

# CURSO PRÁTICO DE BIOINFORMÁTICA:

## Aula 1 - Lógica de Programação

Renan Paulo Martin

15/07/19

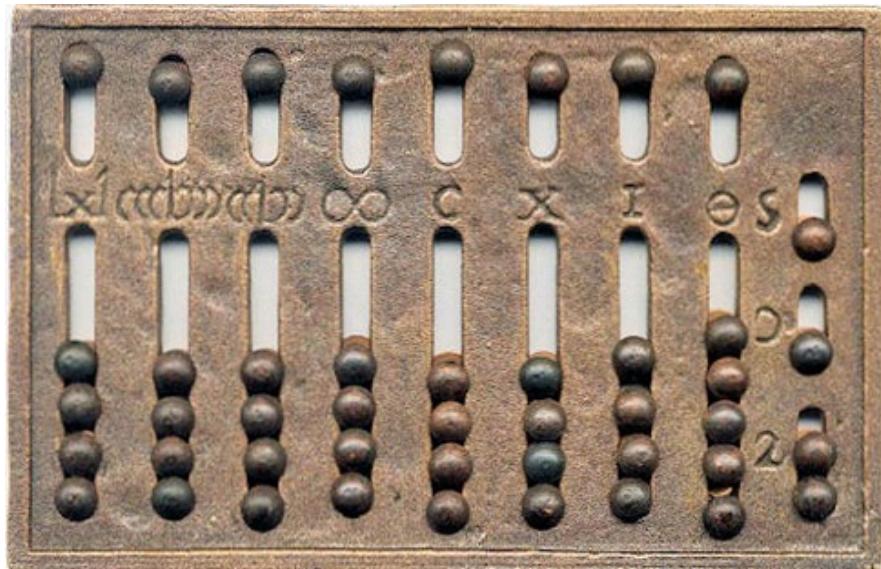
# Tópicos Abordados

- Introdução à lógica de programação
  - História da Computação
  - Linguagem de Programação e Algoritmo
  - Estrutura de Dados, de Controle e de Repetição
- Ferramentas avançadas de excel
  - Lógica de programação com excel
  - Banco de dados com excel
- Atividade Prática de Lógica com Excel

# História da Computação (Primórdios)

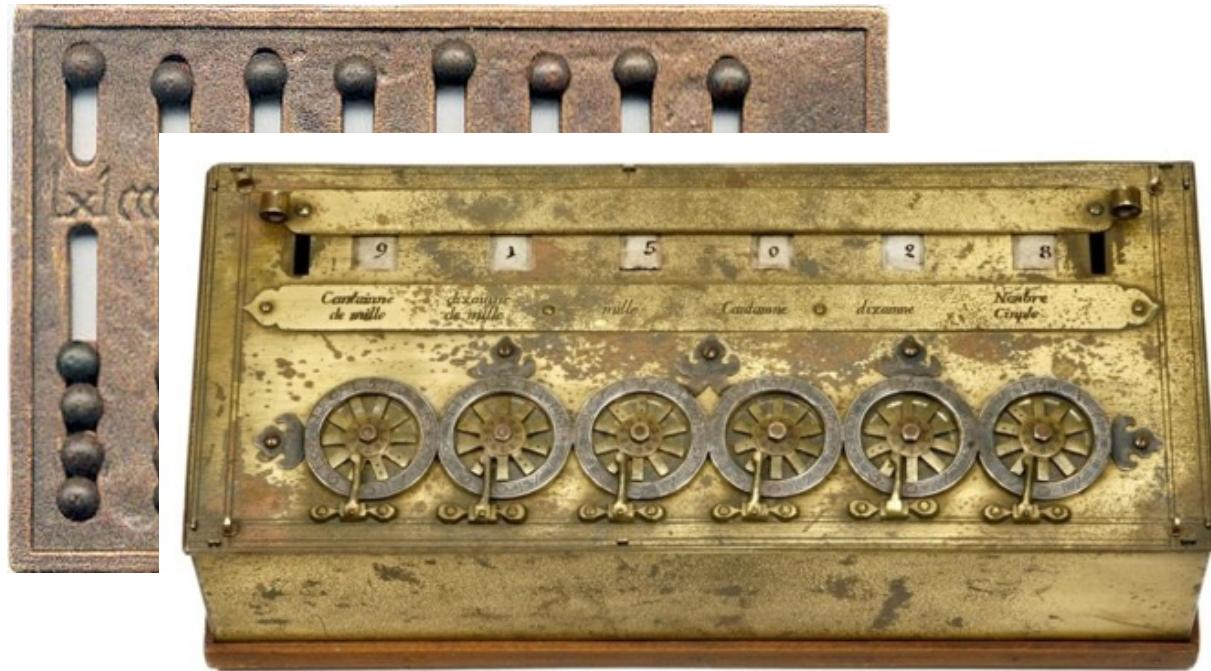
- Ábaco ( $\cong$  2.000 anos atrás)
- Máquina de somar de Blaise Pascal (1642)
- Máquina Analítica de Charles Babbage (1837)
  - Primeiro programa de computador Ada Lovelace (1843)
- Cartões perfurados de Herman Hollerith (1890)
- Calculadora eletromecânica de Howard Aiken (1937)
- Colossus de Alan Turing ( $\cong$  1940)
- ENIAC - 300 multiplicações por segundo (1945)

# História da Computação (Primórdios)



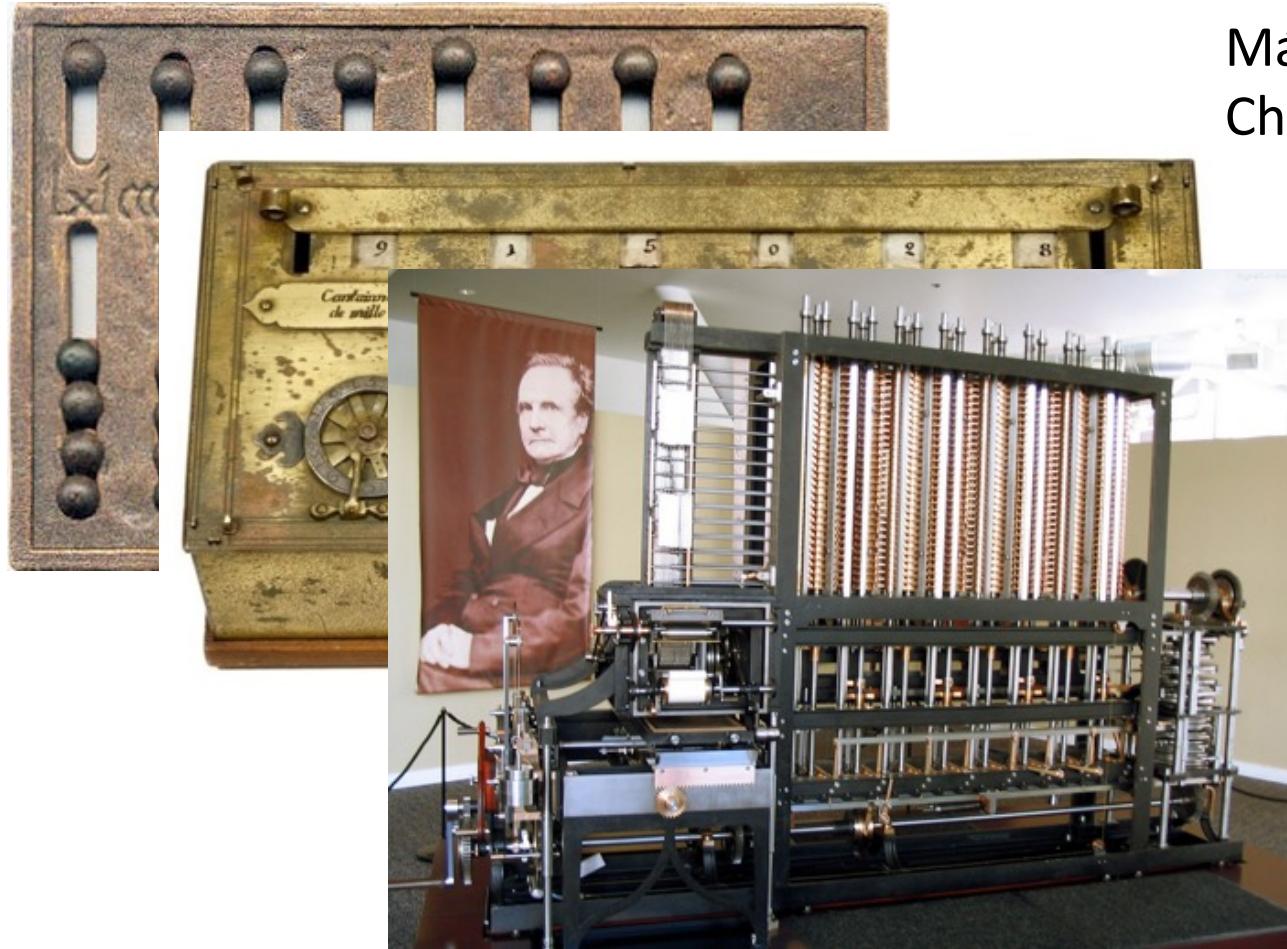
Ábaco ( $\cong$  2.000 anos atrás)

# História da Computação (Primórdios)



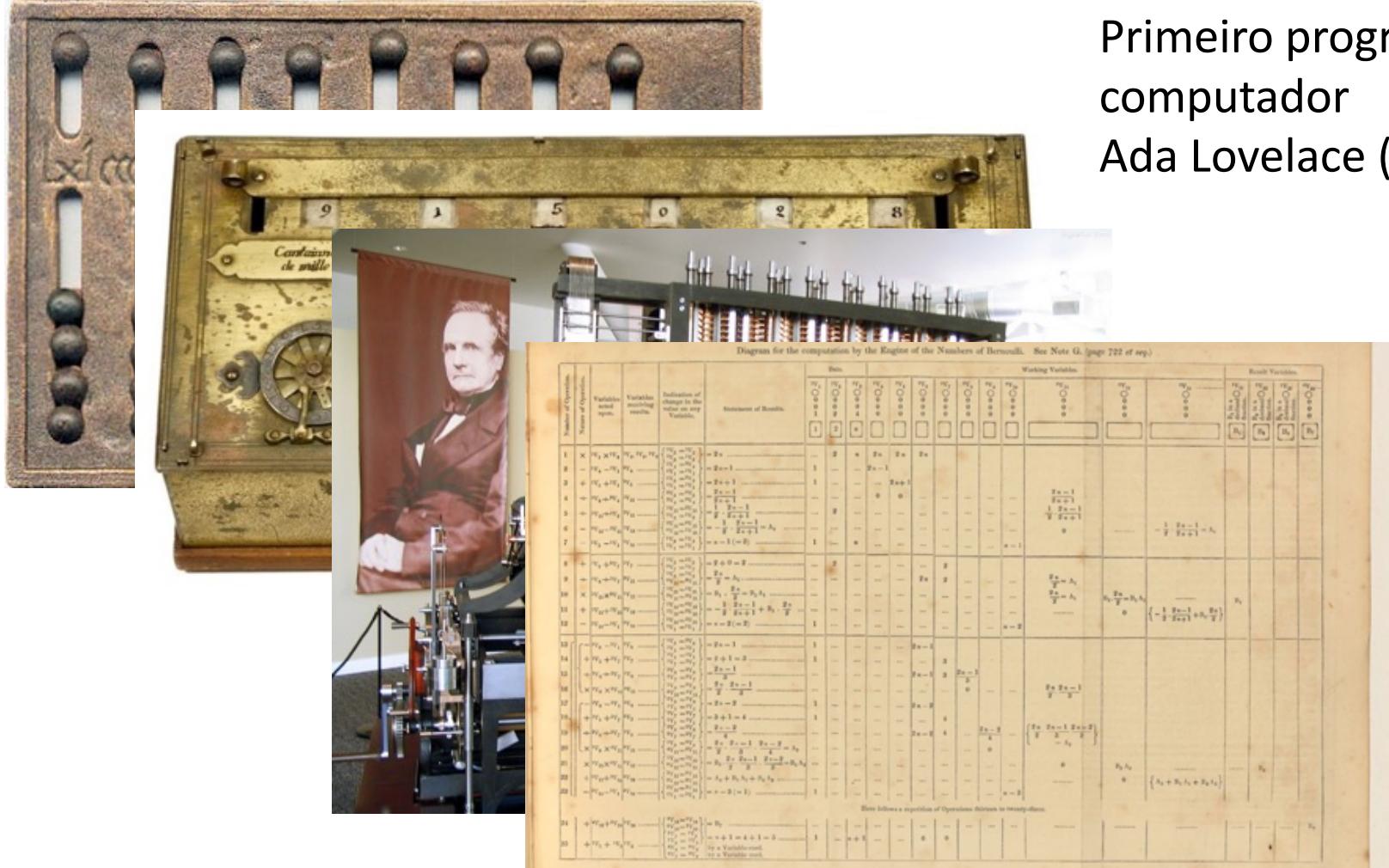
Máquina de somar de Blaise Pascal (1642)

# História da Computação (Primórdios)



Máquina Analítica de  
Charles Babbage (1837)

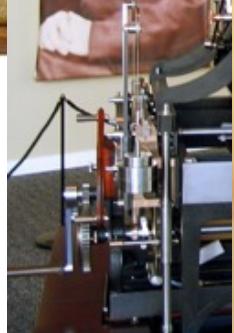
# História da Computação (Primórdios)



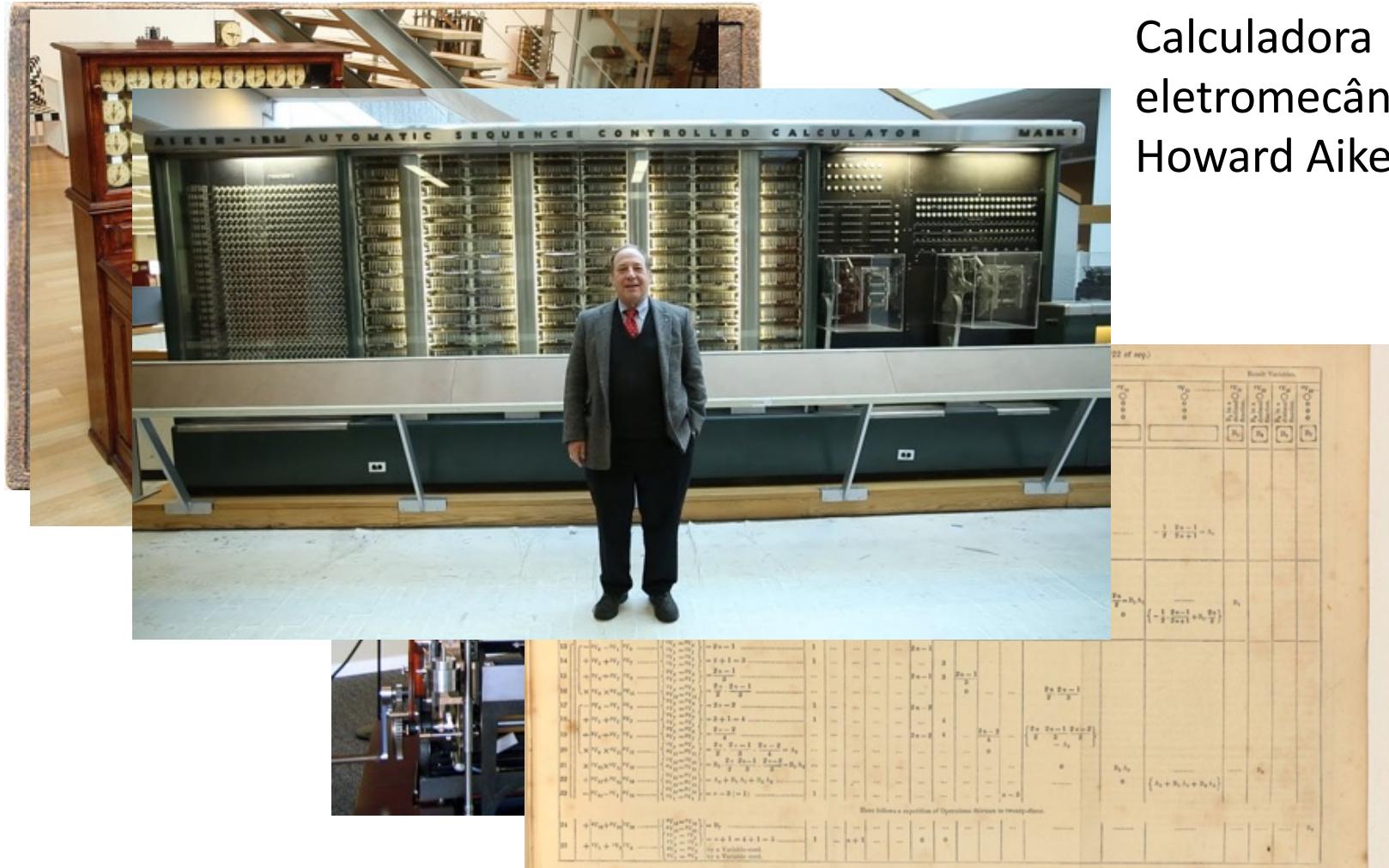
# História da Computação (Primórdios)



