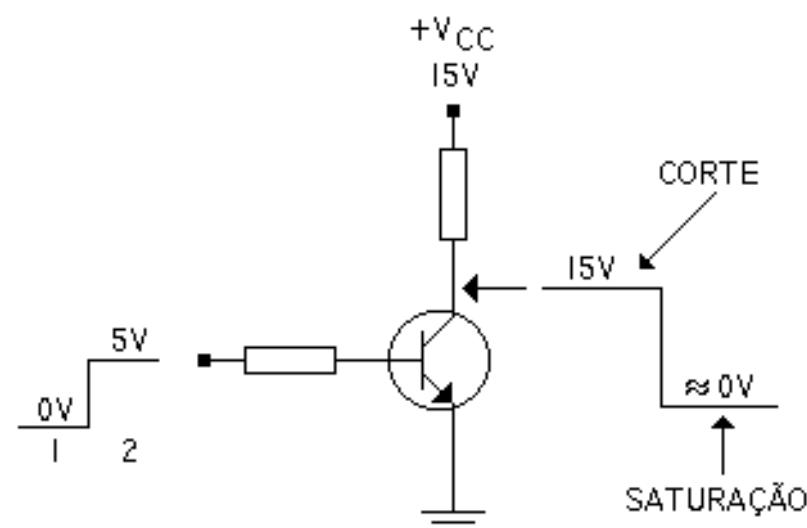
E	S
0	5V
5V	0

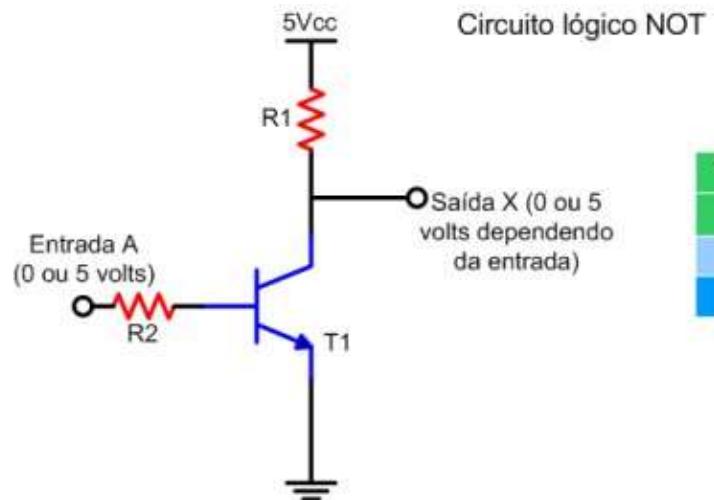
# Funcionamento dos circuitos lógicos

## O Transistor

$$V_{cc} = R_c I_c + V_{ce}$$

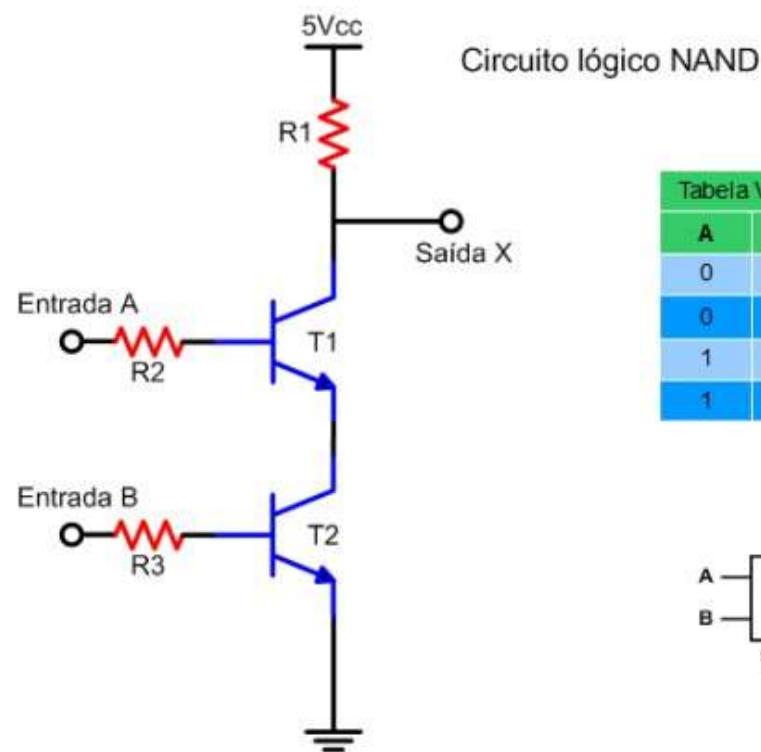
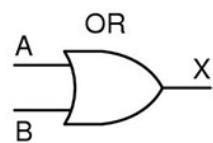
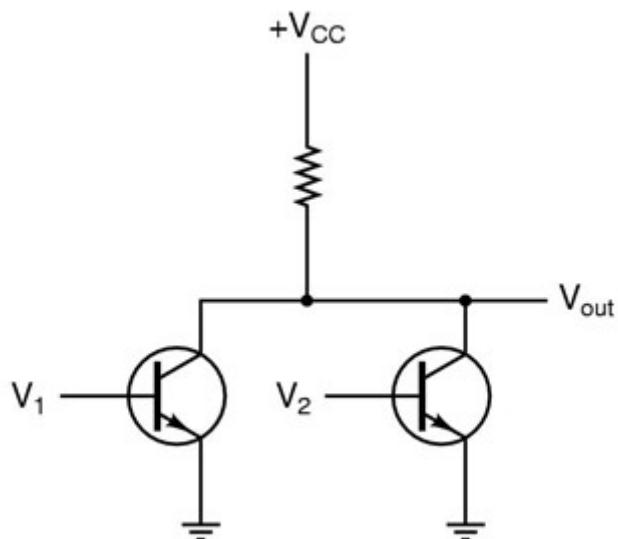
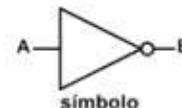






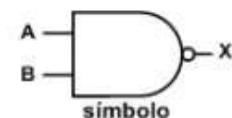
**Tabela Verdade NOT**

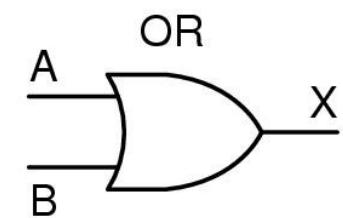
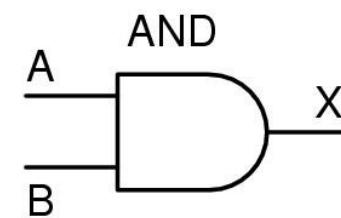
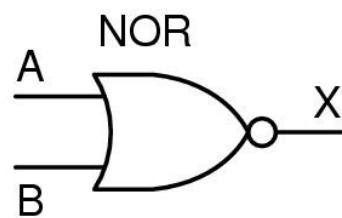
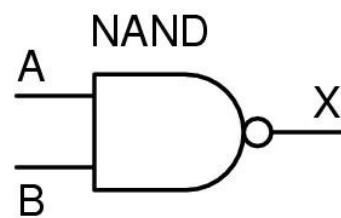
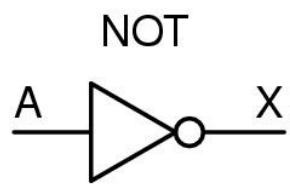
A	X
0	1
1	0



**Tabela Verdade NAND**

A	B	X
0	0	1
0	1	1
1	0	1
1	1	0





A	X
0	1
1	0

(a)

A	B	X
0	0	1
0	1	1
1	0	1
1	1	0

(b)

