



MATRIZES

Prof. André Vignatti

AULAS PASSADAS: VETORES

vetor: organiza muitas variáveis (dados)

1	2	3	4	5	6
5	5	15	25	20	30

muitas variáveis



dificuldade em manipular



manipulações ingênuas e espertas



exercícios complicados

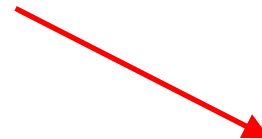
vetor: importante pois representa dados do mundo real

um vetor de números:

1	2	3	4	5	6
5	5	15	25	20	30

uma possível abstração:

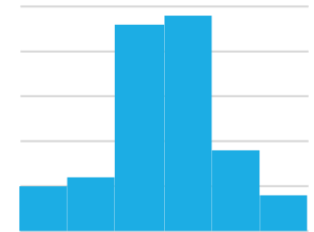
0 à 12	13 à 17	18 à 35	36 à 60	60 à 79	≥ 80
5%	5%	15%	25%	20%	30%



uma única dimensão!
(faixa etária)

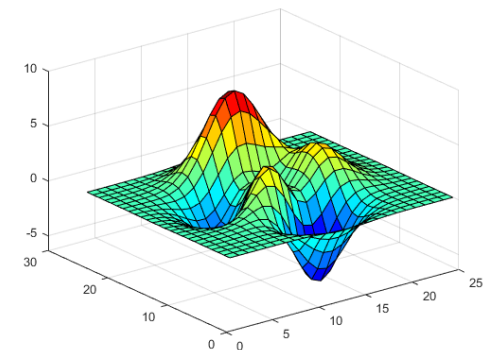
o vetor é limitado! só 1 dimensão!

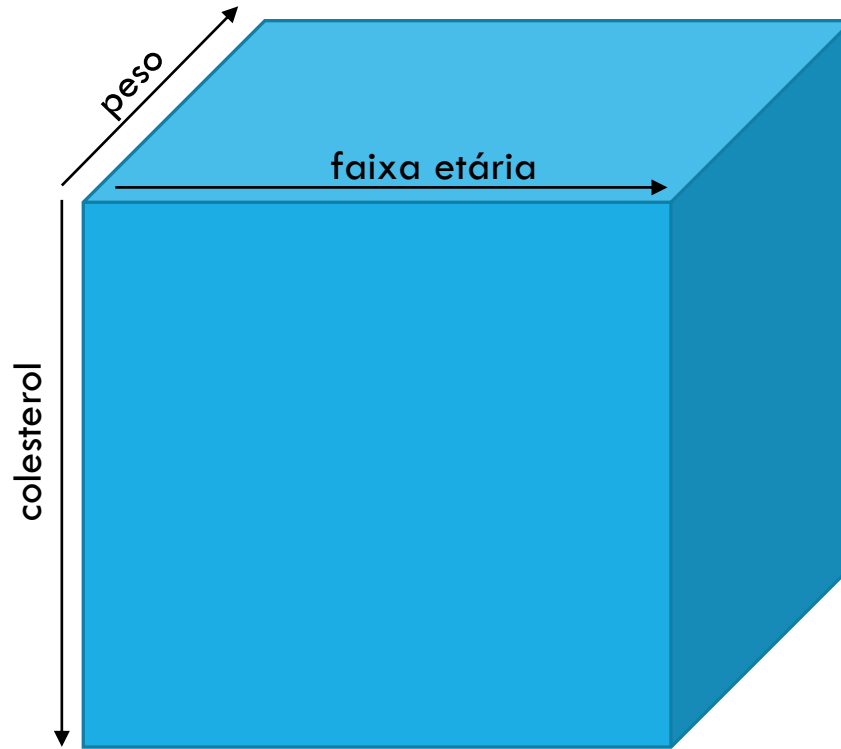
0 à 12	13 à 17	18 à 35	36 à 60	60 à 79	≥ 80
5%	5%	15%	25%	20%	30%



ex: faixa etária X colesterol (2 dimensões)

	0 à 12	13 à 17	18 à 35	36 à 60	60 à 79	≥ 80
< 200	1%	1%	1%	2%	2%	6%
200 à 240	1%	1%	2%	4%	4%	6%
240 à 280	1%	1%	5%	8%	6%	7%
> 280	2%	2%	7%	11%	8%	11%





três dimensões: faixa etária, colesterol, peso

n dimensões: faixa etária, colesterol, peso, altura, tipo sanguíneo, ...