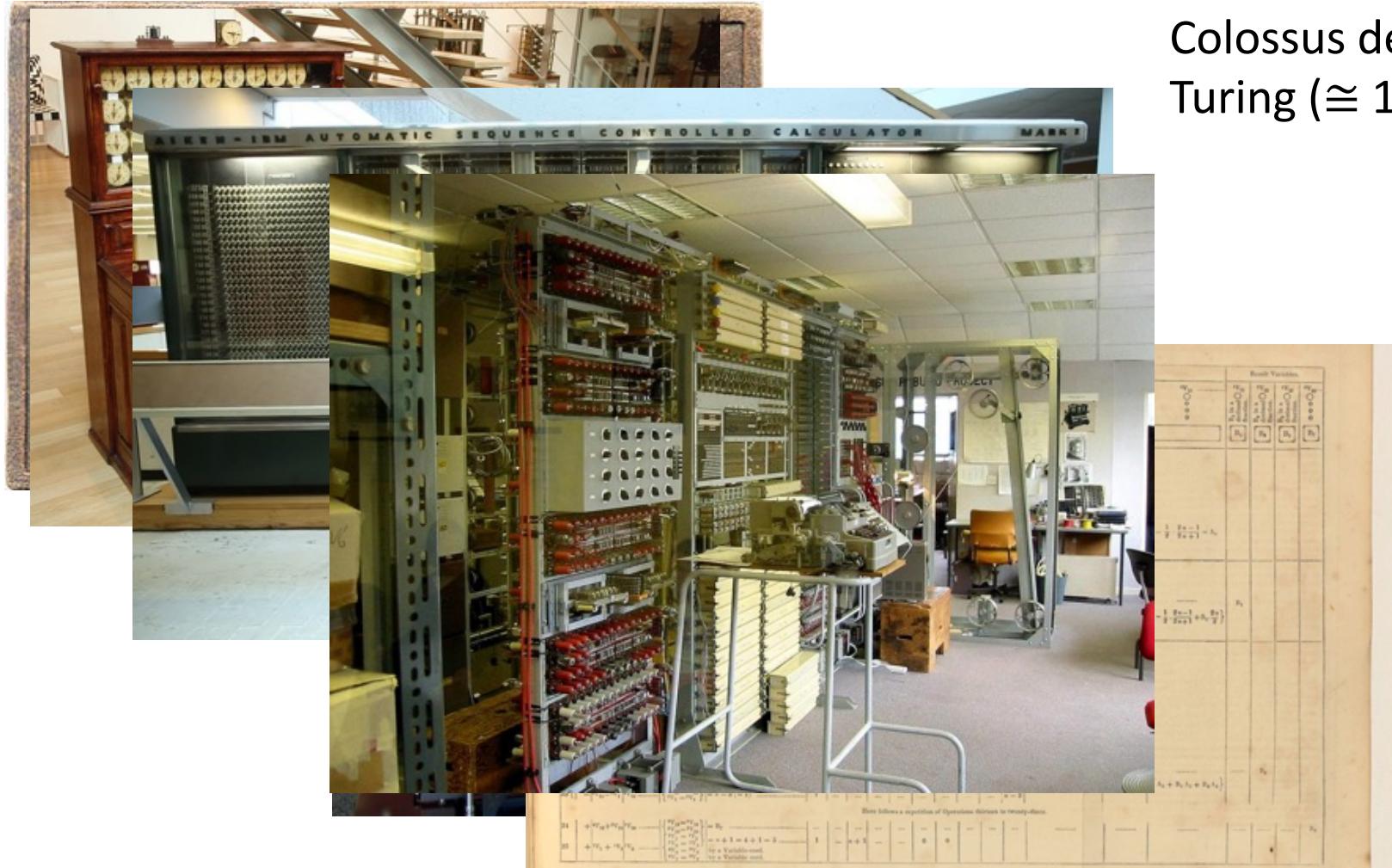
# Cartões perfurados de Herman Hollerith (1890)



# História da Computação (Primórdios)



# História da Computação (Primórdios)



Colossus de Alan Turing ( $\cong 1940$ )