A	B	X
0	0	1
0	1	0
1	0	0
1	1	0

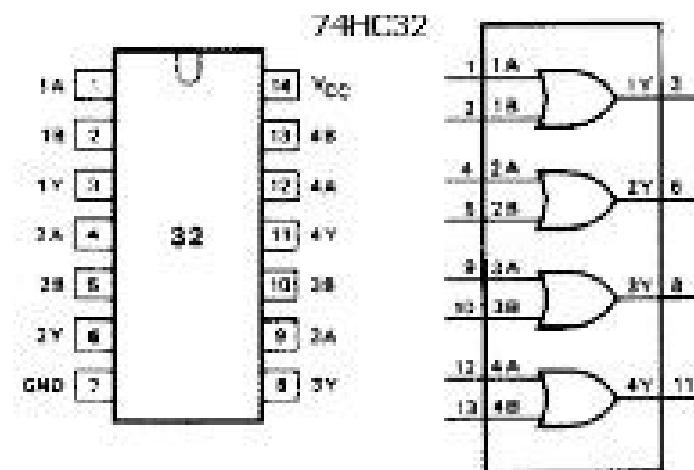
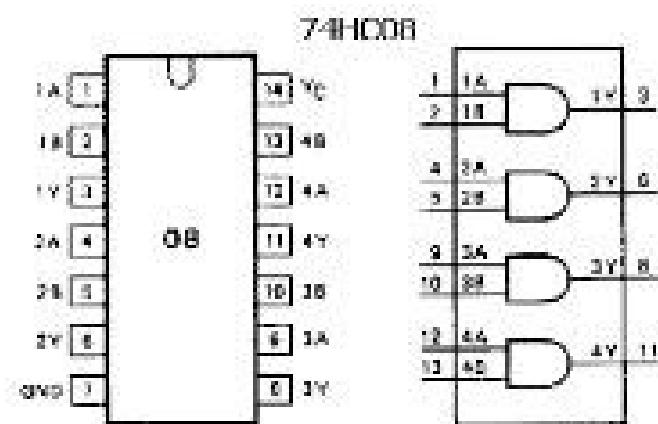
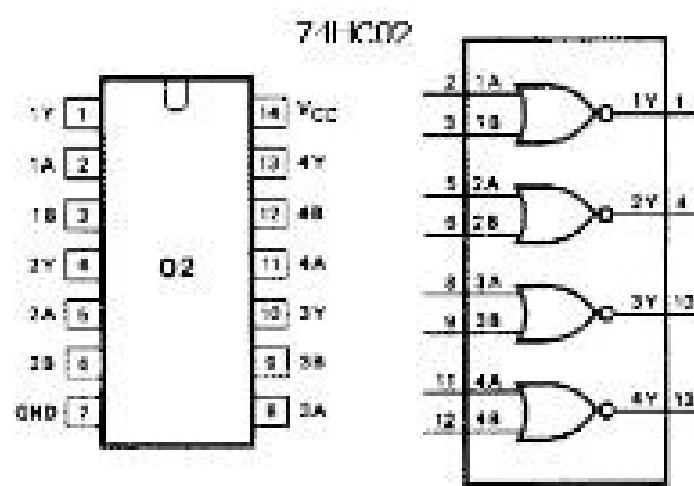
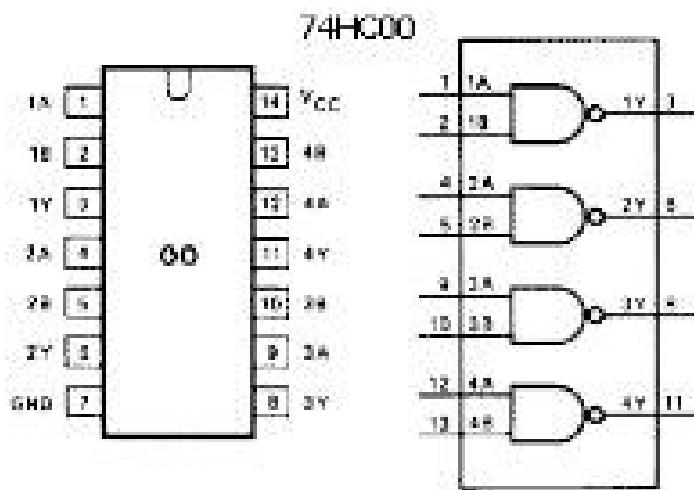
(c)

A	B	X
0	0	0
0	1	0
1	0	0
1	1	1

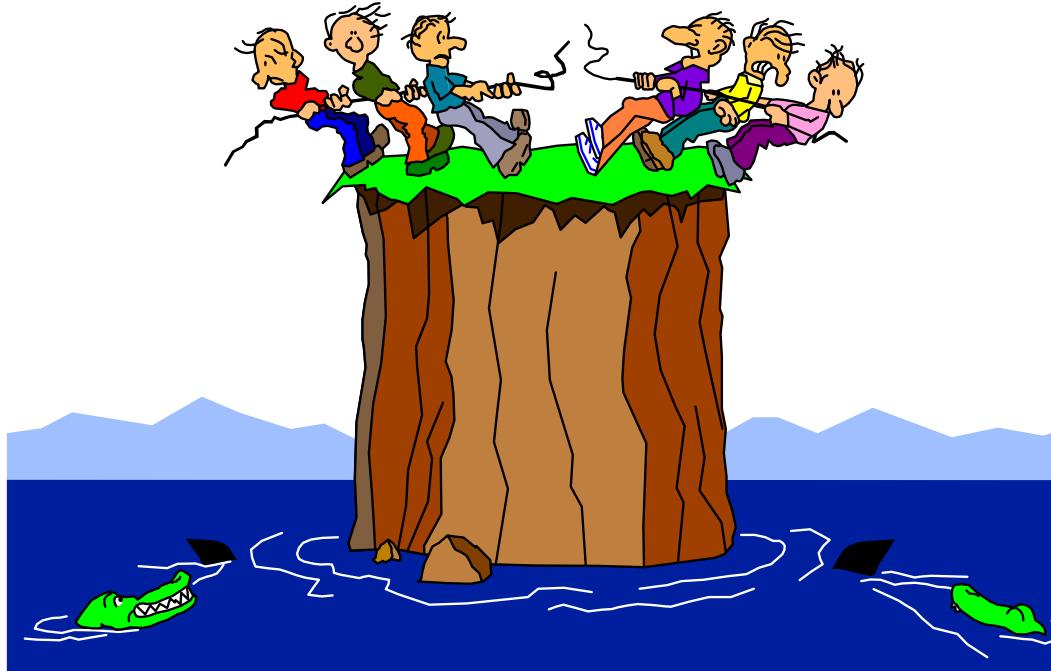
(d)

A	B	X
0	0	0
0	1	1
1	0	1
1	1	1

(e)



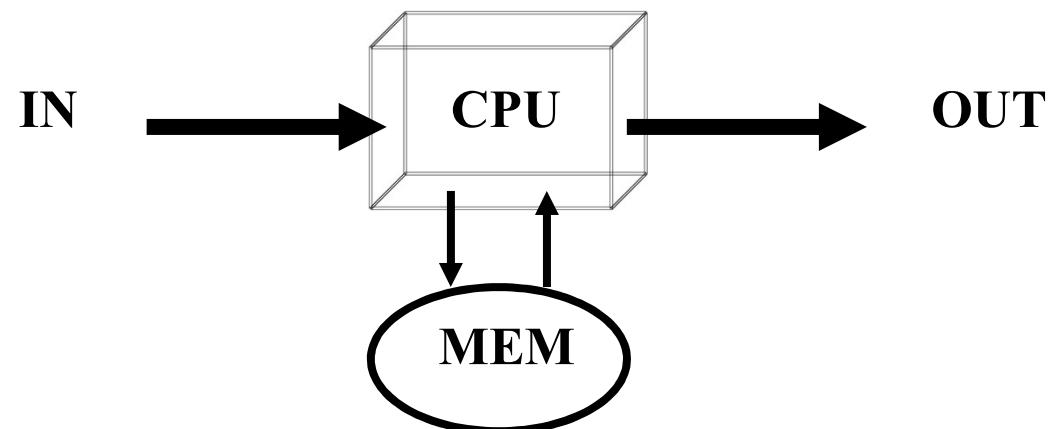
## 1.2 Sistemas Centralizados



# Modelo de Von Neumann

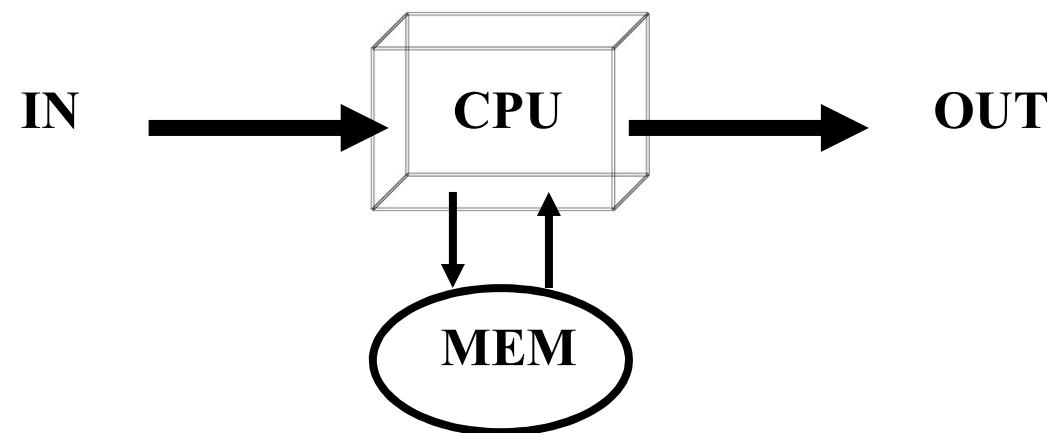
- RESTRIÇÕES DO MODELO**

- Linguagem de Baixo Nível**
- Memória Linear e Uniforme**
- Processamento Seqüencial e Centralizado**