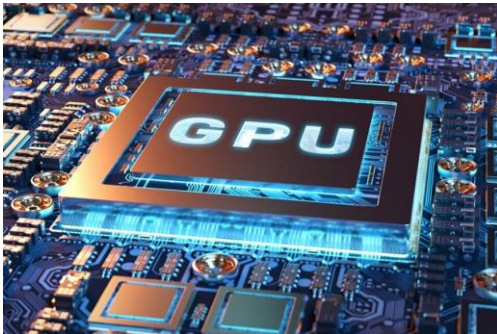
álgebra linear

AULA DE HOJE

aula de hoje: **2D**

matrizes: estruturas organizadas por dois índices (2D)

	0 à 12	13 à 17	18 à 35	36 à 60	60 à 79	≥ 80
< 200	1%	1%	1%	2%	2%	6%
200 à 240	1%	1%	2%	4%	4%	6%
240 à 280	1%	1%	5%	8%	6%	7%
> 280	2%	2%	7%	11%	8%	11%



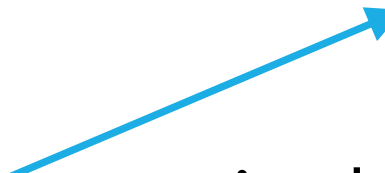
GPU: processador de propósito **específico**, especializado em operações com matrizes e vetores

nosso plano:

- primeiro: matrizes “puras”
- depois: matrizes como imagens

MATRIZ

posições contíguas
de memória



matriz: é um vetor organizado em 2D.

	1	2	3	4
1	4	6	2	1
2	9	0	0	2
3	8	7	3	9
4	1	2	3	4
5	0	1	0	1


opções de declaração:

1) **var m: array [1..5, 1..4] of integer;**

2) **type matriz = array [1..5, 1..4] of integer;**
var m: matriz;

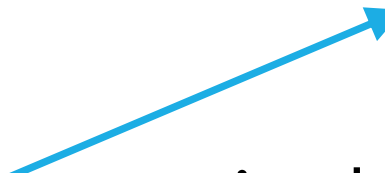
3) **type vetor = array [1.. 4] of integer;**
matriz = array [1.. 5] of vetor ;
var m: matriz;

matriz é um vetor
de vetores!



MATRIZ

posições contíguas
de memória



matriz: é um vetor organizado em 2D.

	1	2	3	4
1	4	6	2	1
2	9	0	0	2
3	8	7	3	9
4	1	2	3	4
5	0	1	0	1

atribuindo valores:

1) `m[1,3] := 12345;`

2) `read (m[1,3]) ;`

CARREGANDO MATRIZES

```
program ler_matriz;  
var w: array [0..50,1..10] of real;  
    i, j: integer;  
  
begin  
    for i:= 0 to 50 do  
        for j:= 1 to 10 do  
            read (w[i,j]);  
        end;  
    end;  
end.
```

CARREGANDO MATRIZES

```
const maxLin = 50; maxCol = 40;  
type matriz= array [0..maxLin, 1..maxCol] of integer;
```

```
procedure ler_matriz (var w: matriz; var n,m: integer);  
var i,j: integer;  
begin  
    read (n); (* n deve estar no intervalo 1..maxLin *)  
    read (m); (* m deve estar no intervalo 1..maxLin *)  
  
    for i:= 1 to n do  
        for j:= 1 to m do  
            read (w[i,j]);  
        end;  
    end;
```

IMPRIMINDO MATRIZES

```
procedure imprimir_matriz (var w: matriz; n,m: integer);  
var i,j: integer;  
begin  
    for i:= 1 to n do  
        begin  
            for j:= 1 to m do  
                write (w[i,j], ' ');  
            writeln;  (* muda de linha a cada fim de coluna *)  
        end;  
    end;  
end;
```

IMPRIMINDO TRANSPOSTA

```
procedure imprimir_transposta (var w: matriz; n,m: integer);  
var i,j: integer;  
begin  
    for i:= 1 to m do  
        begin  
            for j:= 1 to n do  
                write (w[j,i], ' ');  
            writeln;  
        end;  
    end;  
end;
```

Observação: é só a impressão que muda! (e não os dados na memória)

IMPRIMINDO LINHAS E COLUNAS

```
procedure imprimir_uma_linha (var w: matriz; n,m: integer; K: integer);  
  (* imprime a linha K da matriz *)  
  var j: integer;  
  begin  
    for j:= 1 to m do  
      write (w[K,j], ' '); (* K fixo na primeira posicao *)  
    writeln;  
  end;
```

Na matriz do 1º slide, se k=2, imprime “9 0 0 2”

```
procedure imprimir_uma_coluna(var w: matriz; n,m: integer; K: integer);  
  (* imprime a coluna K da matriz *)  
  var i: integer;  
  begin  
    for i:= 1 to n do  
      writeln (w[i,K]); (* K fixo na segunda posicao *)  
    end;
```

Na matriz do 1º slide, se k=2, imprime “6 0 7 2 1”

IMPRIMINDO OS ELEMENTOS PARES

```
procedure imprimir_os_pares(var w: matriz; n,m: integer);  
var i,j : integer;  
begin  
    for i:= 1 to n do  
        begin  
            for j:= 1 to m do  
                if eh_par (w[i,j]) then  
                    write (w[i,j], ' ');  
            end;  
            writeln;  
        end;  
end;
```

IMPRIMINDO LINHA E COLUNAS PARES

```
procedure imprimir_as_linhas_e_colunas_pares(var w: matriz; n,m: integer);
var i,j : integer;
begin
    for i:= 1 to n do
        begin
            for j:= 1 to m do
                if eh_par (i) and eh_par(j) then
                    write (w[i,j], ' ');
            writeln;
        end;
    end;
end;
```