# História da Computação (Primórdios)



ENIAC (1945)  
300 multiplicações  
por segundo

# História da Computação



Transistor

# História da Computação



Círculo Integrado

# História da Computação



Micro processador

# História da Computação



Computador Portátil

# História da Computação



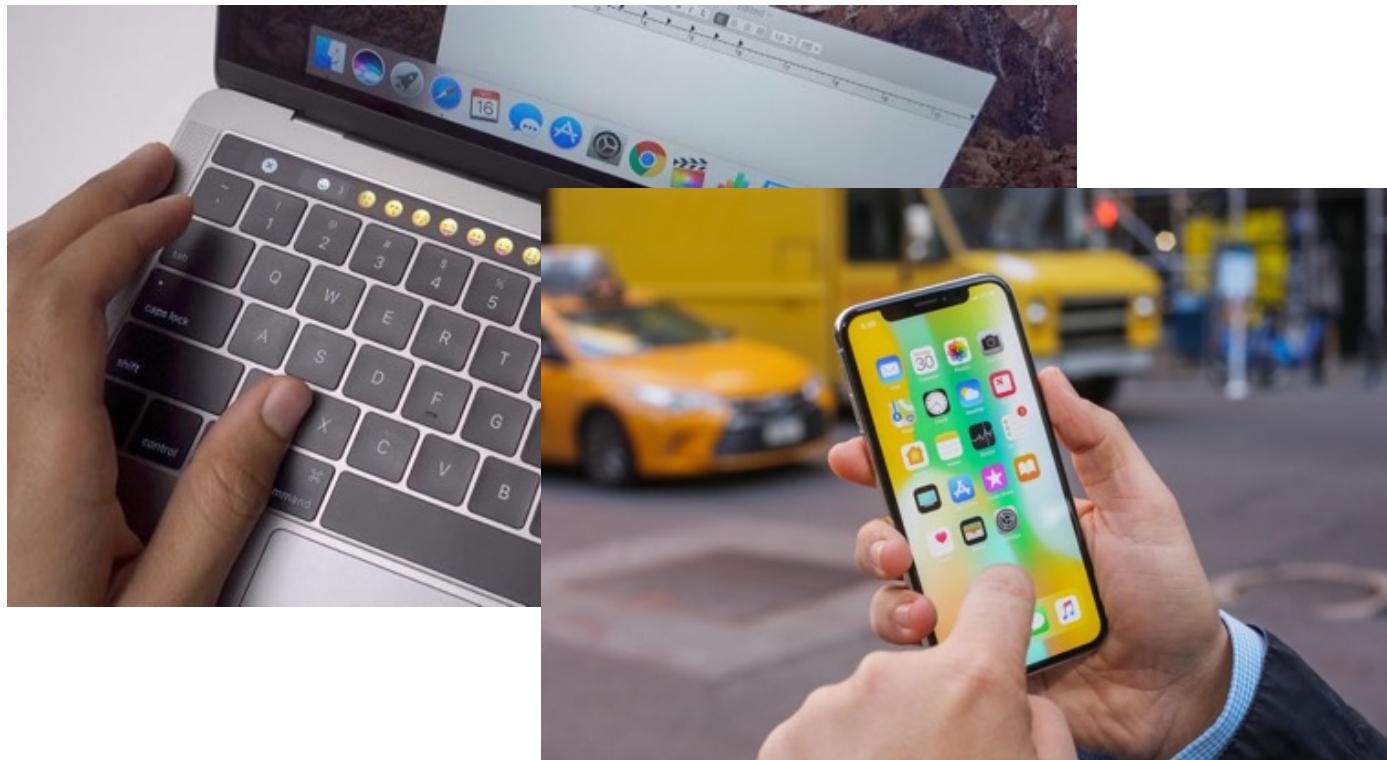
Interface Gráfica do Usuário



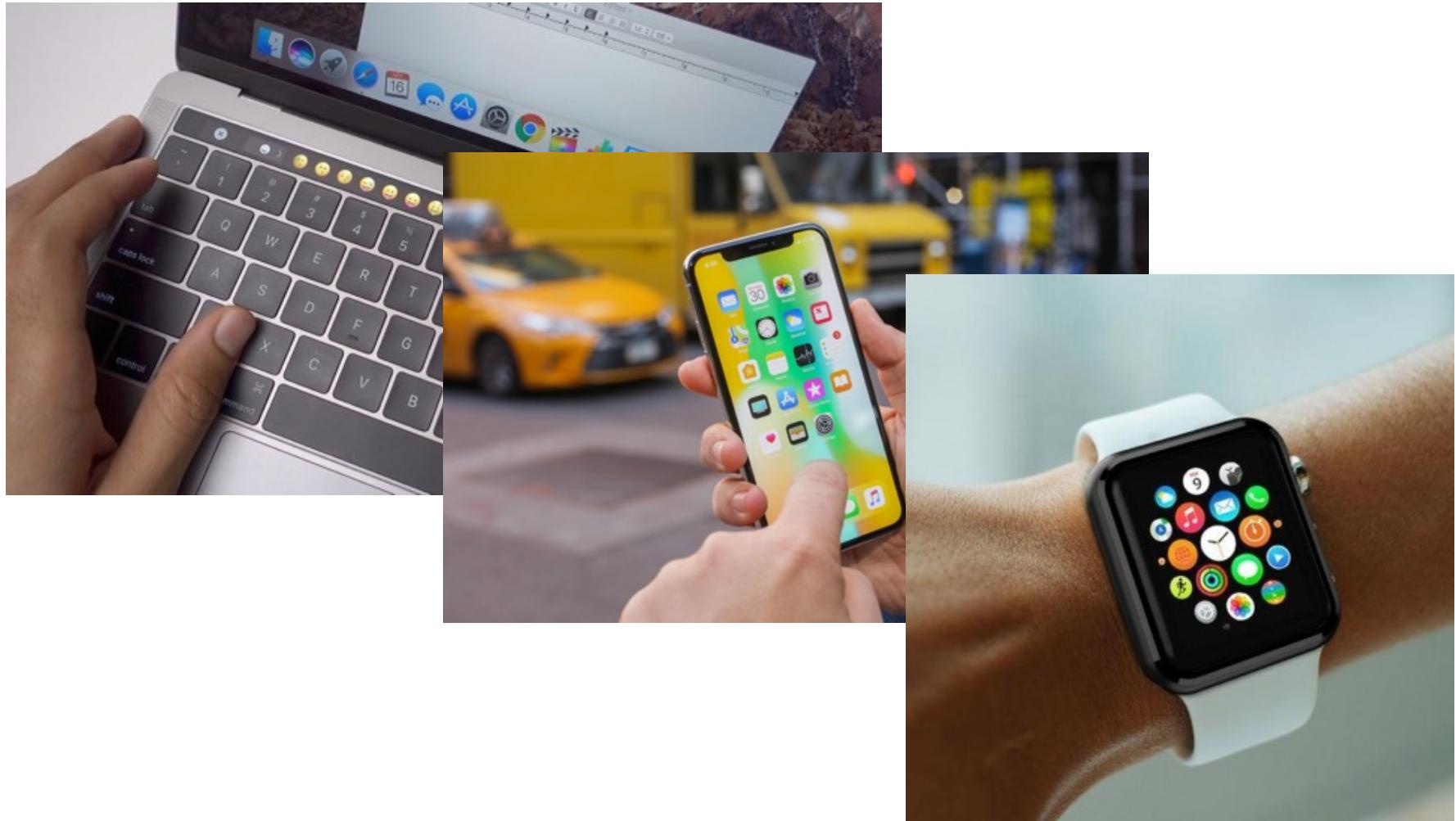
# História da Computação



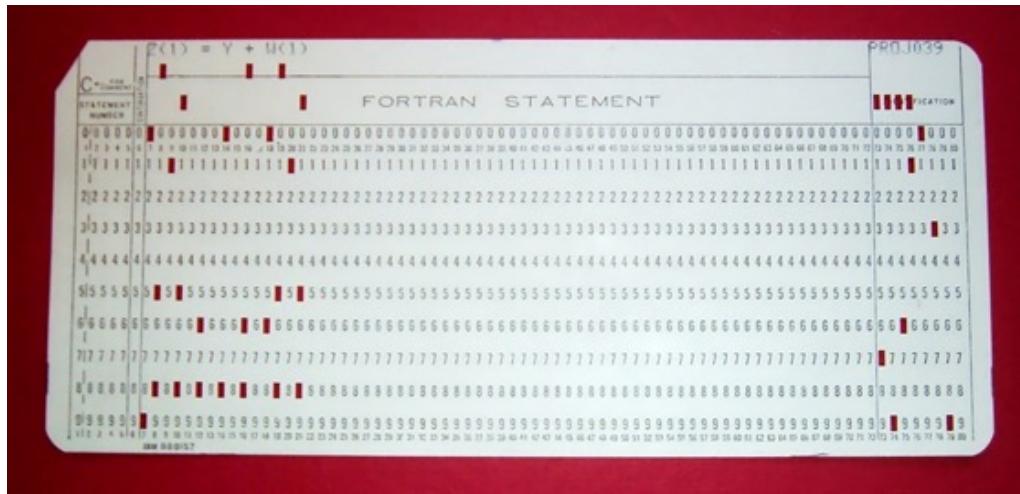
# História da Computação



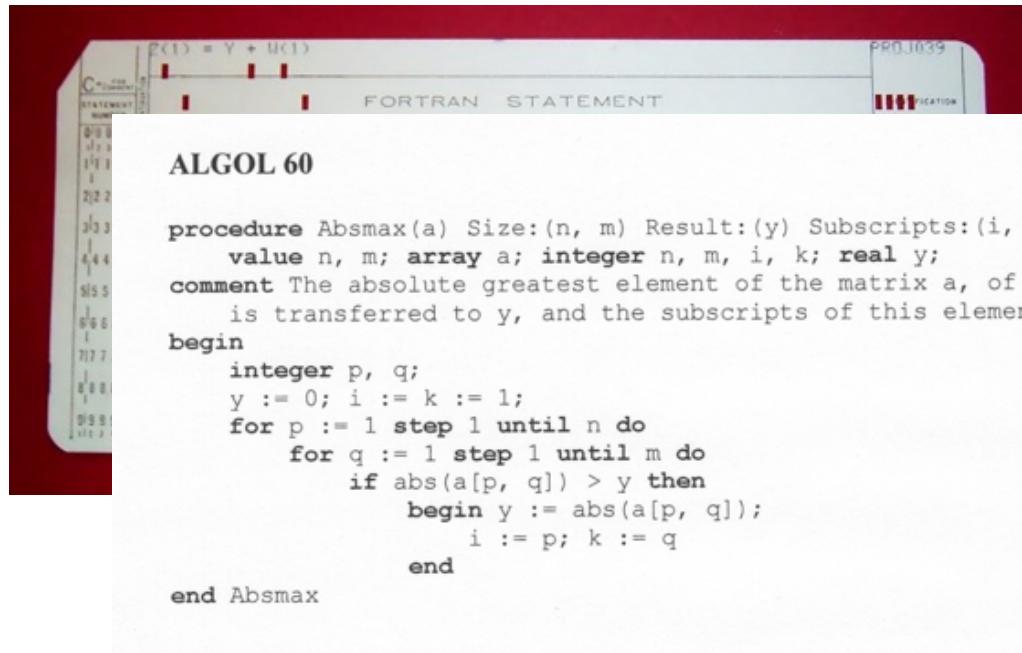
# História da Computação



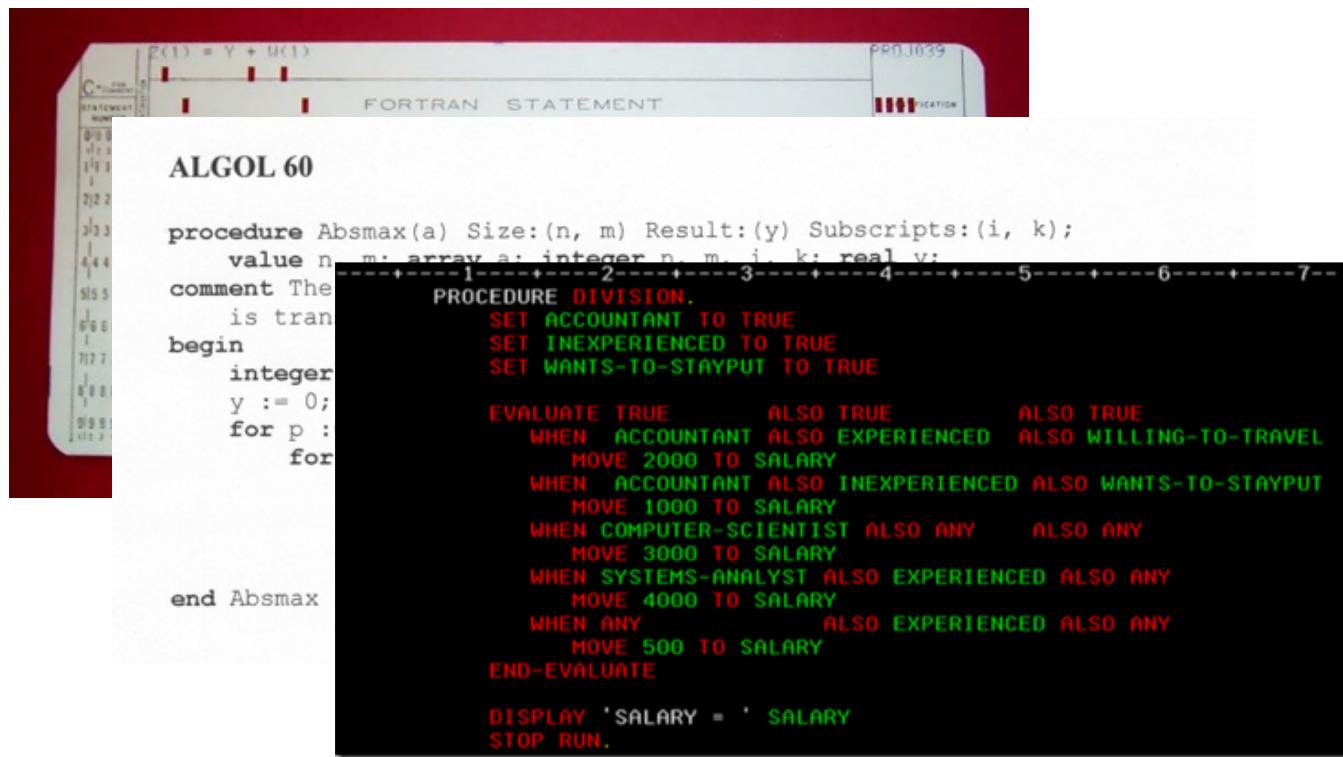
# Linguagens de Programação



# Linguagens de Programação



# Linguagens de Programação



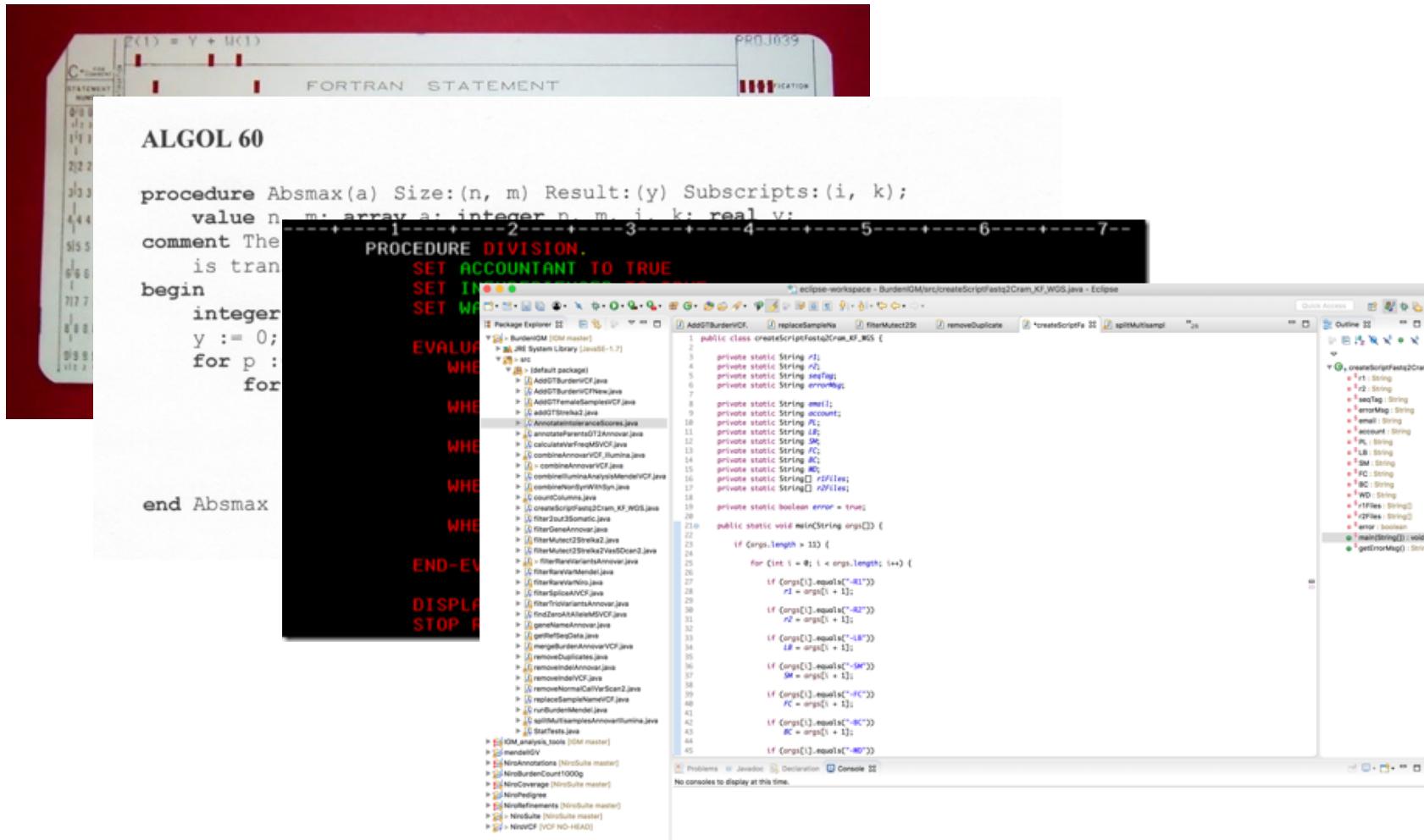
The image shows a vintage computer terminal screen with a dark background. At the top, there is a header bar with the text "FORTRAN STATEMENT" in the center. On the left side of the header, there is some small, illegible text. On the right side, there is a section labeled "PROJECT" with several colored bars below it.

The main area of the screen displays ALGOL 60 code. The code is color-coded, with different colors used for various keywords and identifiers. The code defines a procedure named "Absmax" with parameters "a", "Size:(n, m)", "Result:(y)", and "Subscripts:(i, k)". It includes comments about the parameters and a detailed description of the algorithm using pseudocode-like statements. The code ends with "end Absmax".

```
ALGOL 60

procedure Absmax(a) Size:(n, m) Result:(y) Subscripts:(i, k);
  value n, m: array a: integer n, m, i, k: real v:
comment The
  is tran
begin
  integer
  y := 0;
  for p :
    for
      EVALUATE TRUE ALSO TRUE ALSO TRUE
      WHEN ACCOUNTANT ALSO EXPERIENCED ALSO WILLING-TO-TRAVEL
        MOVE 2000 TO SALARY
      WHEN ACCOUNTANT ALSO INEXPERIENCED ALSO WANTS-TO-STAYPUT
        MOVE 1000 TO SALARY
      WHEN COMPUTER-SCIENTIST ALSO ANY ALSO ANY
        MOVE 3000 TO SALARY
      WHEN SYSTEMS-ANALYST ALSO EXPERIENCED ALSO ANY
        MOVE 4000 TO SALARY
      WHEN ANY ALSO EXPERIENCED ALSO ANY
        MOVE 500 TO SALARY
      END-EVALUATE
      DISPLAY 'SALARY = ' SALARY
      STOP RUN.
end Absmax
```

# Linguagens de Programação



# Arquitetura

# Arquitetura



# Arquitetura



# Arquitetura



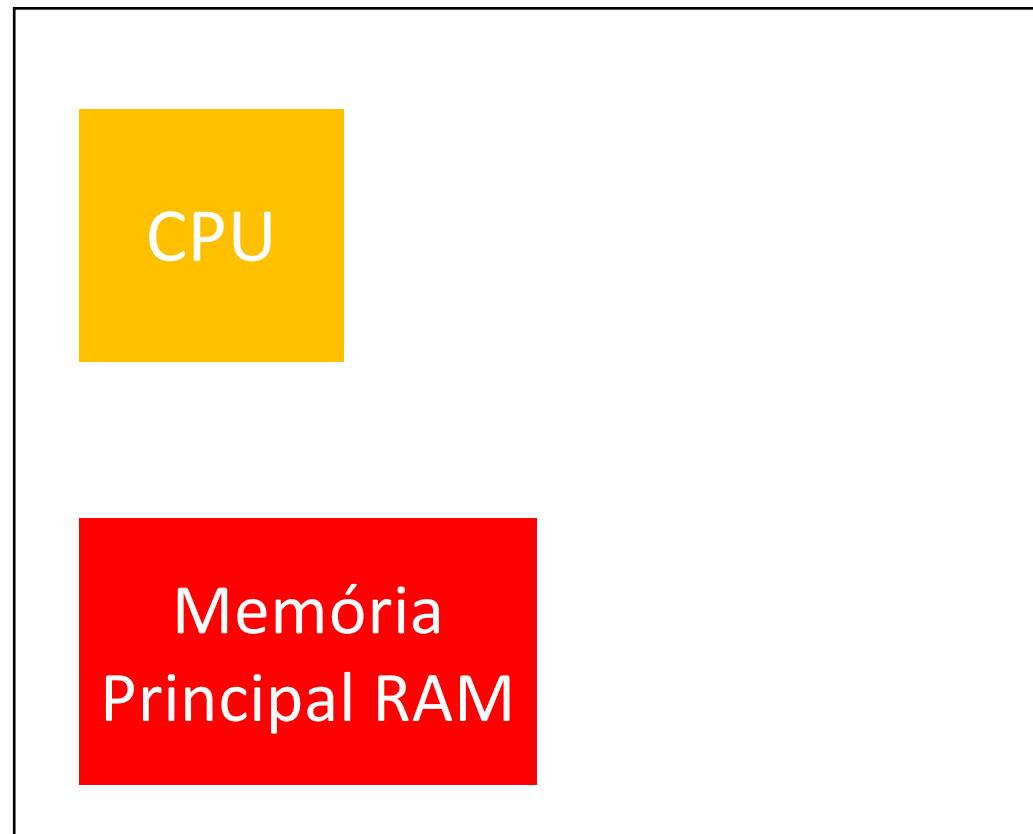
# Arquitetura de Computadores

# Arquitetura de Computadores



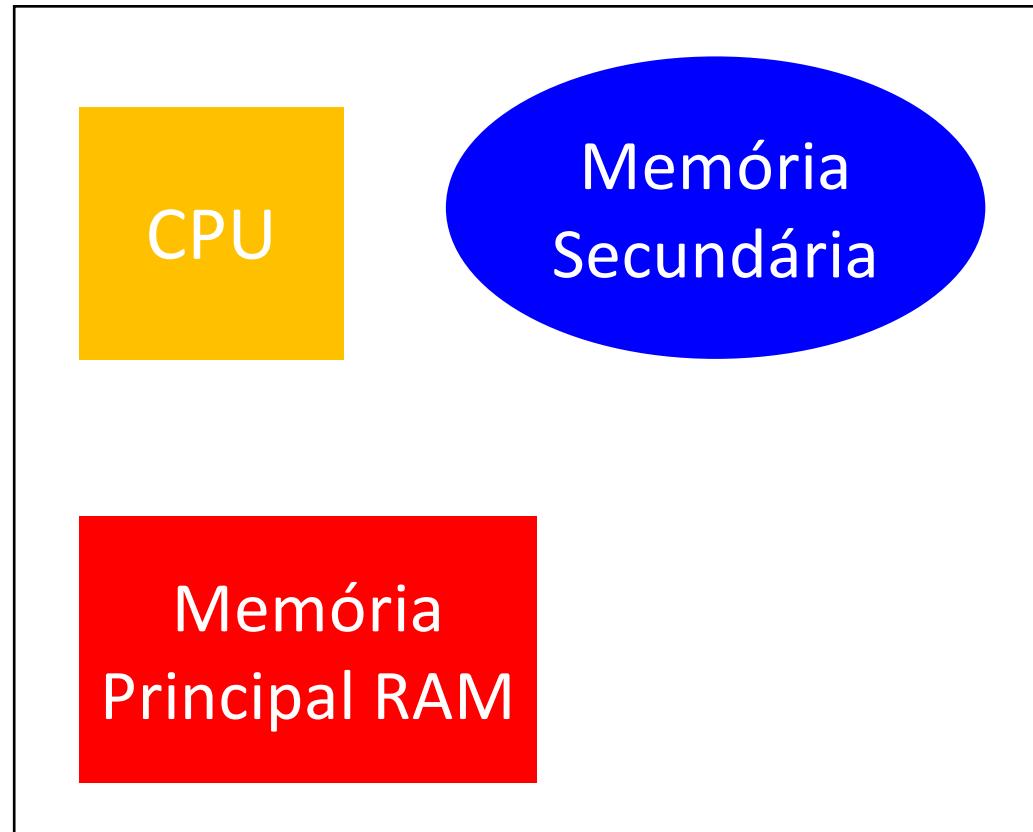
- Processador
- Realiza as operações lógicas

# Arquitetura de Computadores



- Memória de acesso aleatório
- Dados são gravados de forma não permanente
- Possui alta velocidade de gravação e leitura
- Computadores pessoais (2GB - 64GB)

# Arquitetura de Computadores



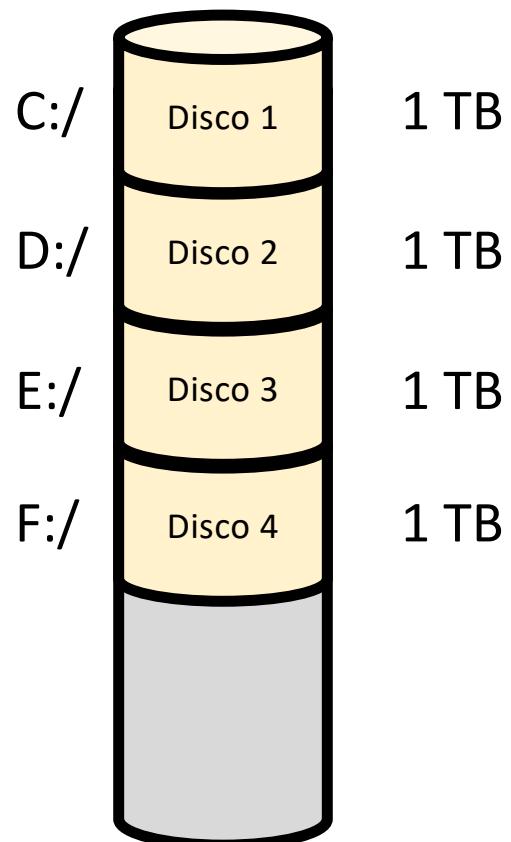
- Disco Rígido ou Estado Sólido
- Dados são gravados de forma permanente
- Velocidade de leitura e de escrita mais lentas que RAM
- 256GB - 4TB

# Memória Secundária

- Discos Independentes
  - HDD - Hard Drive Disk ou SDD - Solid State Drive Disk
  - C:/, D:/, E:/ ...
- Array de discos
- RAID - Redundant Array of Independent Disks
  - Melhoria no desempenho de leitura e escrita
  - Redundância de dados
  - Expansão do volume
  - Segurança
  - Todas as opções acima