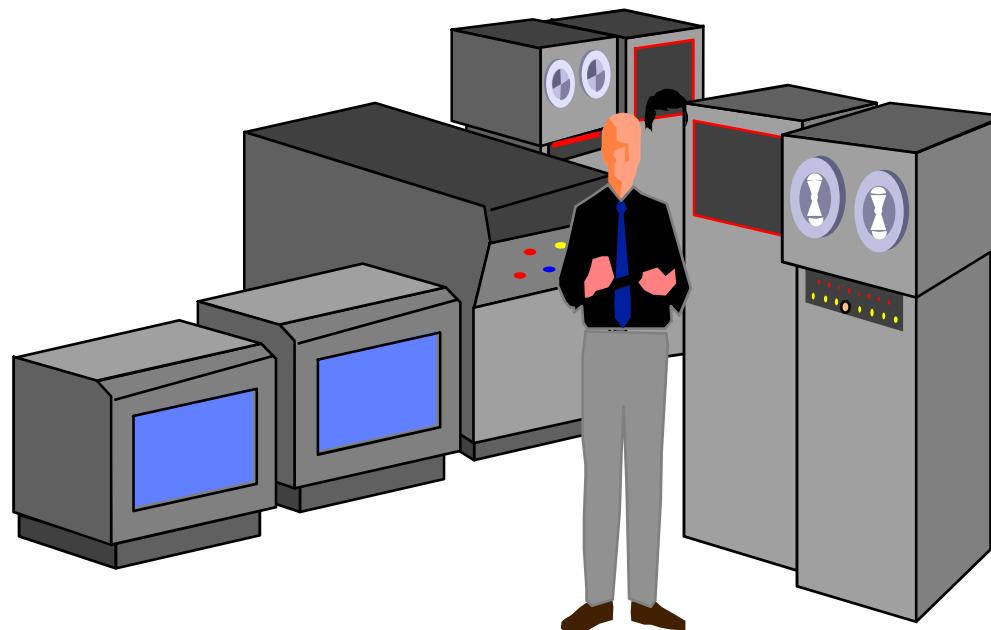


# Modelo de Von Neumann

**COMO EVOLUIU O MODELO DE VON NEUMANN ?**



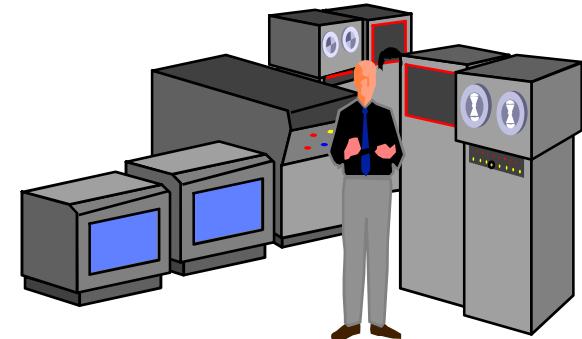
# Sistemas Centralizados



**MAINFRAME, UMA MÁQUINA VON NEUMANN !**

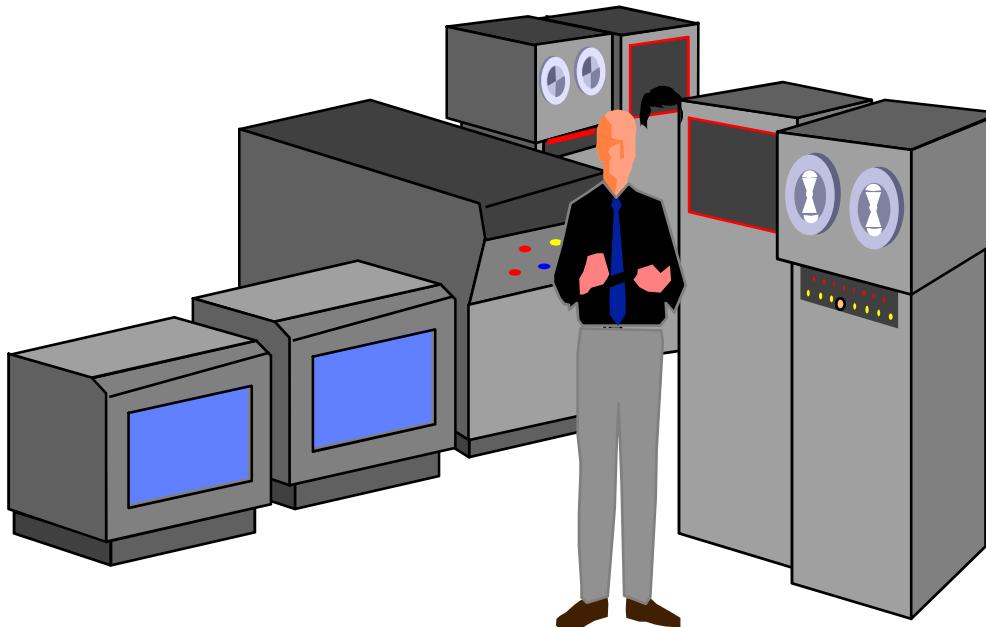
# Sistemas Centralizados

## Evolução em Software

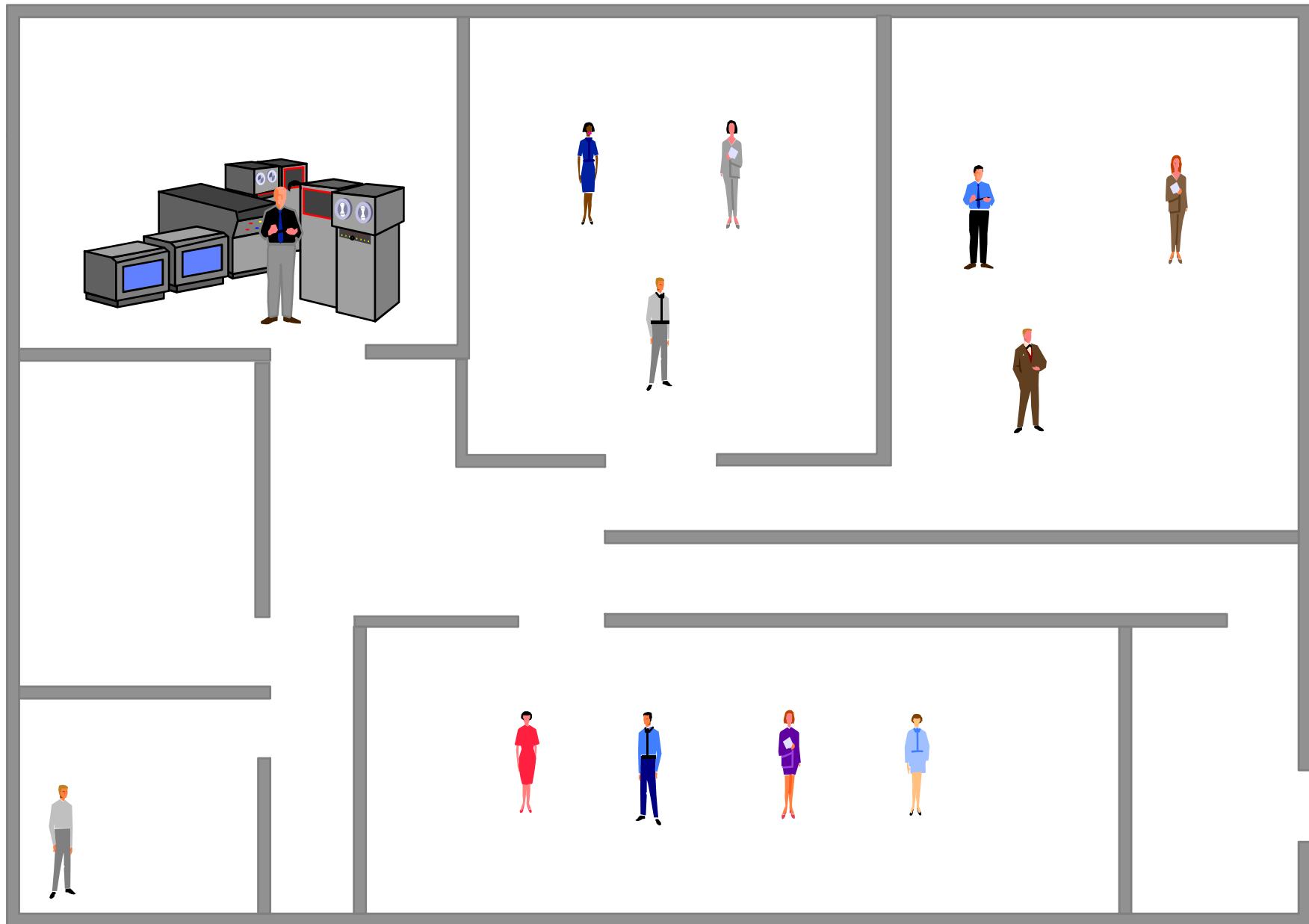


- Multiprogramação ou  
Multiprocessamento ?