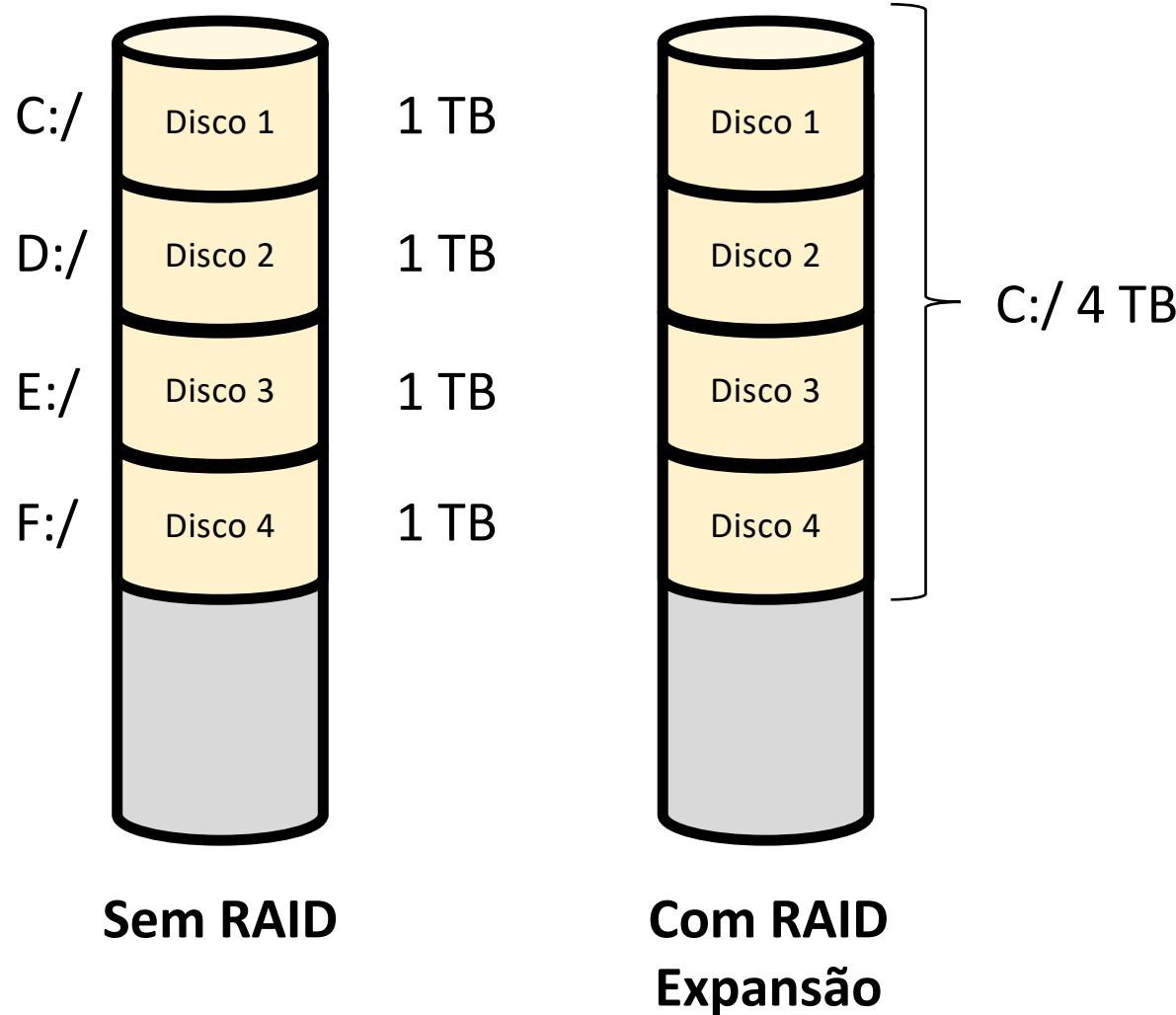
# RAID

# RAID

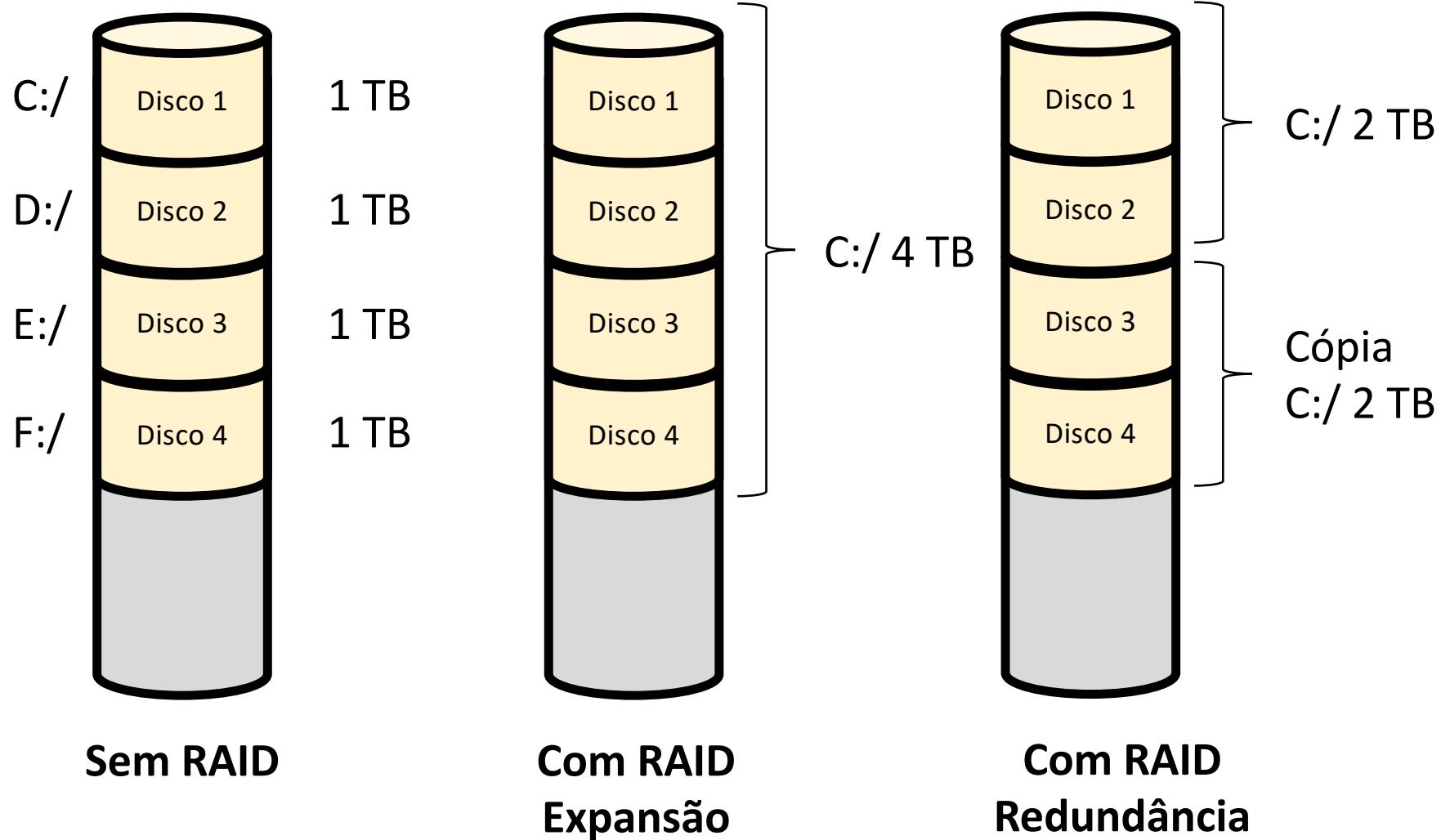


Sem RAID

# RAID

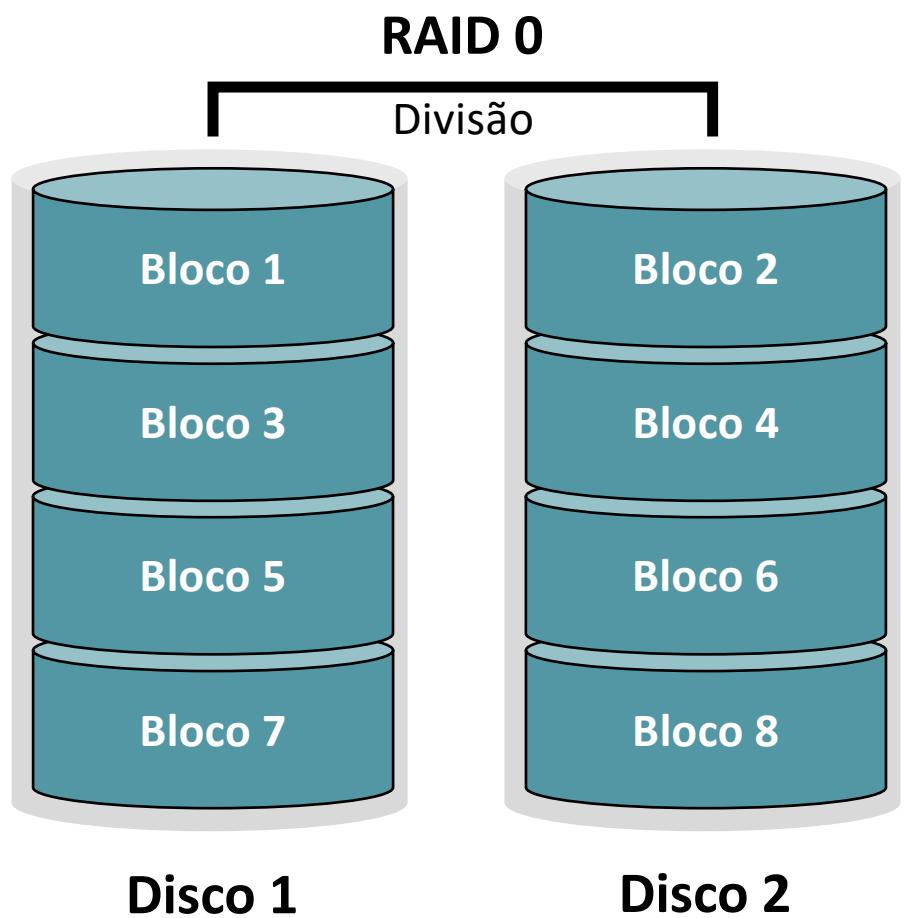


# RAID

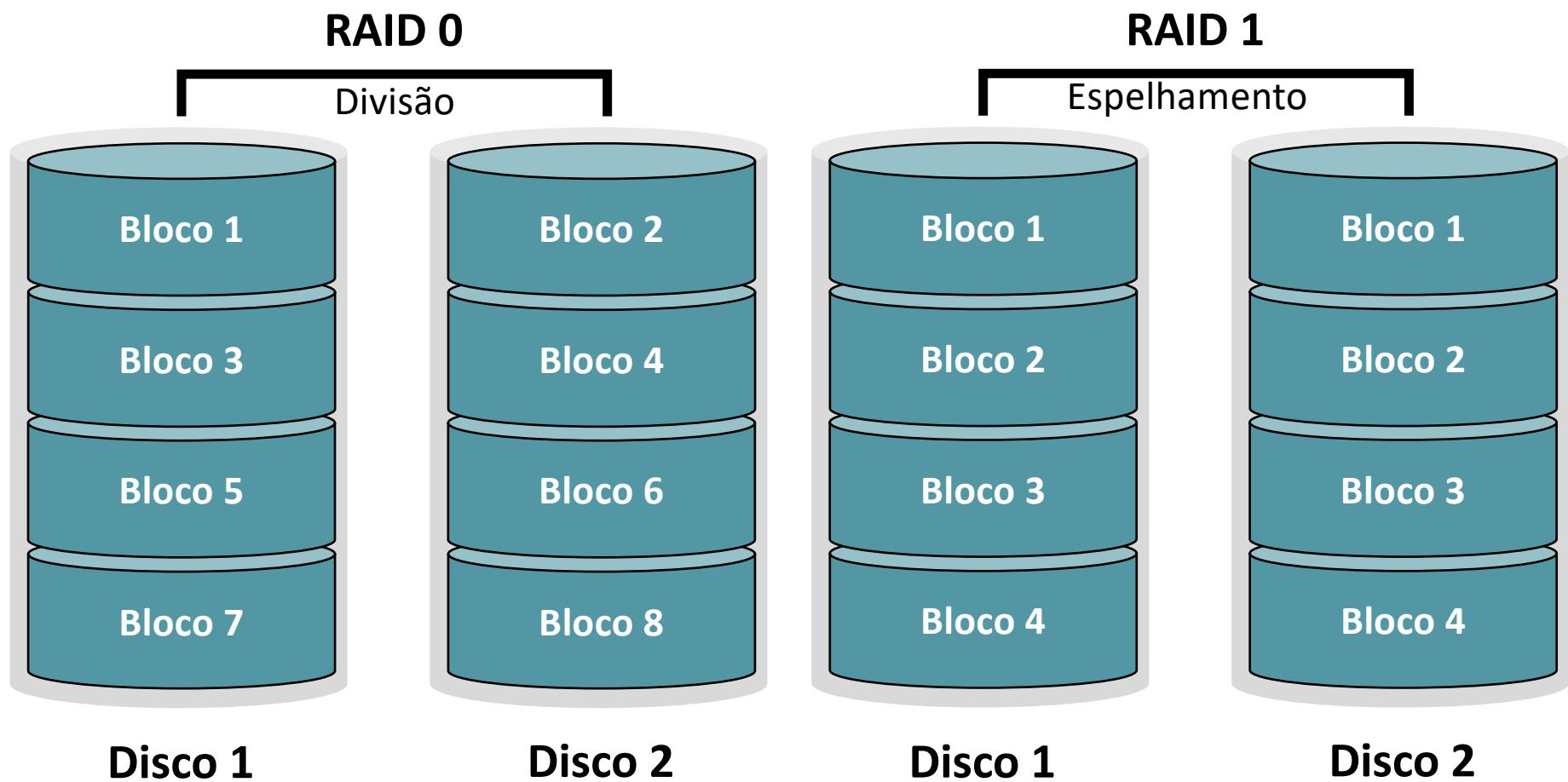


# Tipos de RAID

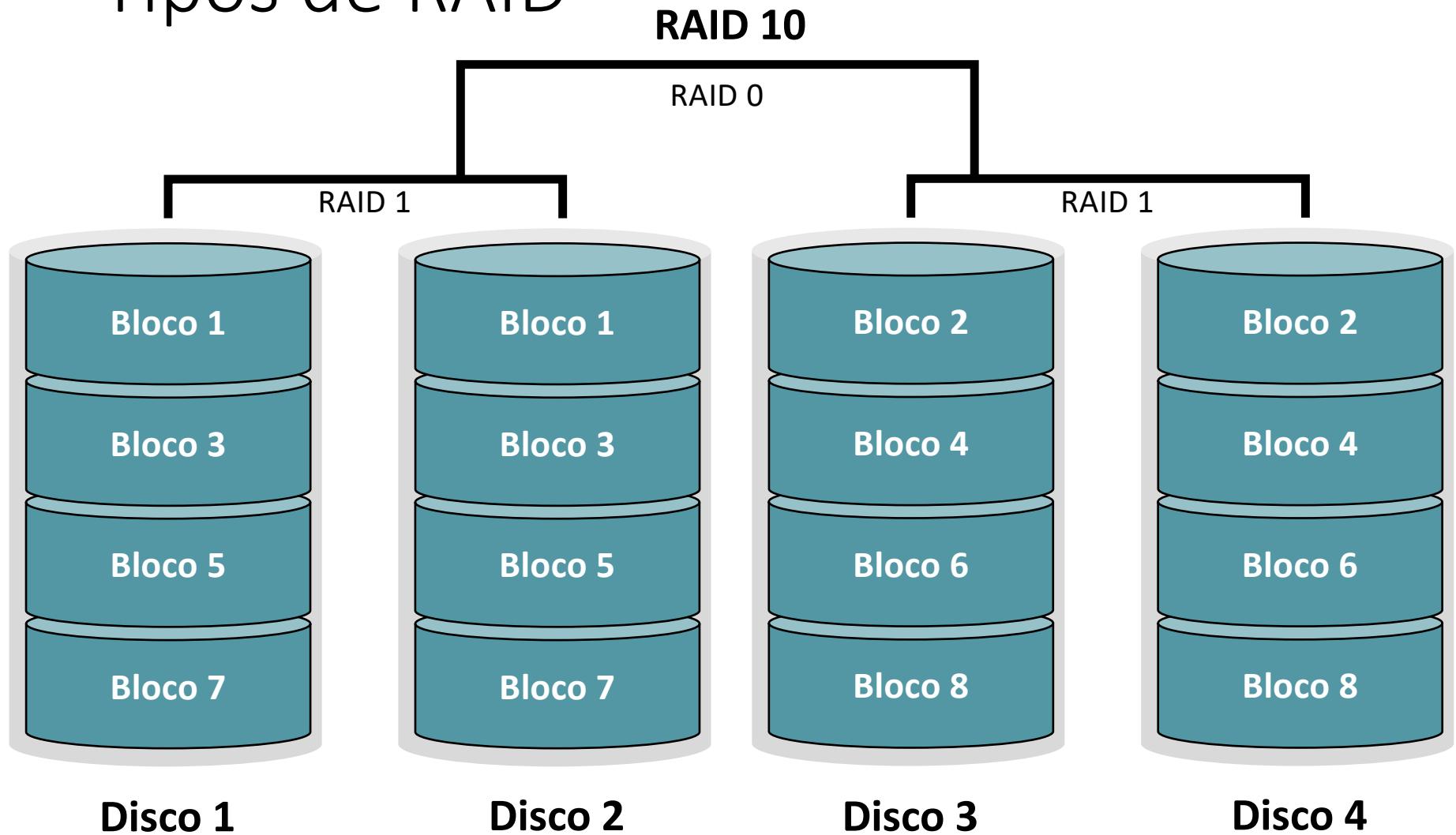
# Tipos de RAID



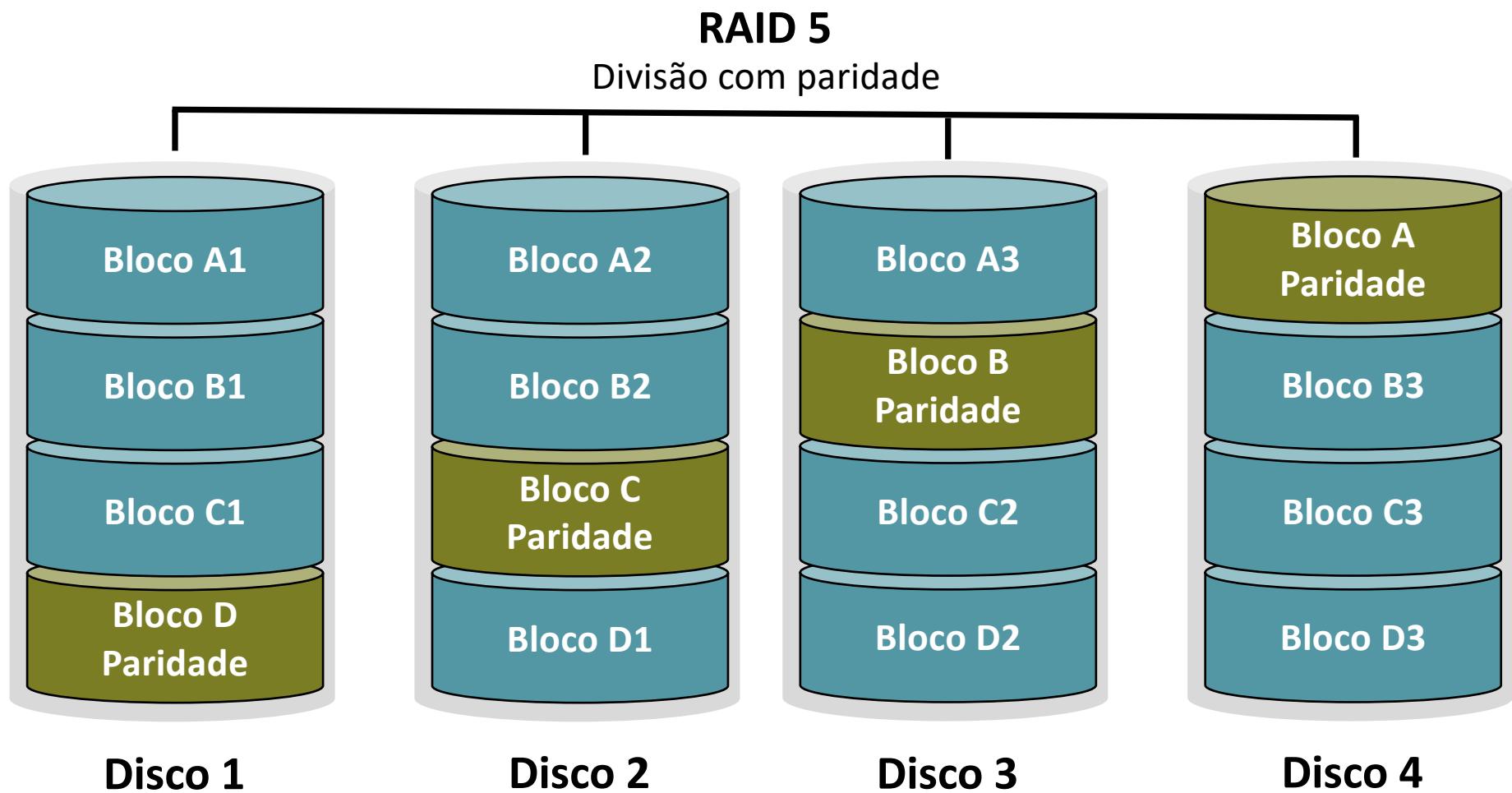
# Tipos de RAID



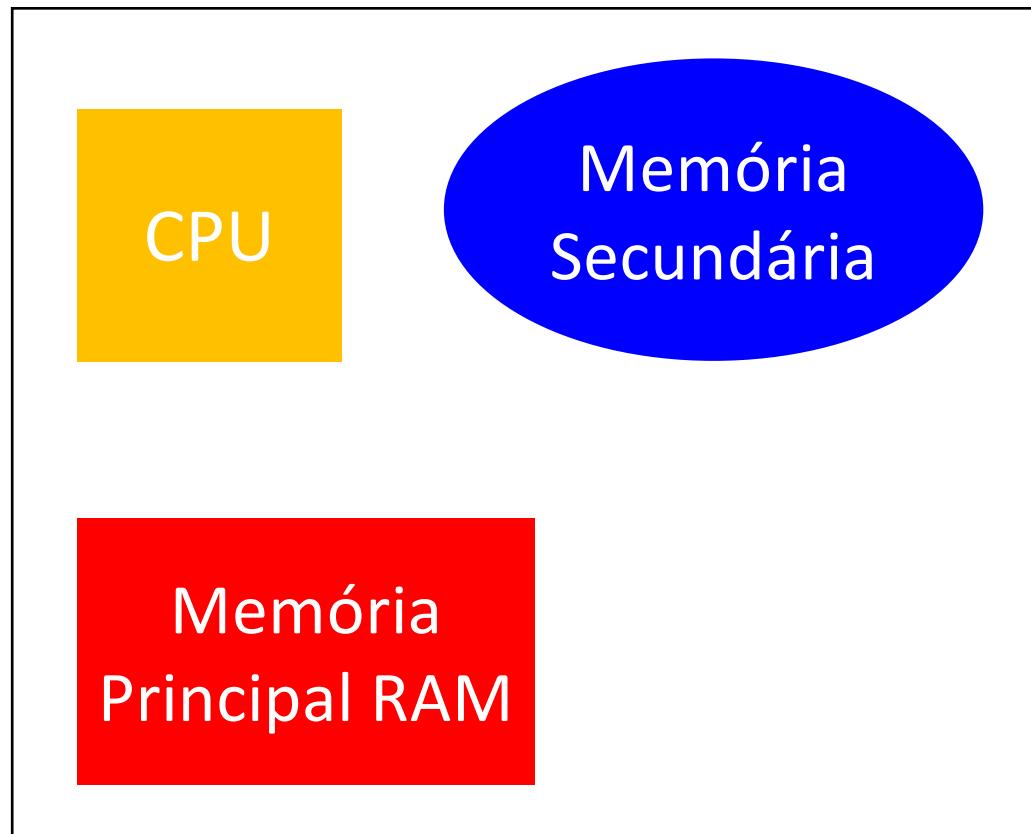
# Tipos de RAID



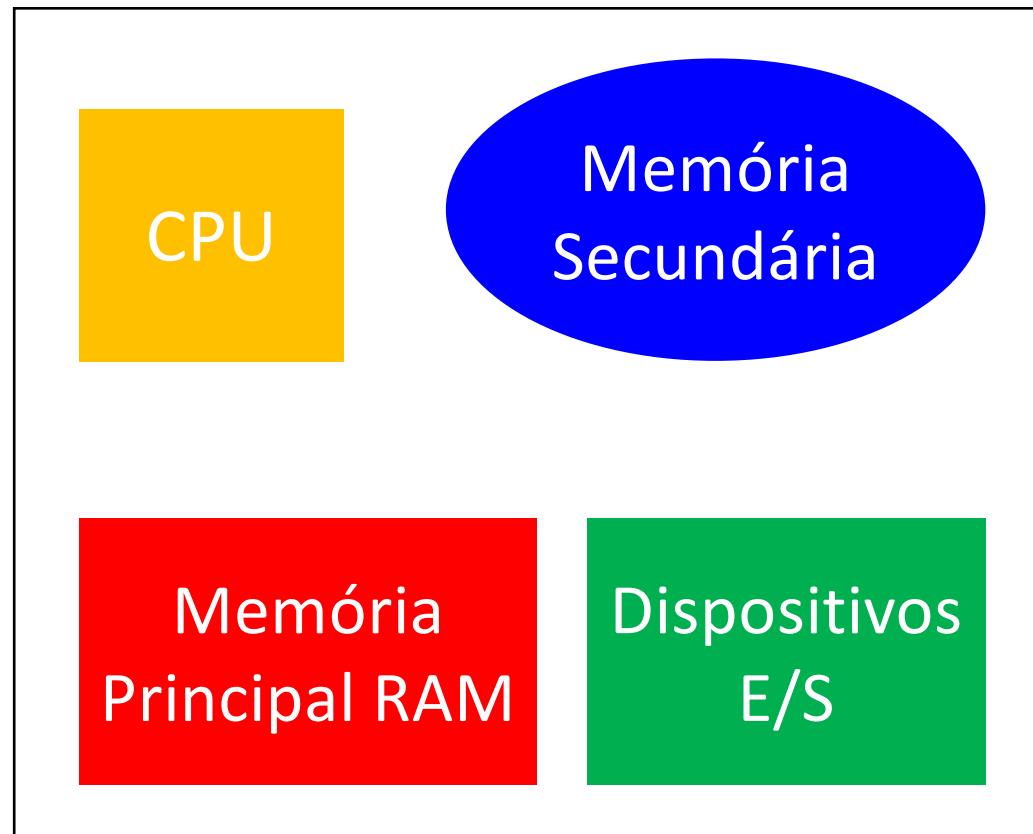
# Tipos de RAID



# Arquitetura de Computadores

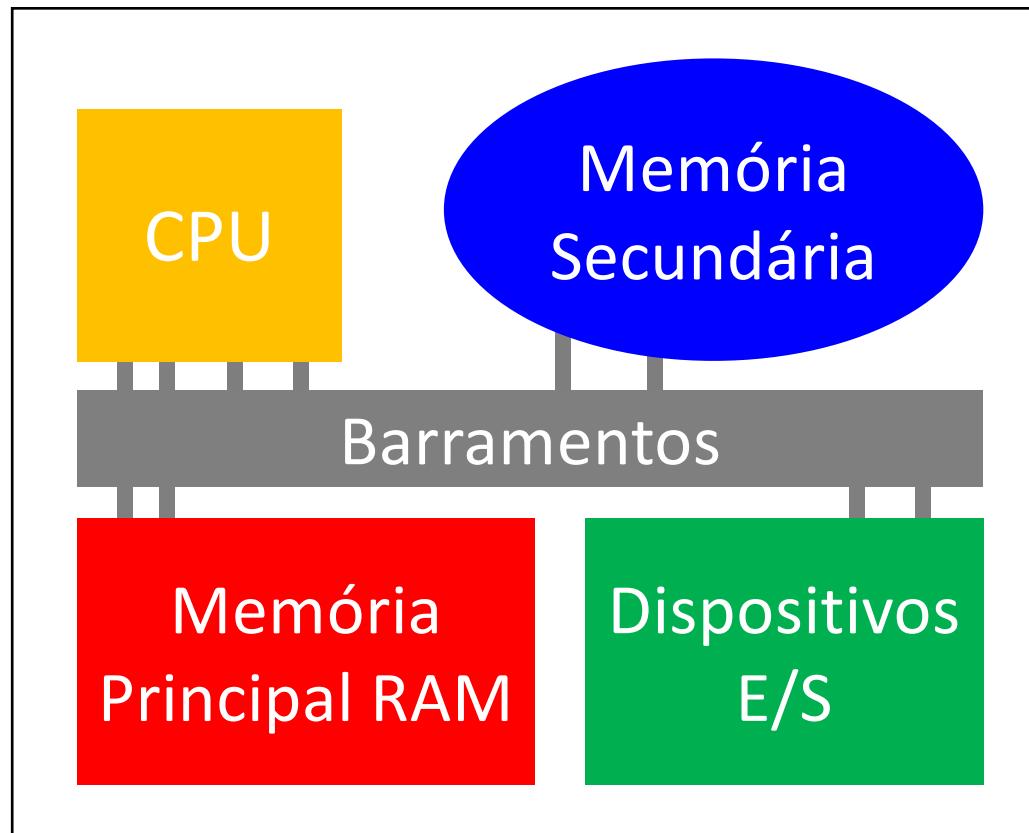


# Arquitetura de Computadores



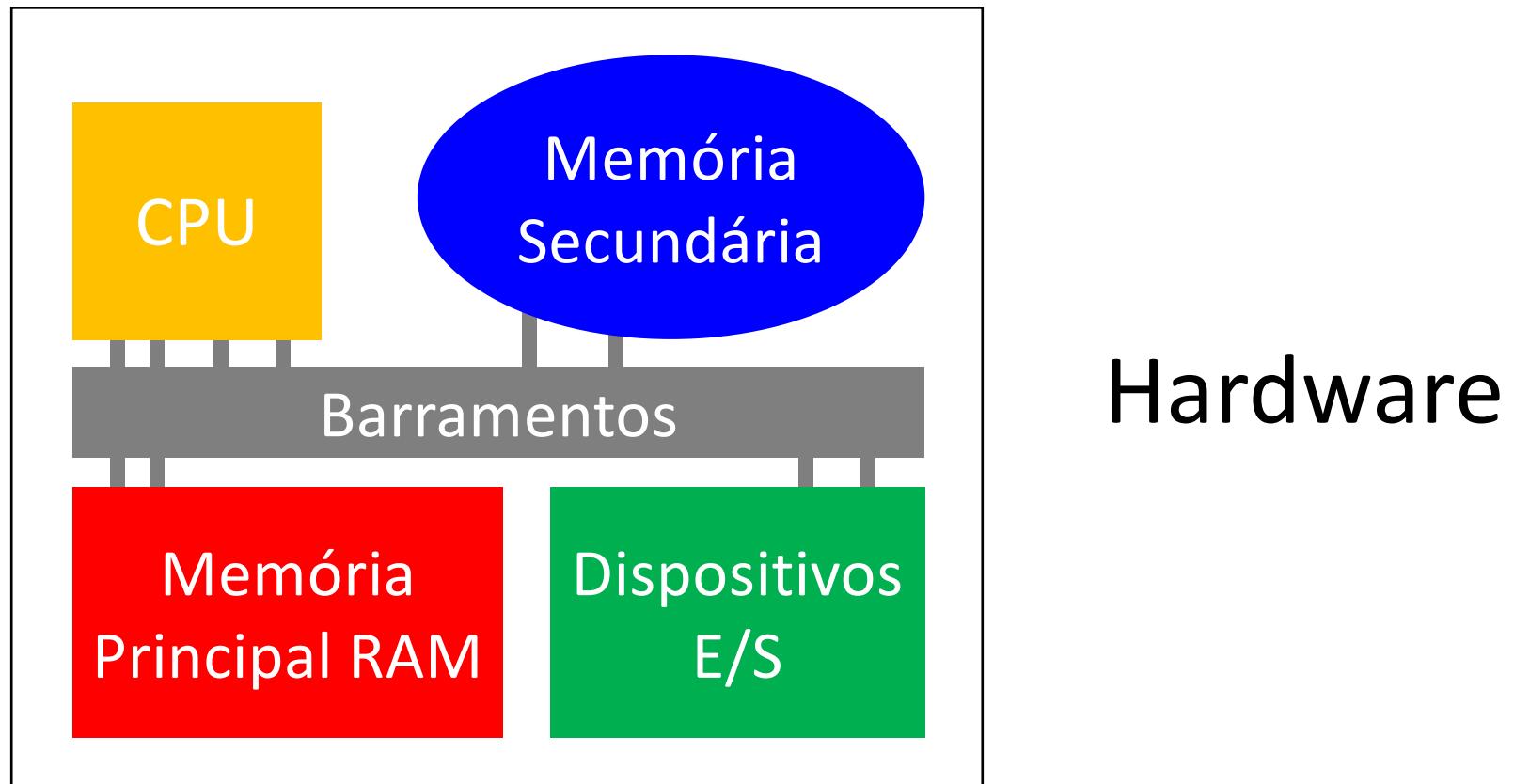
- Utilizados para entrada e saída de dados
  - Mouse
  - Teclado
  - Monitor
  - Impressora
  - Dispositivos USB

# Arquitetura de Computadores



- Conexões entre os componentes
- Conexões com velocidades e capacidades distintas
- Específicas para componente