# Sistemas Centralizado em Batch

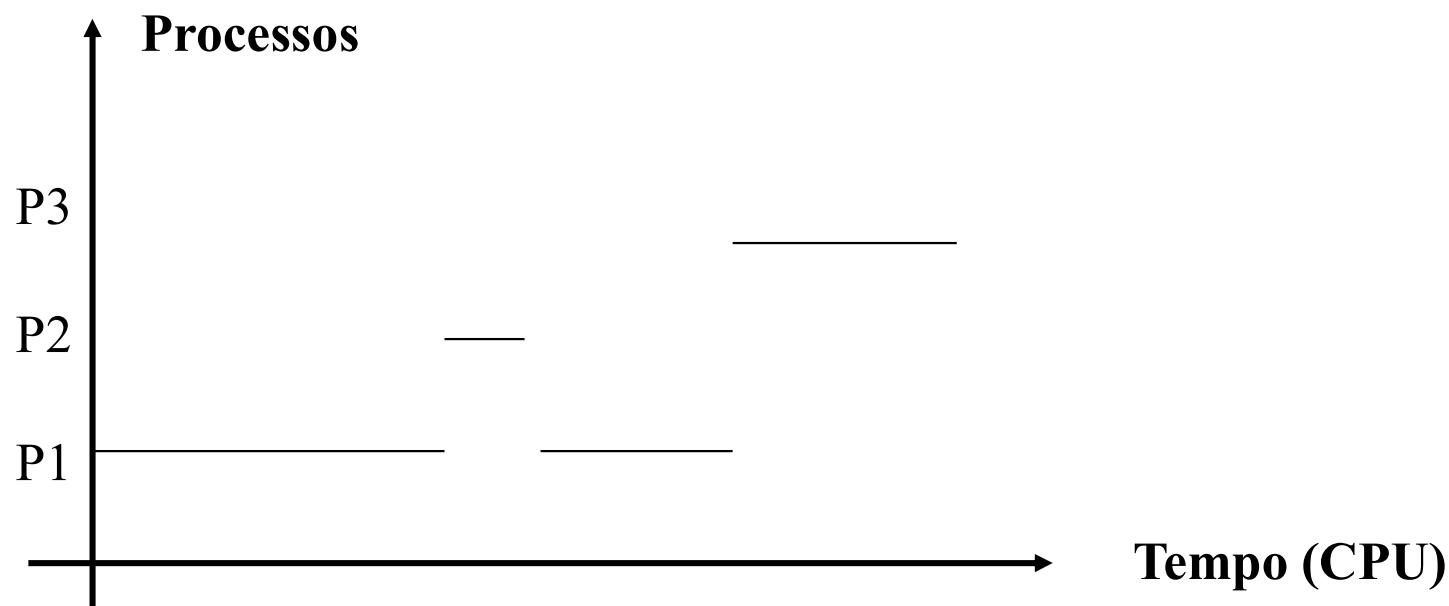
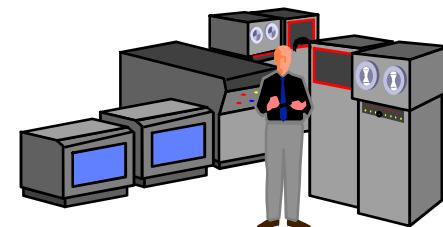