# Arquitetura de Computadores



# Arquitetura de Computadores

Alto Nível

Baixo Nível

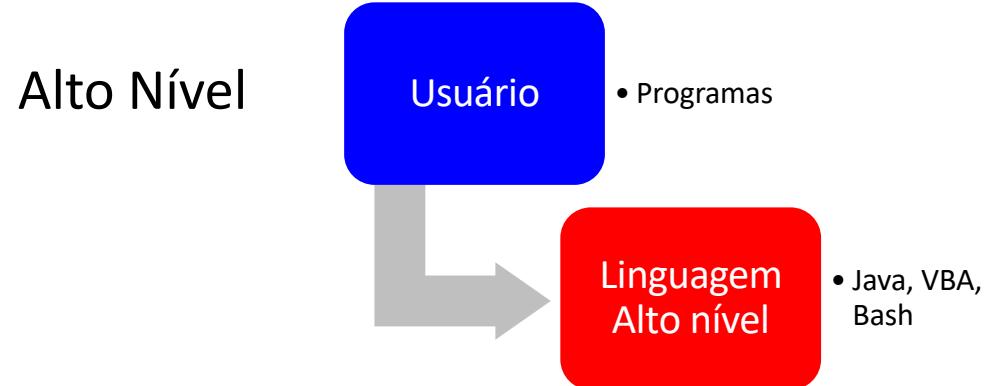
# Arquitetura de Computadores

Alto Nível



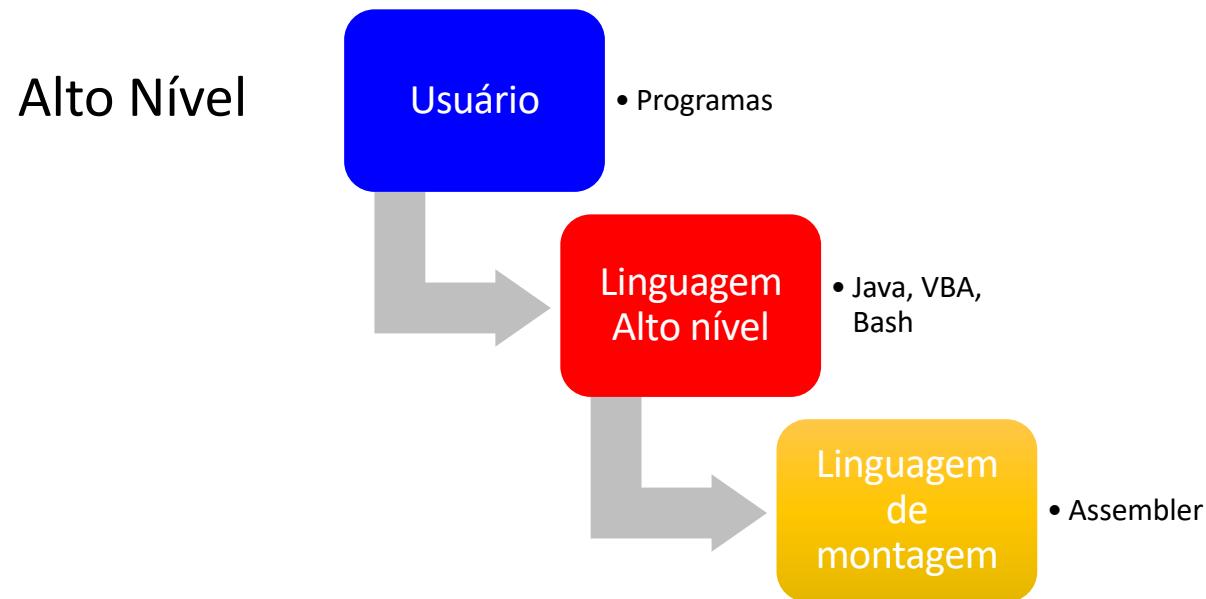
Baixo Nível

# Arquitetura de Computadores



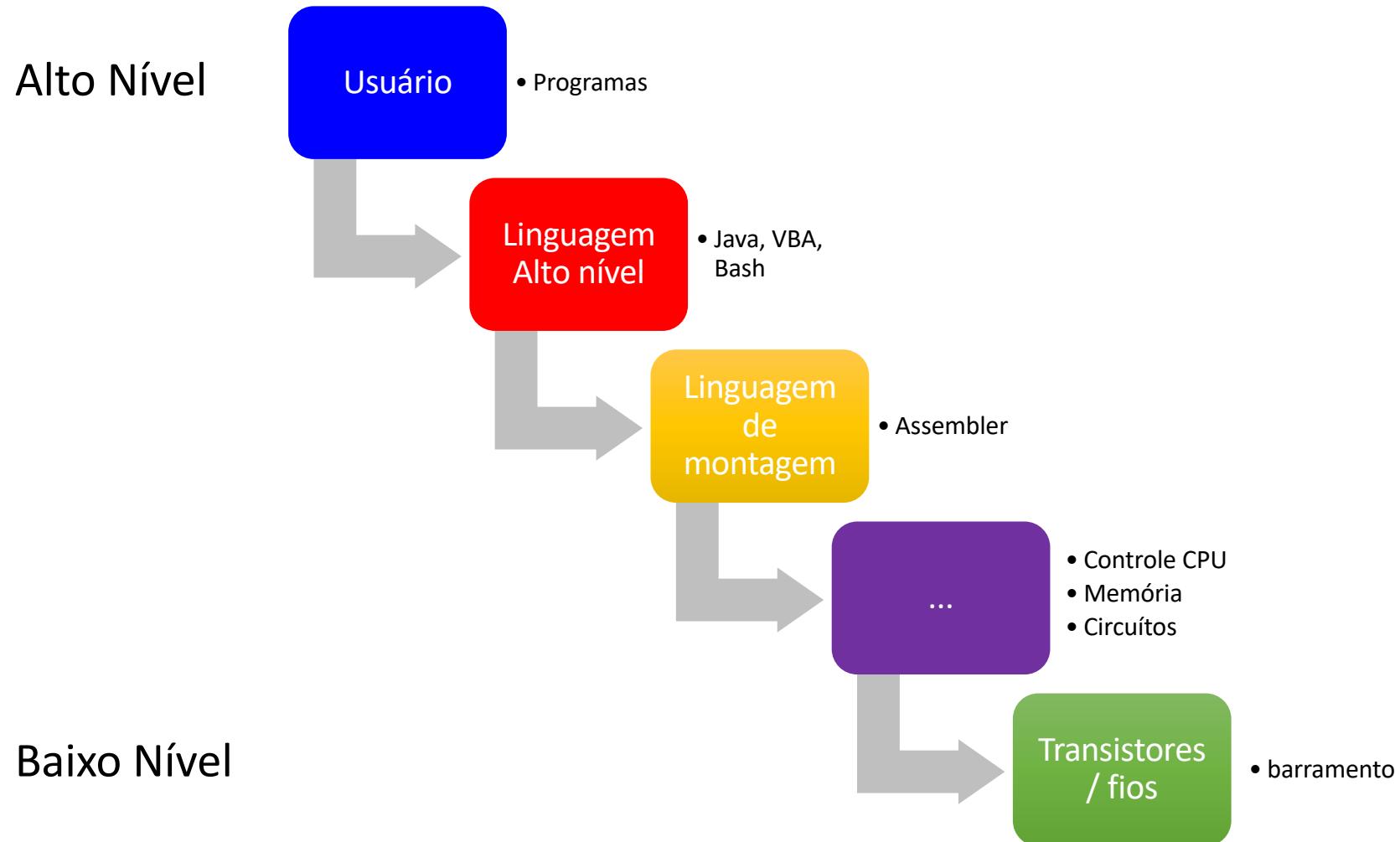
Baixo Nível

# Arquitetura de Computadores

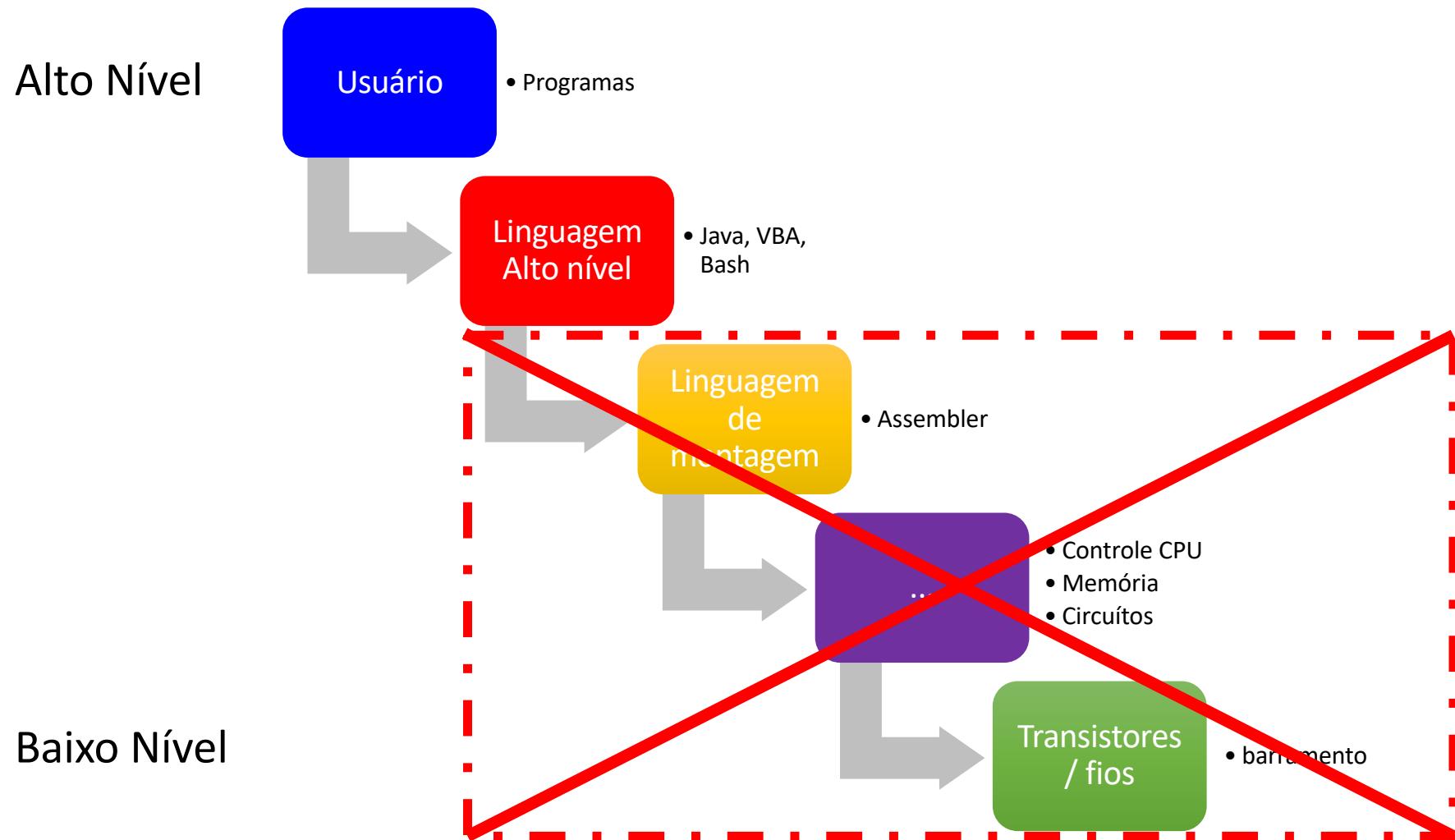


Baixo Nível

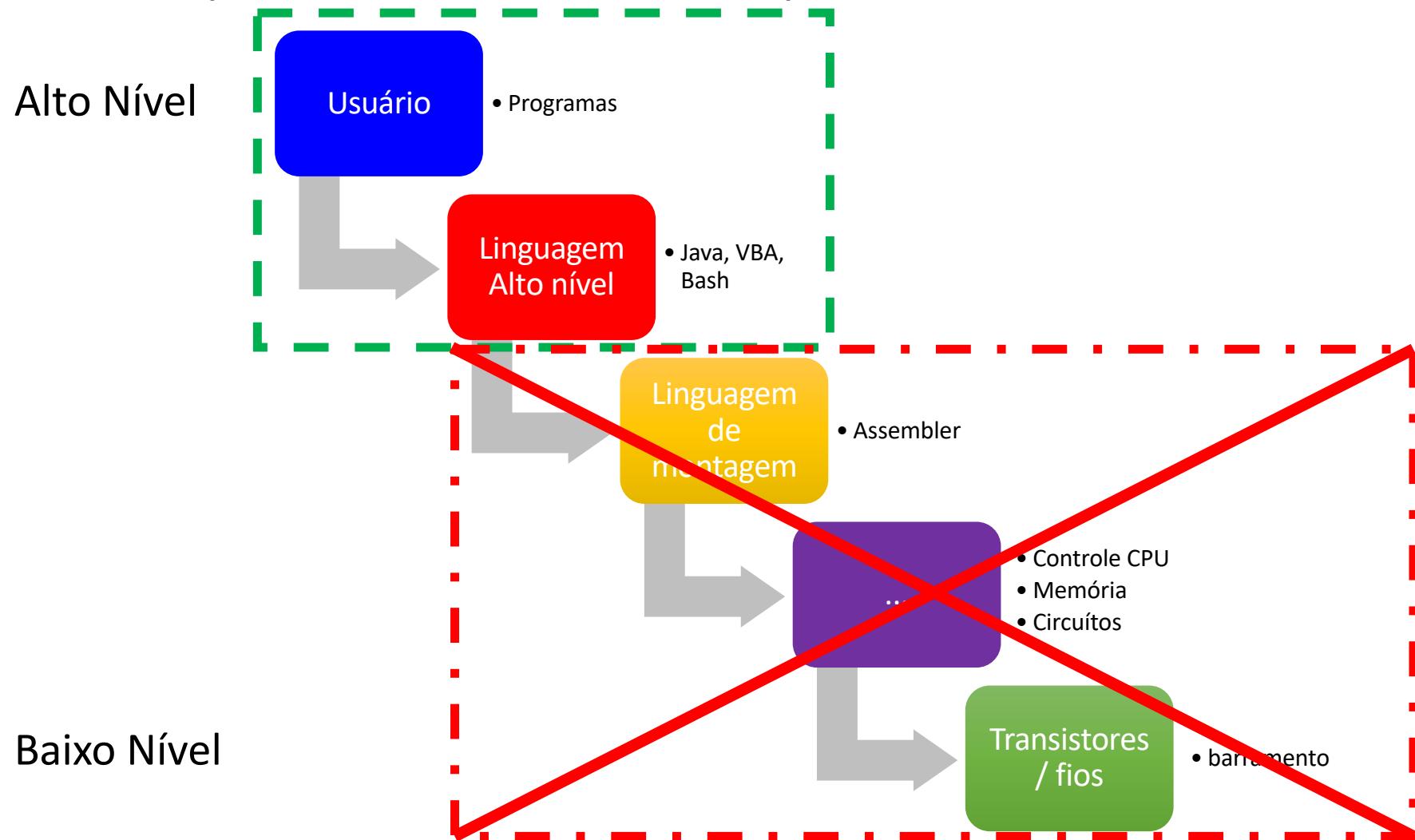
# Arquitetura de Computadores



# Arquitetura de Computadores



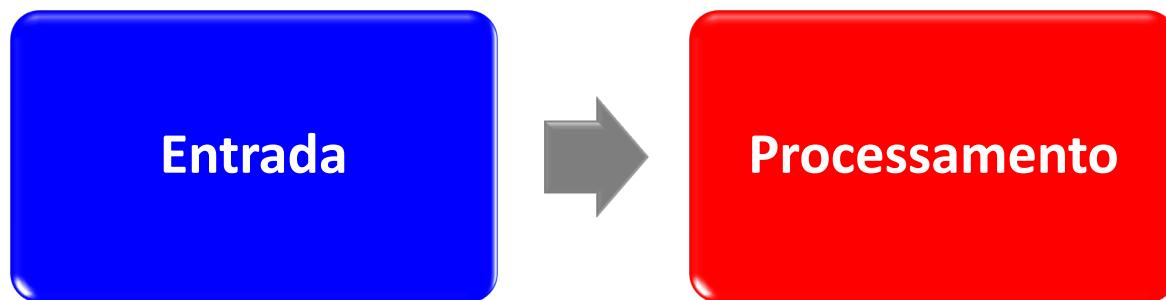
# Arquitetura de Computadores



# Arquitetura de Computadores

Entrada

# Arquitetura de Computadores



# Arquitetura de Computadores



# Arquitetura de Computadores



# Lógica

- Lógica (do grego λογική (logos) - raciocínio)
- Organização racional de um pensamento:
  - Se Chr1 > Chr10 E Chr10 > Chr20 Logo Chr1 > Chr20
- Raciocínio lógico é utilizado na construção de algoritmos
- Algoritmos são transformados em programas de computador utilizando uma linguagem de programação
- Computador é utilizado para resolver um dado problema com base no programa que foi desenvolvido

# Algoritmo

- Sequencia de etapas para se atingir um objetivo final
- Exemplos:
  - Receita de um bolo
  - Programação de uma viagem
  - Cálculo de uma equação
  - Métodos de um dado experimento
- Uma vez concebido, pode ser aplicado com qualquer linguagem de programação

# Algoritmo

- Cálculo da área de um triângulo:
  - Receber o valor da base
  - Receber o valor da altura
  - Realizar a operação base vezes altura
  - Exibir o resultado
- Classificar um triângulo:
  - Isóceles - dois lados iguais
  - Equilátero - 3 lados iguais
  - Escaleno - nenhum lado igual

# Tipos de Dados

- Tipos primitivos:
  - Numéricos:
    - Inteiro
      - {... , -3, -2, -1, 0, 1, 2, 3, ...}, nulo
    - Real
  - Literal
    - Caractere
      - Alfanuméricos (0-9), alfabéticos (A-Z, a-z) e especiais ()(\*^%\$#@!...)
  - Lógico
    - Verdadeiro ou falso
    - Sim ou não
    - 1 ou 0

# Operação com Numérico

- INT e INT = INT
- INT e REAL = REAL
- REAL e REAL = REAL

# Operação com Numérico

- INT e INT = INT
- INT e REAL = REAL
- REAL e REAL = REAL
- Exemplo:
  - $10 + 5,5 = 15,5$

# Operação com Numérico

- INT e INT = INT
- INT e REAL = REAL
- REAL e REAL = REAL
- Exemplo:
  - $10 + 5,5 = 15,5$
  - $1,1 * 2,2 = 2,42$

# Operação com Numérico

- INT e INT = INT
- INT e REAL = REAL
- REAL e REAL = REAL
- Exemplo:
  - $10 + 5,5 = 15,5$
  - $1,1 * 2,2 = 2,42$
  - $5 / 2 = 2$

# Operação com Numérico

- INT e INT = INT
- INT e REAL = REAL
- REAL e REAL = REAL
- Exemplo:
  - $10 + 5,5 = 15,5$
  - $1,1 * 2,2 = 2,42$
  - $5 / 2 = 2$  Resto 1

# Variáveis

- Estrutura para armazenar valores
- Constantes
  - Não se alteram durante a execução do algoritmo
  - $\pi$  (PI), Frequência alélica de alélhos raros e etc...
- Variáveis
  - Se alteram durante a execução do algoritmo
  - Base de um triângulo para calcular área de N triangulos
- Possui nome
  - Letras, números, e caractere “\_” (under score)

# Operadores

- Utilizados para representar expressões de cálculo, comparação, condição ou atribuição
- Atribuição
  - Atribui um valor para uma variável (= ou <-)
- Aritméticos
  - Utilizado para realizar operações matemáticas (+, -, \*, /)
- Relacionais
  - Utilizado para realizar comparações (>, >=, <, <=, ==, != <>)
- Lógicos
  - Operações com valores lógicos (&&, ||, !)

# Tabela Verdade

Operador	1º Operador A	2º Operador B	Resultado C	Operação
E / AND	Verdadeiro	Verdadeiro	Verdadeiro	$C = A \&\& B$
	Verdadeiro	Falso	Falso	
	Falso	Verdadeiro	Falso	
	Falso	Falso	Falso	
OU / OR	Verdadeiro	Verdadeiro	Verdadeiro	$C = A    B$
	Verdadeiro	Falso	Verdadeiro	
	Falso	Verdadeiro	Verdadeiro	
	Falso	Falso	Falso	

E / AND - Somente será VERDADEIRO se todos os operadores forem VERDADEIROS

OU / OR - Somente será FALSO se todos os operadores forem FALSOS

# Estrutura de Controle

- Decisão
  - Verificar uma condição para tomada de decisão
  - Lado A = Lado B != Lado C, então Isóceles
- Simples
  - Se isso / Então aquilo
- Composta
  - Se isso / Então aquilo / Senão aquilo outro
- Encadeada
  - Se isso / Então aquilo / Senão (Se isso / Então aquilo / Senão aquilo outro)

# Classificação de Raridade (SE)

- Nova
  - $AF = 0$
- Muito Rara
  - $AF > 0 \ \&\& AF \leq 0.001$
- Rara
  - $AF > 0.001 \ \&\& AF \leq 0.01$
- Comum
  - $AF > 0.01$

# Classificação do Triângulo (SE + E)

Se  $A = B \ \&\& B = C$

Então equilátero

Senão

Se  $A \neq B \ \&\& A \neq C \ \&\& B \neq C$

Então escaleno

Senão

isóceles

# Classificação de Raridade

Se AF = 0 Então

Nova

Senão

Se AF <= 0.001 Então

Muito Rara

Senão

Se AF <= 0.01 Então

Rara

Senão

Comum

# Estrutura de Repetição

- Utilizado para repetir um segmento do programa
- Teste de saída no início
  - Enquanto - Faça
- Teste de Saída no final
  - Repita - Até
- Com variável de controle
  - Para - Até - Faça

# Vetor e Matriz

- Vetor
  - Conjunto unidimensional de variáveis do mesmo tipo
  - Agrupados continuamente na memória
  - Acessado por um índice
  - Uma coluna ou uma linha do excel
- Matriz
  - Conjunto multidimensional de variáveis do mesmo tipo
  - Acessado por multiplos indices
  - Planilha do excel, arquivo do excel com multiplas planilhas, uma pasta com multiplos arquivos de excel ...

# Modularização

- Conjunto de instruções para realizar uma determinada tarefa
- Procedimento
  - Não há retorno de valor
  - O fluxo do programa é desviado para esse procedimento e retorna ao ponto de chamada
- Função
  - Há retorno de valor
  - O valor de retorno é usado no programa
- Atributos
  - Valores que são utilizados na execução desse bloco

# Função calcularAreaTriangulo

Fluxo principal

    base = 10

    altura = 5

    area = calcularAreaTriangulo(base, altura)

    Exibir area

Fim Fluxo principal

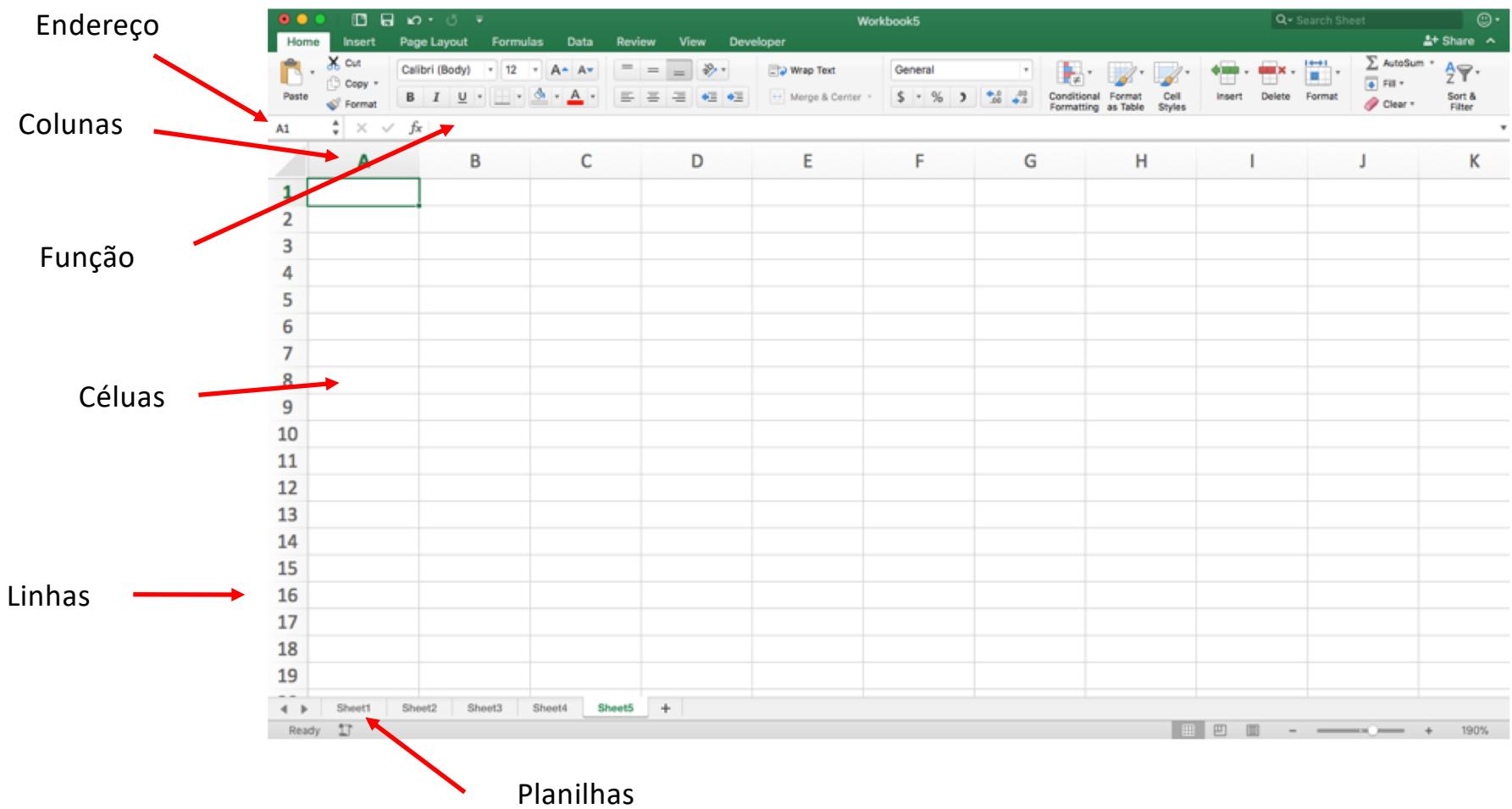
Função calcularAreaTriangulo(base, altura)

    area = base \* altura

    retorna area

Fim Função

# Editor de Planilhas (Excel)



# Editor de Planilhas (Excel)

- Arquivo
  - Planilhas
    - Colunas
    - Linhas
      - Células
      - Dados
- Células possuem um endereço (Coluna X Linha)
- Pode-se realizar operações entre diferentes células
- Existem diversas funções pré-programadas
  - =SOMA(A1:A20) - Vetor
  - =SOMA(A1, A2, A3, ..., A10) - variáveis

# Função SE()

- =SE(TESTE LÓGICO; VALOR SE VERDADEIRO; VALOR SE FALSO)
- =IF(TESTE LÓGICO, VALOR SE VERDADEIRO, VALOR SE FALSO)
- TESTE LÓGICO
  - Necessariamente precisa ter como retorno um valor lógico
    - VERDADEIRO OU FALSE
- VALOR DE RETORNO
  - Se for literal precisa estar entre aspas “VALOR”
  - Se for numérico ou uma variável não pode ter aspas
  - Pode ser uma outra função qualquer inclusive um SE()

# Função SE()

	A	B	C
1	Variante	AF	CLASSIFICAÇÃO
2	1	0.01535088	
3	2	0.00527241	
4	3	0.02617801	
5	4	0	
6	5	0.04812834	
7	6	0.00348432	
8	7	0.0006865	
9	8	0.01347305	
10	9	0.01703578	
11	10	0.01046025	
12	11	0.0095057	
13	12	0	
14	13	0.00598086	
15	14	0.00154799	
16	15	0.0091638	
17	16	0.05586592	
18	17	0.03773585	
19	18	0.00323625	
20	19	0.01084599	
21	20	0.00825472	

# Função SE()

Se AF = 0 Então

Nova

Senão

Se AF <= 0.001 Então

Muito Rara

Senão

Se AF <= 0.01 Então

Rara

Senão

Comum

	A	B	C
1	Variante	AF	CLASSIFICAÇÃO
2	1	0.01535088	
3	2	0.00527241	
4	3	0.02617801	
5	4	0	
6	5	0.04812834	
7	6	0.00348432	
8	7	0.0006865	
9	8	0.01347305	
10	9	0.01703578	
11	10	0.01046025	
12	11	0.0095057	
13	12	0	
14	13	0.00598086	
15	14	0.00154799	
16	15	0.0091638	
17	16	0.05586592	
18	17	0.03773585	
19	18	0.00323625	
20	19	0.01084599	
21	20	0.00825472	

# Função SE()

Se AF = 0 Então

Nova

Senão

Se AF <= 0.001 Então

Muito Rara

Senão

Se AF <= 0.01 Então

Rara

Senão

Comum

	A	B	C
1	Variante	AF	CLASSIFICAÇÃO
2	1	0.01535088	COMUM
3	2	0.00527241	
4	3	0.02617801	
5	4	0	
6	5	0.04812834	
7	6	0.00348432	
8	7	0.0006865	
9	8	0.01347305	
10	9	0.01703578	
11	10	0.01046025	
12	11	0.0095057	
13	12	0	
14	13	0.00598086	
15	14	0.00154799	
16	15	0.0091638	
17	16	0.05586592	
18	17	0.03773585	
19	18	0.00323625	
20	19	0.01084599	
21	20	0.00825472	

=IF(B2=0,"NOVA",IF(B2<=0.001,"MUITO RARA",IF(B2<=0.01,"RARA","COMUM")))

# Função SE()

Se AF = 0 Então

Nova

Senão

Se AF <= 0.001 Então

Muito Rara

Senão

Se AF <= 0.01 Então

Rara

Senão

Comum

	A	B	C
1	Variante	AF	CLASSIFICAÇÃO
2	1	0.01535088	COMUM
3	2	0.00527241	RARA
4	3	0.02617801	COMUM
5	4	0	NOVA
6	5	0.04812834	COMUM
7	6	0.00348432	RARA
8	7	0.0006865	MUITO RARA
9	8	0.01347305	COMUM
10	9	0.01703578	COMUM
11	10	0.01046025	COMUM
12	11	0.0095057	RARA
13	12	0	NOVA
14	13	0.00598086	RARA
15	14	0.00154799	RARA
16	15	0.0091638	RARA
17	16	0.05586592	COMUM
18	17	0.03773585	COMUM
19	18	0.00323625	RARA
20	19	0.01084599	COMUM
21	20	0.00825472	RARA

=IF(B2=0,"NOVA",IF(B2<=0.001,"MUITO RARA",IF(B2<=0.01,"RARA","COMUM")))

# Função E()

- =E(TESTE LÓGICO 1; TESTE LÓGICO 2; ...)
- = AND(TESTE LÓGICO 1, TESTE LÓGICO 2, ...)
- TESTE LÓGICO
  - Necessariamente precisa ter como retorno um valor lógico
    - VERDADEIRO OU FALSE

Operador	1º Operador A	2º Operador B	Resultado C	Operação
E / AND	Verdadeiro	Verdadeiro	Verdadeiro	C = A && B
	Verdadeiro	Falso	Falso	
	Falso	Verdadeiro	Falso	
	Falso	Falso	Falso	

E / AND - Somente será VERDADEIRO se todos os operadores forem VERDADEIROS

# Função SE() + E()

	A	B	C	D	E
1	EXEMPLO	A	B	C	CLASSIFICAÇÃO
2	1	7	6	6	
3	2	10	5	8	
4	3	8	8	8	

# Função SE() + E()

	A	B	C	D	E
1	EXEMPLO	A	B	C	CLASSIFICAÇÃO
2	1	7	6	6	
3	2	10	5	8	
4	3	8	8	8	

Se  $A = B \ \&\& B = C$

Então equilátero

Senão

Se  $A \neq B \ \&\& A \neq C \ \&\& B \neq C$

Então escaleno

Senão

isóceles

# Função SE() + E()

	A	B	C	D	E
1	EXEMPLO	A	B	C	CLASSIFICAÇÃO
2	1	7	6	6	ISOCELES
3	2	10	5	8	
4	3	8	8	8	

Se A = B && B = C

Então equilátero

Senão

Se A != B && A != C && B != C

Então escaleno

Senão

isóceles

```
=IF(AND(B2=C2,C2=D2),"E  
QUILATERO",IF(AND(B2<>  
C2,B2<>D2,C2<>D2),"ESCA  
LENO","ISOCELES"))
```

# Função SE() + E()

	A	B	C	D	E
1	EXEMPLO	A	B	C	CLASSIFICAÇÃO
2	1	7	6	6	ISOCELES
3	2	10	5	8	ESCALENO
4	3	8	8	8	EQUILATERO

Se A = B && B = C

Então equilátero

Senão

Se A != B && A != C && B != C

Então escaleno

Senão

isóceles

```
=IF(AND(B2=C2,C2=D2),"E  
QUILATERO",IF(AND(B2<>  
C2,B2<>D2,C2<>D2),"ESCA  
LENO","ISOCELES"))
```

# Função OU()

- =OU(TESTE LÓGICO 1; TESTE LÓGICO 2; ...)
- = OR(TESTE LÓGICO 1, TESTE LÓGICO 2, ...)
- TESTE LÓGICO
  - Necessariamente precisa ter como retorno um valor lógico
    - VERDADEIRO OU FALSE

Operador	1º Operador A	2º Operador B	Resultado C	Operação
E / AND	Verdadeiro	Verdadeiro	Verdadeiro	$C = A \   \ B$
	Verdadeiro	Falso	Verdadeiro	
	Falso	Verdadeiro	Verdadeiro	
	Falso	Falso	Falso	

OU / OR - Somente será FALSO se todos os operadores forem FALSOS

# Função NÃO()

- =NÃO(TESTE LÓGICO)
- = NOT(TESTE LÓGICO)
- Negação de um valor lógico
- =NÃO(VERDADEIRO)
  - FALSO
- =NÃO(FALSO)
  - Verdadeiro

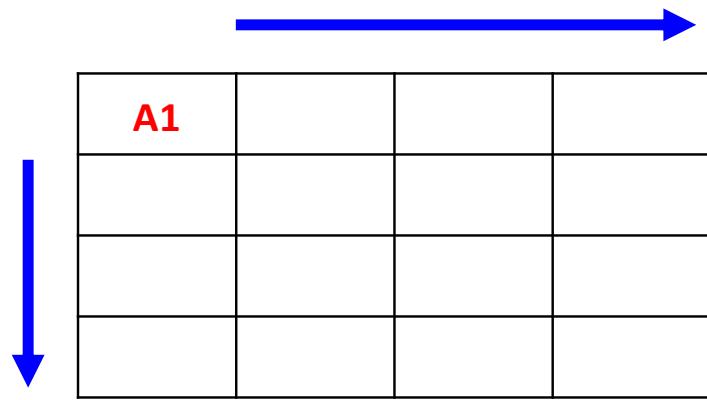
# TRavar INCREMENTAÇÃO

- \$
- Usado para não incrementar linhas e/ou colunas
- \$A\$1
  - Todas as células receberão conteúdo de A1
- \$A1
  - Todas as células receberão conteúdo da coluna A
- A\$1
  - Todas as células receberão conteúdo da linha 1