## Processamento Batch

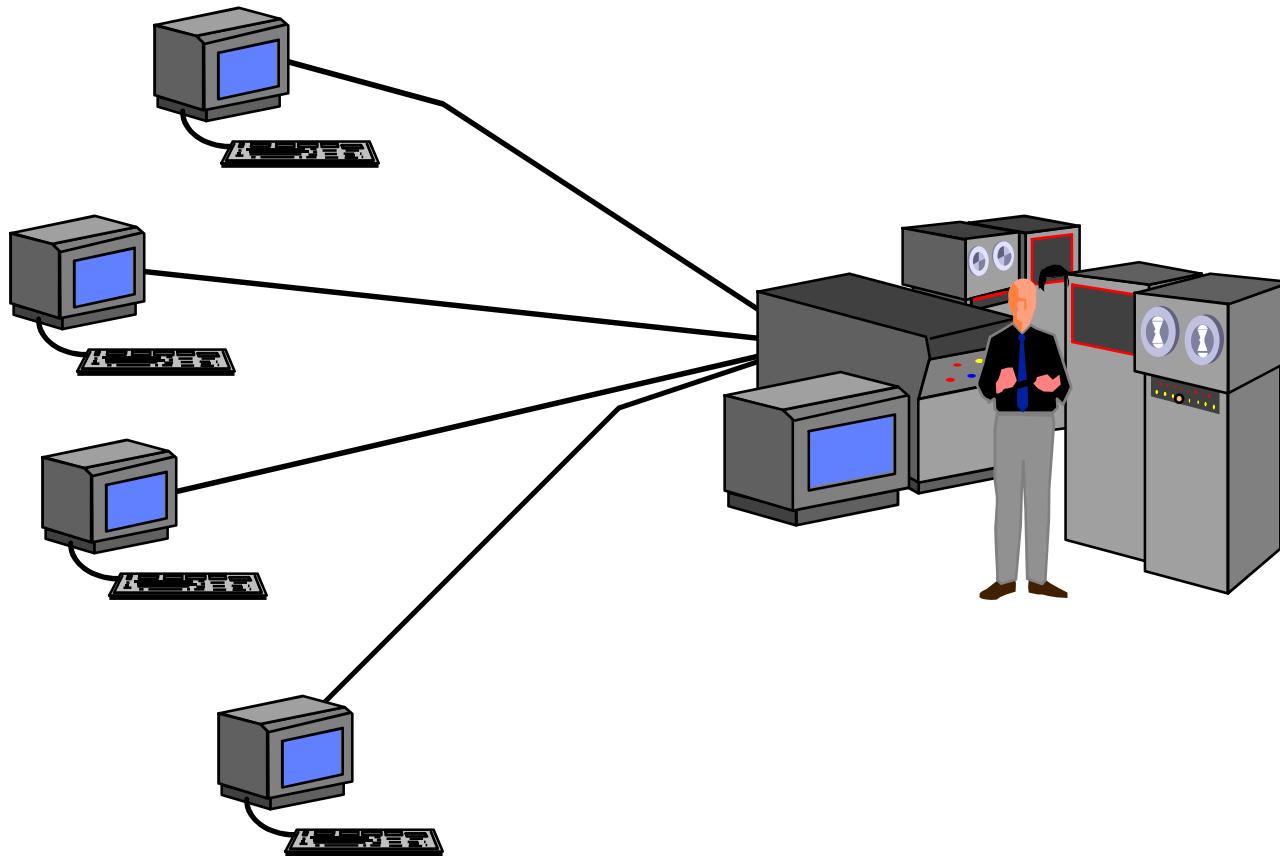


# Multiprogramação

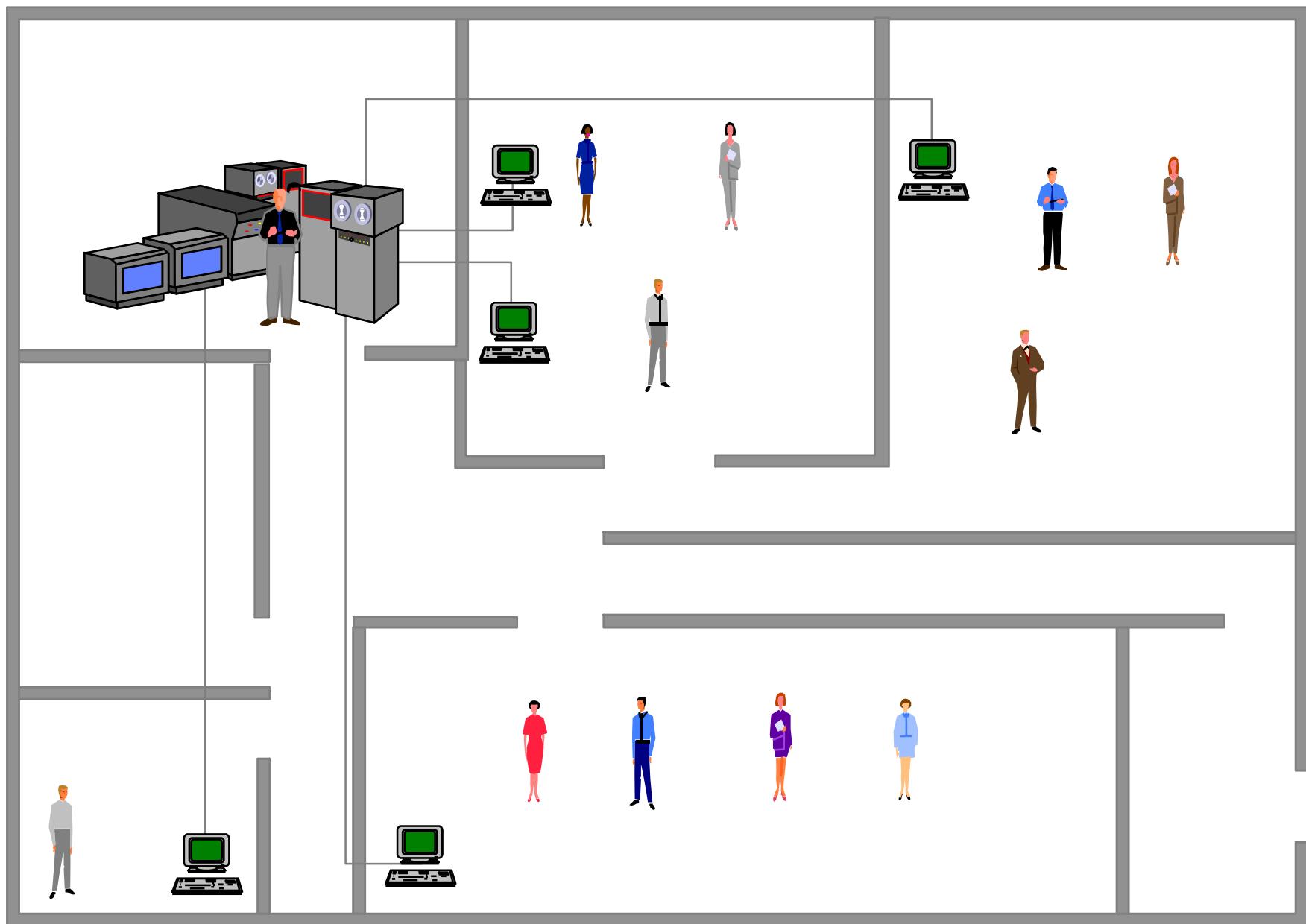
Processamento em Batch



# Sistemas Centralizados on Line

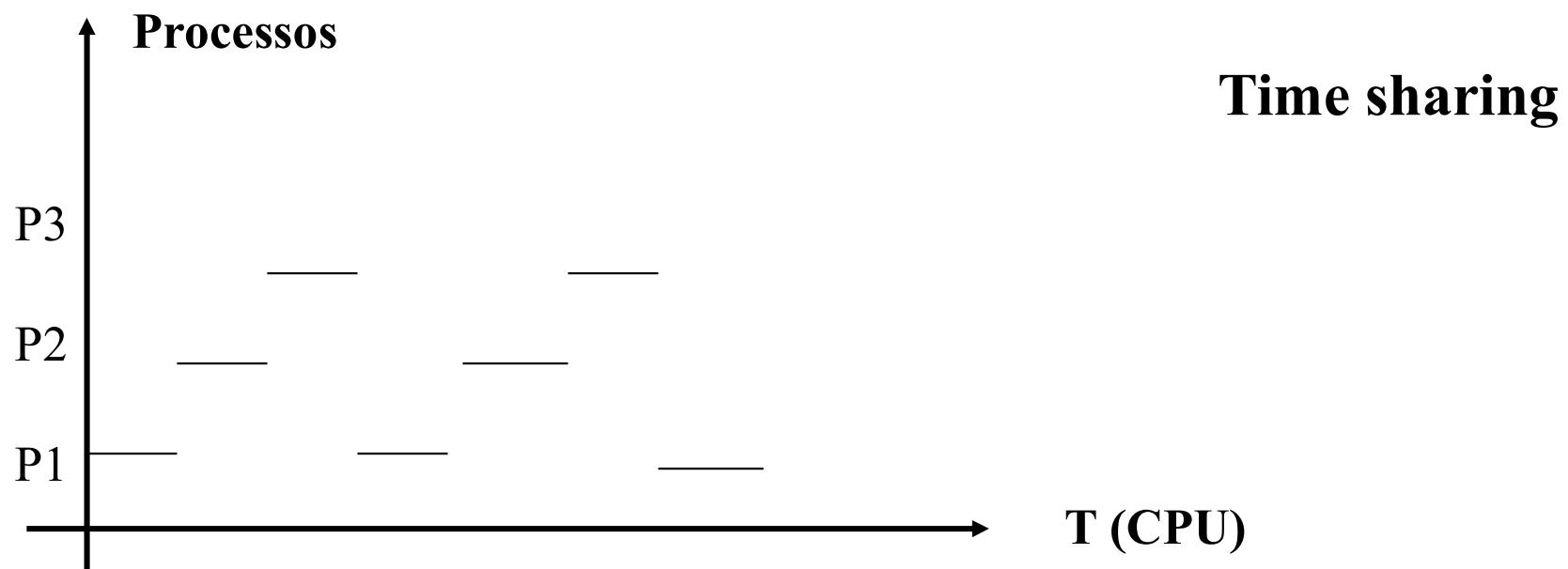
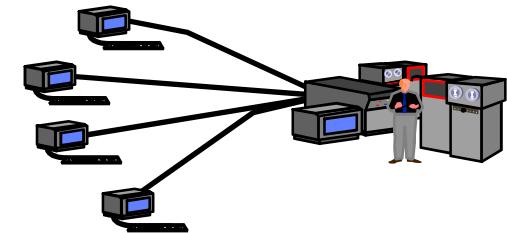


# Processamento On-Line



# Multiprogramação

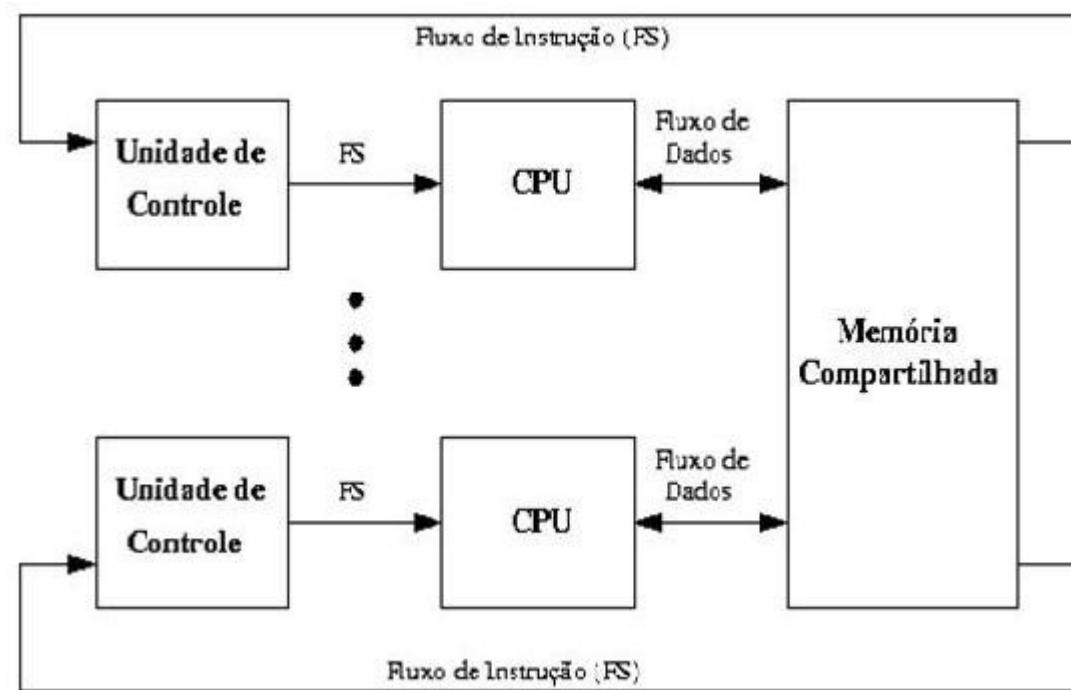
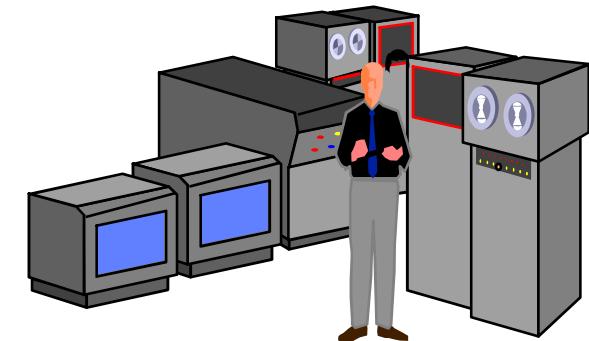
Processamento On-line