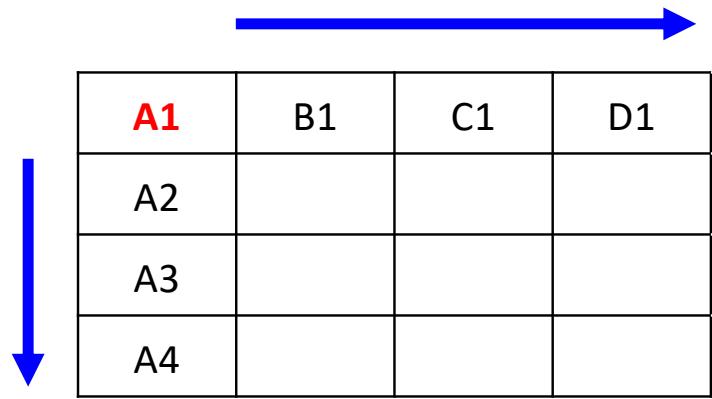
# TRAVAR INTERVALOS

A1			

# TRAVAR INTERVALOS



# TRAVAR INTERVALOS



# TRAVAR INTERVALOS



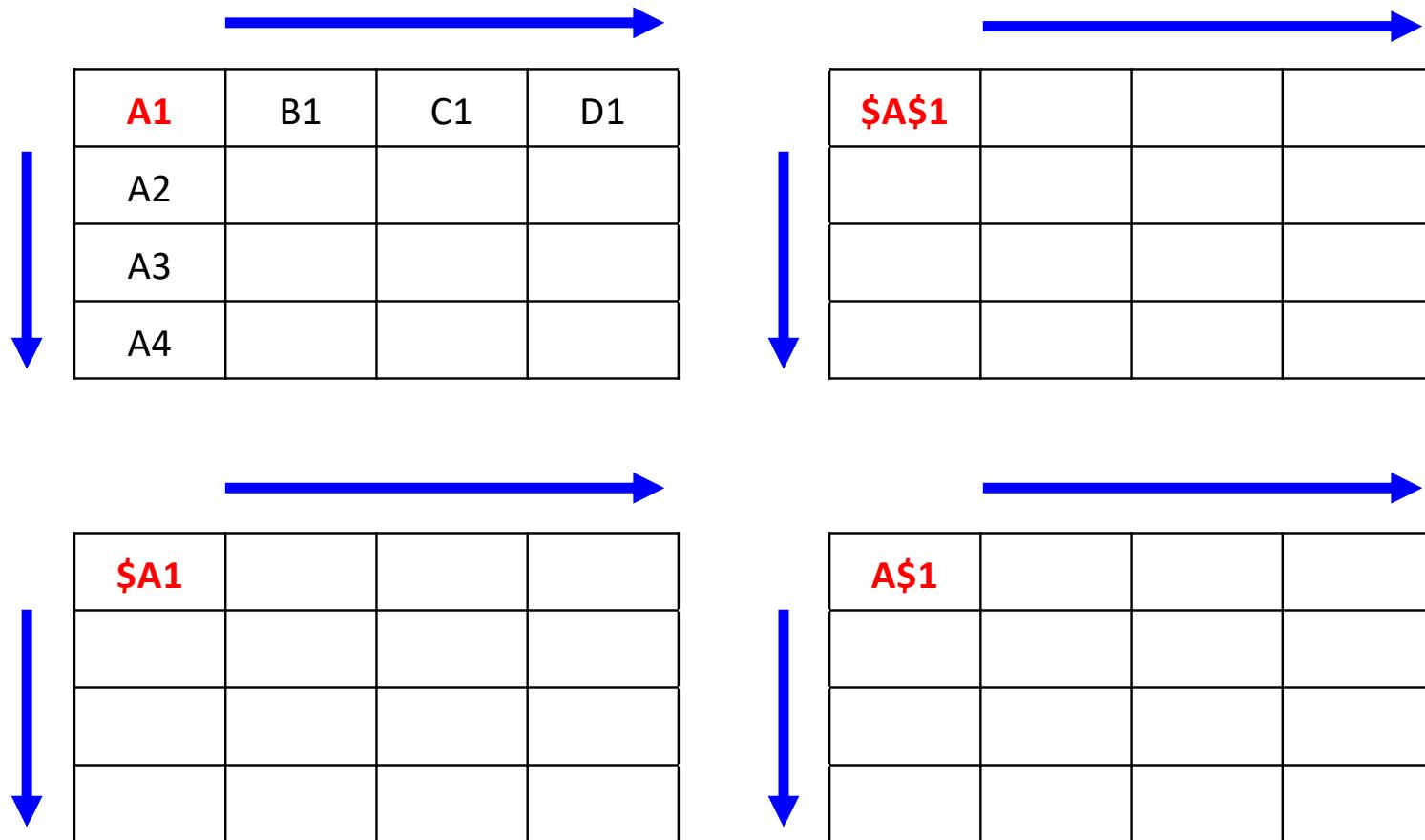
A1	B1	C1	D1
A2			
A3			
A4			

\$A\$1			

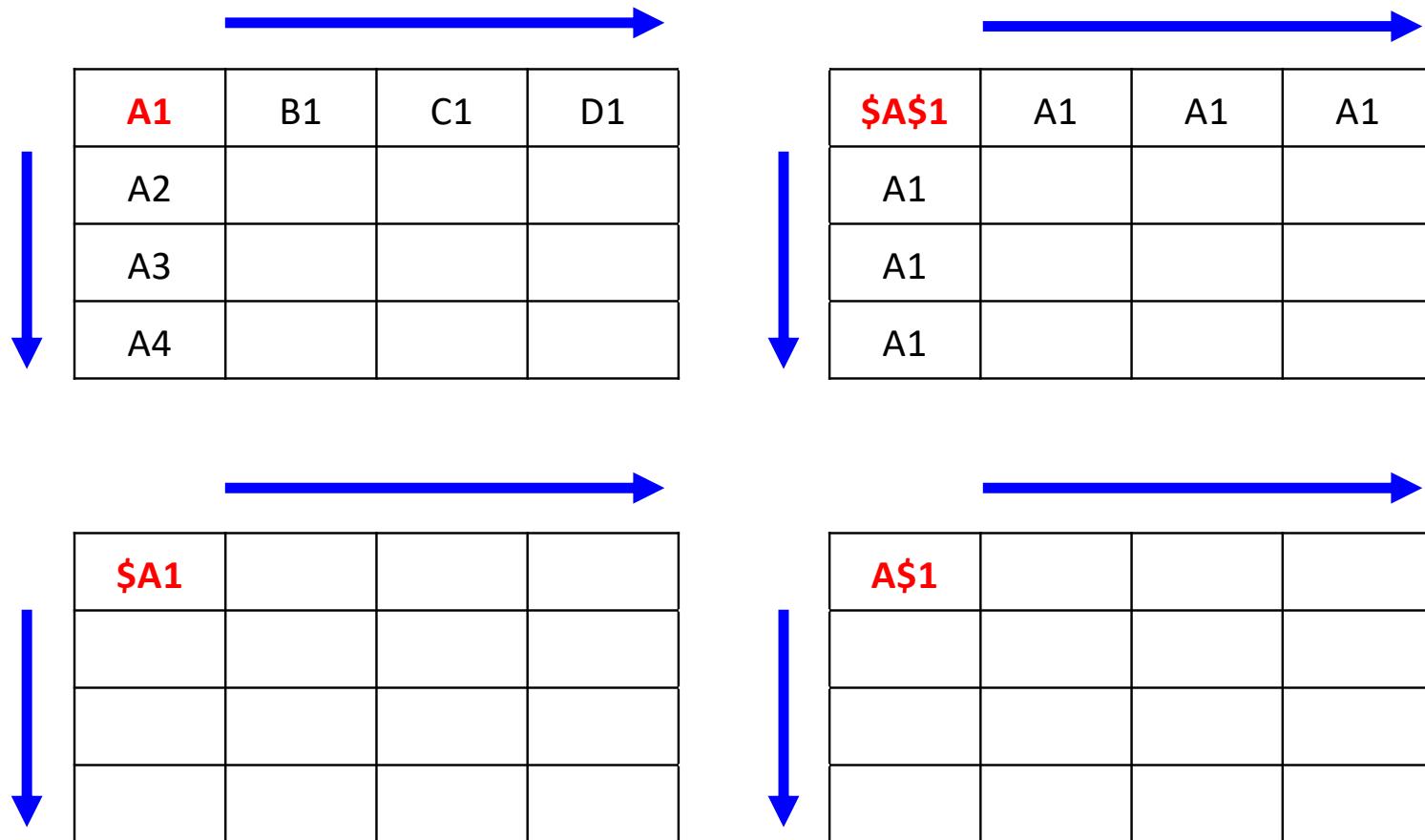
\$A1			

A\$1			

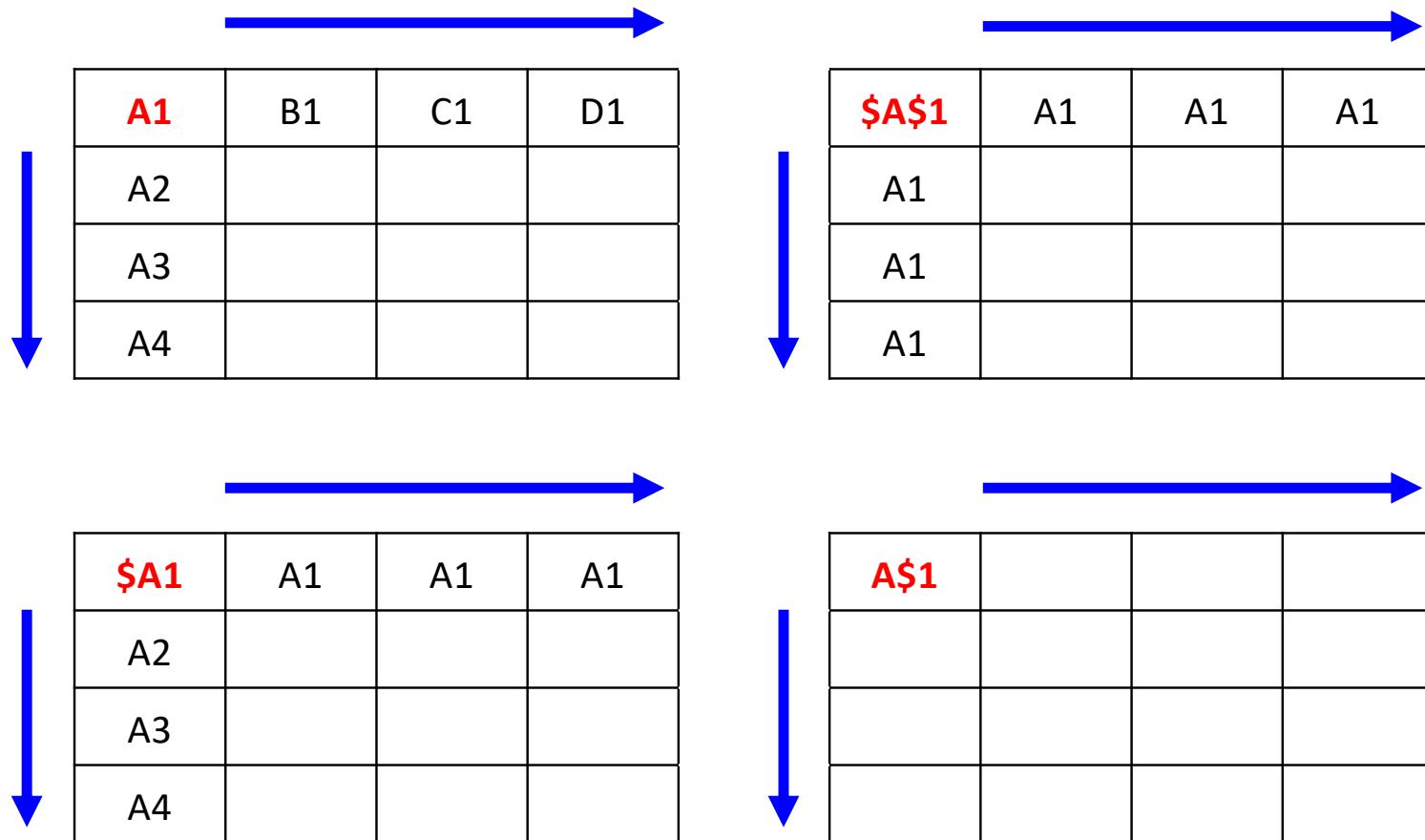
# TRAVAR INTERVALOS



# TRAVAR INTERVALOS



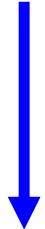
# TRAVAR INTERVALOS



# TRAVAR INTERVALOS



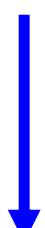
A1	B1	C1	D1
A2			
A3			
A4			



\$A\$1	A1	A1	A1
A1			
A1			



\$A1	A1	A1	A1
A2			
A3			
A4			



A\$1	B1	C1	D1
A2			
A3			
A4			



# Manipulação de Texto

- =DIREITA(TEXTO; TAMANHO)
- =RIGHT(TEXTO, TAMANHO)
- =ESQUERDA(TEXTO; TAMANHO)
- =LEFT(TEXTO, TAMANHO)
- =TAMANHO(TEXTO)
- =LEN(TEXTO)
- =CONCATENAR(TEXTO1; TEXTO2; TEXTO3; ...)
- =CONCATENATE(TEXTO1, TEXTO2, TEXTO3, ...)

# Função PROC.V()

- =PROC.V(VALOR; TABELA; COLUNA; INTERVALO)
- =VLOOKUP(VALOR, TABELA, COLUNA, INTERVALO)
- Busca de um termo em uma tabela e retorna o valor correspondente à coluna indicada
- INTERVALO
  - Valor lógico para indicar se deve buscar o valor mais próximo (ordem crescente)
- =PROC.H(VALOR; TABELA; LINHA; INTERVALO)
- HLOOKUP(VALOR, TABELA, LINHA, INTERVALO)

# Função PROC.V()

- BUSCAR CHR DO ALPK2

# Função PROC.V()

- BUSCAR CHR DO ALPK2

	A	B	C	D	E
1	GENE	CHR	INÍCIO	FIM	FITA
2	TTC9	chr14	71108504	71142077	+
3	LOC100129722	chr9	140144671	140147949	-
4	ATL1	chr14	50999800	51099786	+
5	LOC100506731	chr14	85990776	85998103	-
6	AHI1	chr6	135604670	135818903	-
7	FAM155B	chrX	68725078	68752351	+
8	RNA5SP343	chr11	73909793	73909899	-
9	NKX2-2	chr20	21491655	21494664	-
10	ALPK2	chr18	56148479	56296189	-
11	MTHFD2P7	chr3	179182134	179183119	+
12	FOXN2	chr2	48541783	48606434	+
13	ZNF14	chr19	19821281	19843921	-
14	NCF4	chr22	37257030	37274059	+
15	TRNAS3	chr6	27473607	27473688	-
16	ORC5	chr7	103766788	103848495	-

# Função PROC.V()

- BUSCAR CHR DO ALPK2
- =PROC.V(ALPK2;  
A2:E16; 2; FALSO)

	A	B	C	D	E
1	GENE	CHR	INÍCIO	FIM	FITA
2	TTC9	chr14	71108504	71142077	+
3	LOC100129722	chr9	140144671	140147949	-
4	ATL1	chr14	50999800	51099786	+
5	LOC100506731	chr14	85990776	85998103	-
6	AHI1	chr6	135604670	135818903	-
7	FAM155B	chrX	68725078	68752351	+
8	RNA5SP343	chr11	73909793	73909899	-
9	NKX2-2	chr20	21491655	21494664	-
10	ALPK2	chr18	56148479	56296189	-
11	MTHFD2P7	chr3	179182134	179183119	+
12	FOXN2	chr2	48541783	48606434	+
13	ZNF14	chr19	19821281	19843921	-
14	NCF4	chr22	37257030	37274059	+
15	TRNAS3	chr6	27473607	27473688	-
16	ORC5	chr7	103766788	103848495	-

# Função PROC.V()

- BUSCAR CHR DO ALPK2
- =PROC.V(ALPK2;  
A2:E16; **2**; FALSO)

	A	B	C	D	E
1	GENE	CHR	INÍCIO	FIM	FITA
2	TTC9	chr14	71108504	71142077	+
3	LOC100129722	chr9	140144671	140147949	-
4	ATL1	chr14	50999800	51099786	+
5	LOC100506731	chr14	85990776	85998103	-
6	AHI1	chr6	135604670	135818903	-
7	FAM155B	chrX	68725078	68752351	+
8	RNA5SP343	chr11	73909793	73909899	-
9	NKX2-2	chr20	21491655	21494664	-
10	ALPK2	chr18	56148479	56296189	-
11	MTHFD2P7	chr3	179182134	179183119	+
12	FOXN2	chr2	48541783	48606434	+
13	ZNF14	chr19	19821281	19843921	-
14	NCF4	chr22	37257030	37274059	+
15	TRNAS3	chr6	27473607	27473688	-
16	ORC5	chr7	103766788	103848495	-

# Função PROC.V()

- BUSCAR CHR DO ALPK2
- =PROC.V(ALPK2;  
A2:E16; **2**; FALSO)

1      2      3      4      5

	A	B	C	D	E
1	GENE	CHR	INÍCIO	FIM	FITA
2	TTC9	chr14	71108504	71142077	+
3	LOC100129722	chr9	140144671	140147949	-
4	ATL1	chr14	50999800	51099786	+
5	LOC100506731	chr14	85990776	85998103	-
6	AHI1	chr6	135604670	135818903	-
7	FAM155B	chrX	68725078	68752351	+
8	RNA5SP343	chr11	73909793	73909899	-
9	NKX2-2	chr20	21491655	21494664	-
10	ALPK2	chr18	56148479	56296189	-
11	MTHFD2P7	chr3	179182134	179183119	+
12	FOXN2	chr2	48541783	48606434	+
13	ZNF14	chr19	19821281	19843921	-
14	NCF4	chr22	37257030	37274059	+
15	TRNAS3	chr6	27473607	27473688	-
16	ORC5	chr7	103766788	103848495	-

# Função PROC.V()

- BUSCAR CHR DO ALPK2
- =PROC.V(ALPK2;  
A2:E16; **2**; FALSO)
- =PROC.V(ALPK2;  
A2:E16; **3**; FALSO)
- =PROC.V(ALPK2;  
A2:E16; **4**; FALSO)

**1      2      3      4      5**

	A	B	C	D	E
1	GENE	CHR	INÍCIO	FIM	FITA
2	TTC9	chr14	71108504	71142077	+
3	LOC100129722	chr9	140144671	140147949	-
4	ATL1	chr14	50999800	51099786	+
5	LOC100506731	chr14	85990776	85998103	-
6	AHI1	chr6	135604670	135818903	-
7	FAM155B	chrX	68725078	68752351	+
8	RNA5SP343	chr11	73909793	73909899	-
9	NKX2-2	chr20	21491655	21494664	-
10	ALPK2	chr18	56148479	56296189	-
11	MTHFD2P7	chr3	179182134	179183119	+
12	FOXN2	chr2	48541783	48606434	+
13	ZNF14	chr19	19821281	19843921	-
14	NCF4	chr22	37257030	37274059	+
15	TRNAS3	chr6	27473607	27473688	-
16	ORC5	chr7	103766788	103848495	-

# Função PROC.V()

- BUSCAR CHR DO ALPK2
- =PROC.V(ALPK2;  
A2:E16; **2**; FALSO)
- =PROC.V(ALPK2;  
A2:E16; **3**; FALSO)
- =PROC.V(ALPK2;  
A2:E16; **4**; FALSO)
- =PROC.V(ALPK2;  
A2:B16; **2**; FALSO)

1      2      3      4      5

	A	B	C	D	E
1	GENE	CHR	INÍCIO	FIM	FITA
2	TTC9	chr14	71108504	71142077	+
3	LOC100129722	chr9	140144671	140147949	-
4	ATL1	chr14	50999800	51099786	+
5	LOC100506731	chr14	85990776	85998103	-
6	AHI1	chr6	135604670	135818903	-
7	FAM155B	chrX	68725078	68752351	+
8	RNA5SP343	chr11	73909793	73909899	-
9	NKX2-2	chr20	21491655	21494664	-
10	ALPK2	chr18	56148479	56296189	-
11	MTHFD2P7	chr3	179182134	179183119	+
12	FOXN2	chr2	48541783	48606434	+
13	ZNF14	chr19	19821281	19843921	-
14	NCF4	chr22	37257030	37274059	+
15	TRNAS3	chr6	27473607	27473688	-
16	ORC5	chr7	103766788	103848495	-