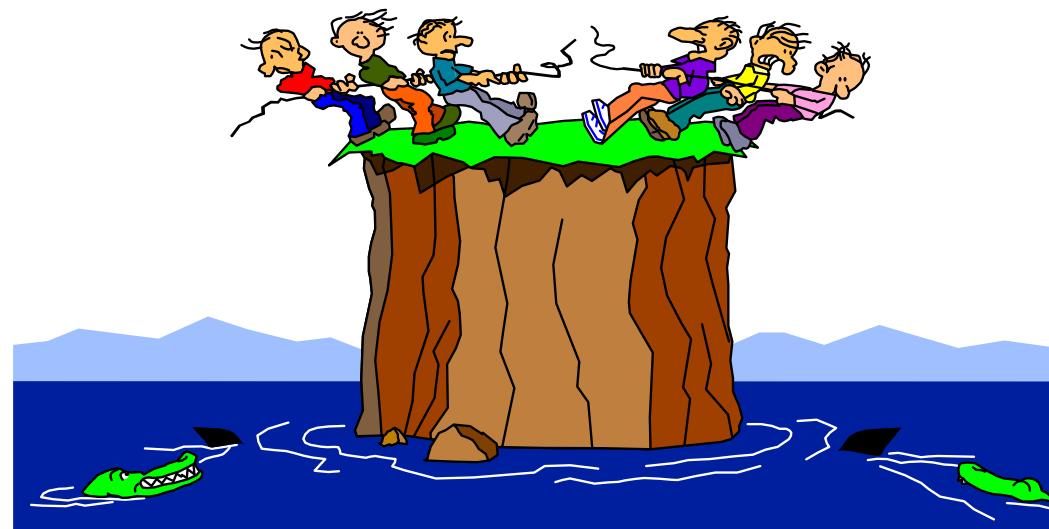
# Sistemas Centralizados

## Evolução em Hardware

- Multiprocessamento



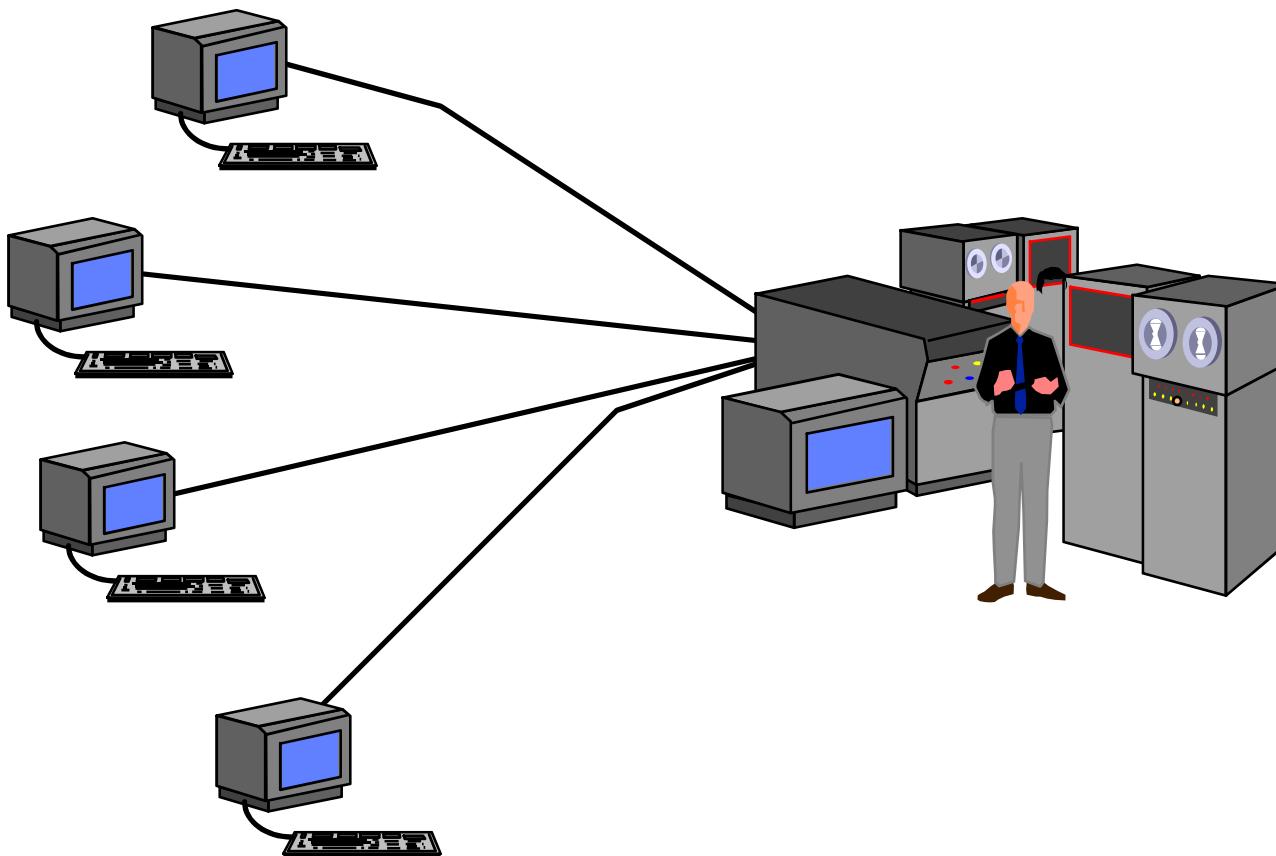
## 1.3 Sistemas Distribuídos



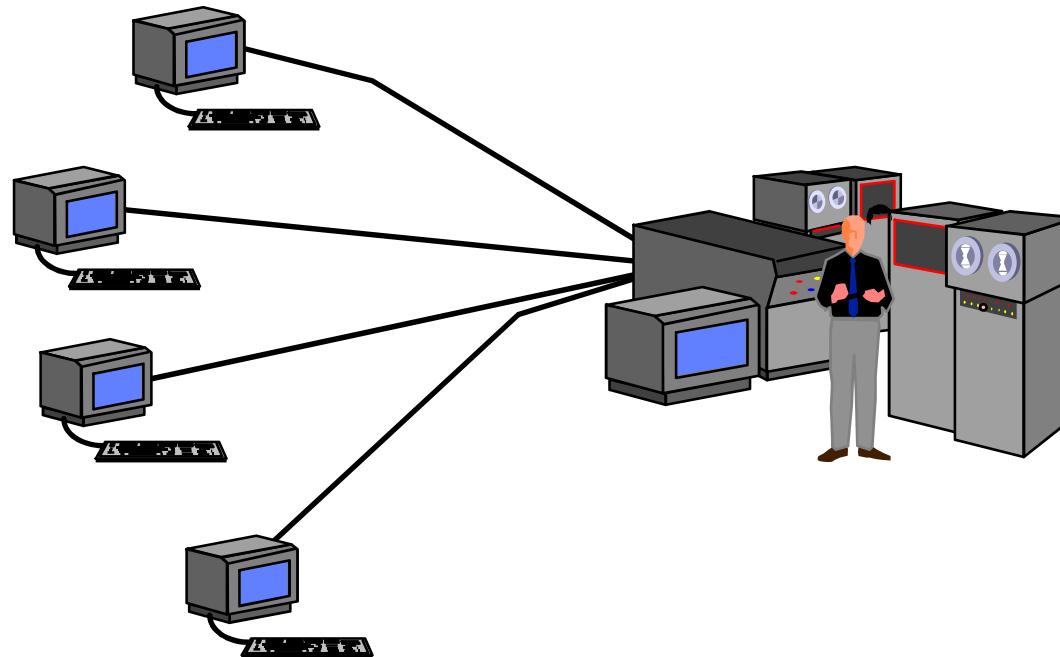
**UM SISTEMA CENTRALIZADO E SEUS TERMINAIS PODEM SER CONSIDERADOS UMA REDE DE COMPUTADORES ?**



# Sistemas Centralizados

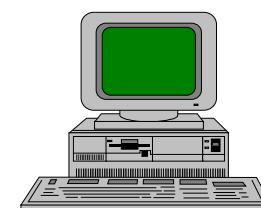
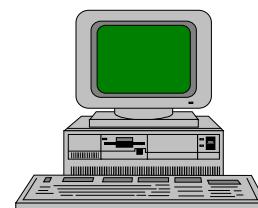
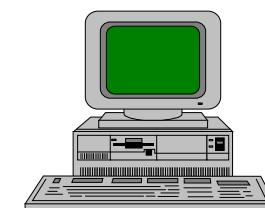
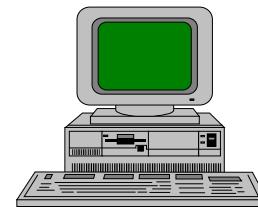
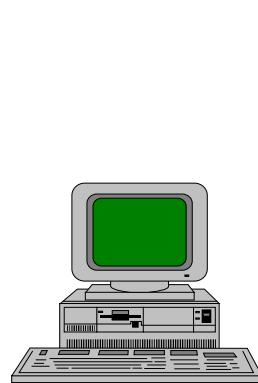


# Sistemas Centralizados



**UM SISTEMA CENTRALIZADO E SEUS TERMINAIS PODEM SER CONSIDERADOS UMA REDE DE COMPUTADORES ?**

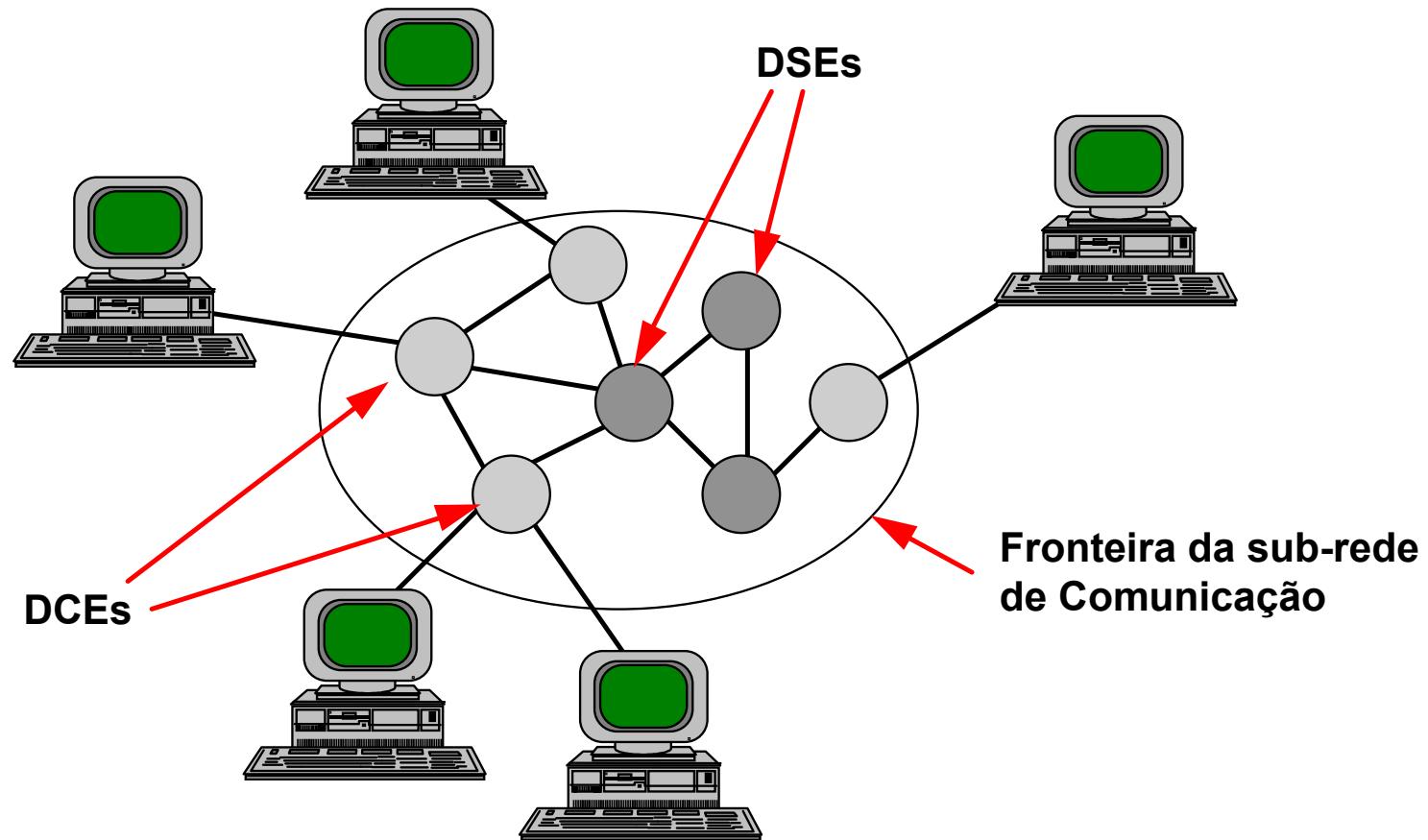
# Surgem os Microcomputadores



# Surgem as Redes de Computadores

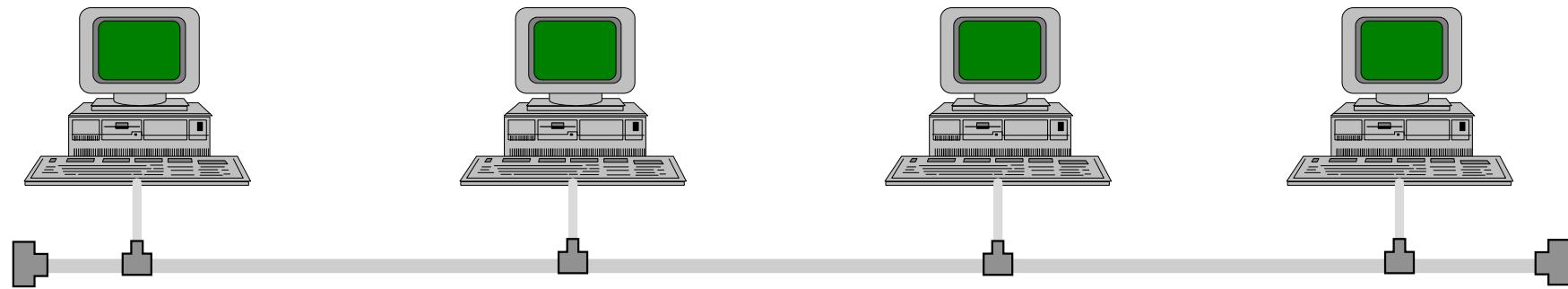


# Redes de Longa Distância (WAN)



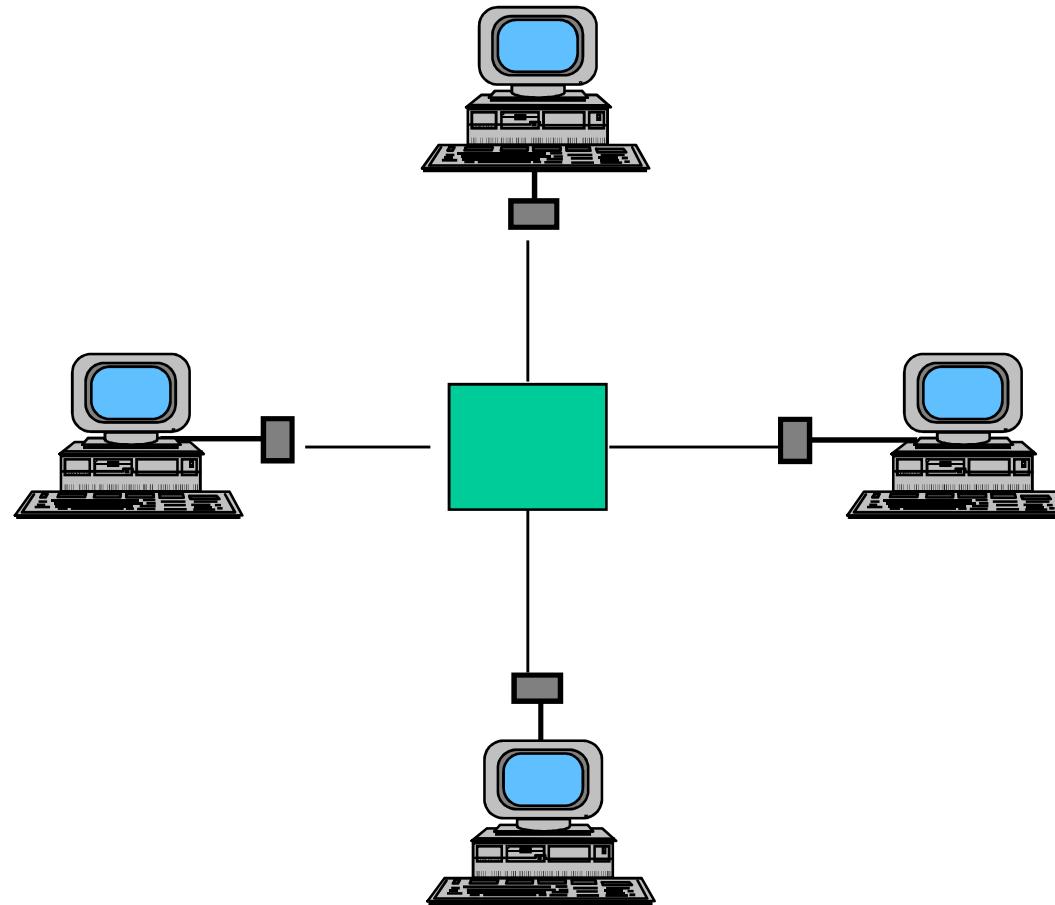
# Rede Local (LAN)

## Topologia em barra



# Rede Local (LAN)

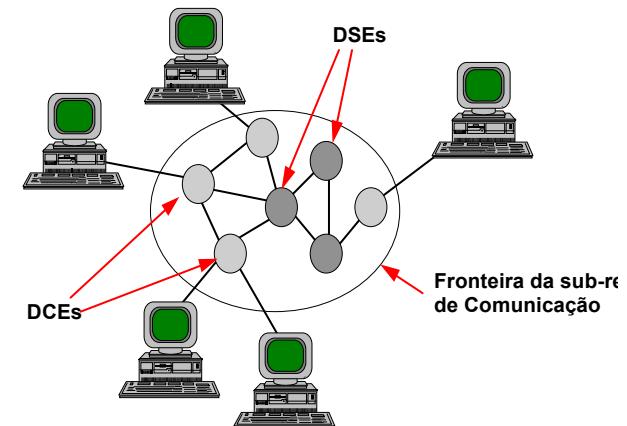
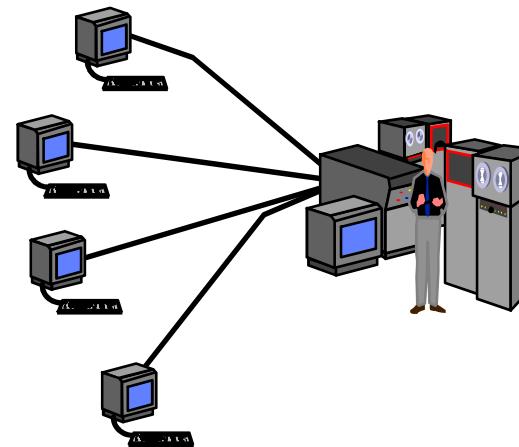
## Topologia em Estrela



# Processos em comunicação

**Processo:** programa que executa num hospedeiro

- processos no mesmo hospedeiro se comunicam usando **comunicação entre processos** definida pelo sistema operacional (SO)
  - processos em hospedeiros distintos se comunicam trocando **mensagens através da rede**



# Arquiteturas das aplicações

- Cliente-servidor
  - Peer-to-peer (P2P)
  - Híbrido de cliente-servidor e P2P

# Processos em comunicação

**Processo cliente:** processo que inicia a comunicação

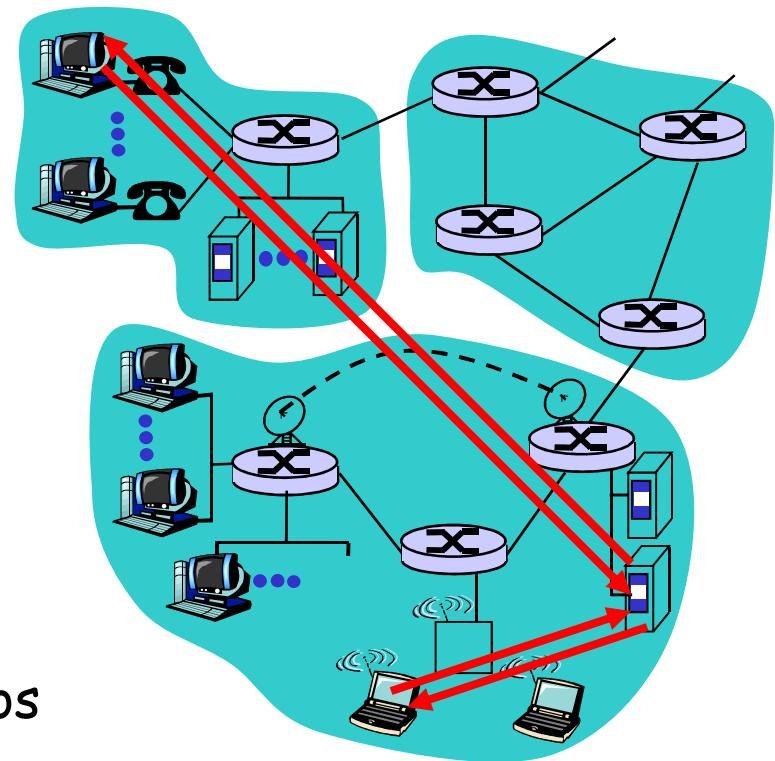
**Processo servidor:** processo que espera para ser contatado

- Aplicações com arquiteturas P2P possuem processos clientes e processos servidores

# Arquitetura cliente-servidor

## Servidor:

- Sempre ligado
- Endereço permanente



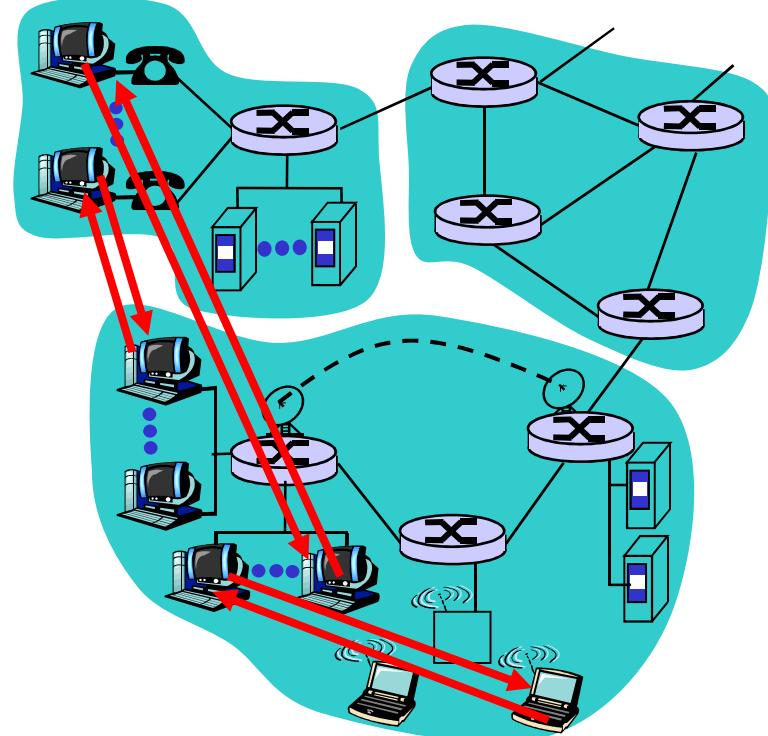
## Cliente:

- Comunica-se com o servidor
- Pode estar conectado intermitentemente
- Pode ter endereços dinâmicos
- Não se comunica diretamente com outros clientes

# Arquitetura P2P pura

- Não há servidor sempre ligado
- Sistemas finais arbitrários se comunicam diretamente
- Pares estão conectados intermitentemente e mudam endereços
- Exemplo: Gnutella

Altamente escalável  
Porém, difícil de gerenciar



# Internet das Coisas

