

Aula 34: Tabelas de dados: linhas de matriz e vetores paralelos

Lógica de Programação – Superior · Bloco 3

José Dihego · IFBA

O que você precisa saber hoje

Objetivo: Guardar registros (vários dados de um mesmo item) em linhas de matriz ou em vetores paralelos.

- Registro: cada item é uma **linha** e cada dado, uma **coluna**: `tab[i][0]`, `tab[i][1]`
- Vetores paralelos: `pressao[i]`, `glicemia[i]` e `risco[i]` falam do mesmo paciente `i`
- Varrer a tabela: um laço nas linhas, comparando com o item novo
- Lista sem repetição: procure antes de inserir em `v[total++]`
- `abs` (valor absoluto) vem de `<stdlib.h>`

```
int tab[MAX][3];                // pressão, glicemia, risco
for (int i = 0; i < m; i++)
    scanf("%d %d %d", &tab[i][0], &tab[i][1], &tab[i][2]);
int votos = 0;
for (int i = 0; i < m; i++) {
    int d = abs(p - tab[i][0]) + abs(g - tab[i][1]);
    if (d < limiar) votos += tab[i][2];
}
```

Erros que custam pontos

Errado	Certo	Por quê
<code>tab[0][i]</code>	<code>tab[i][0]</code>	linha = paciente, coluna = dado
<code>d = (p - tab[i][0]) + (g - tab[i][1]);</code>	<code>abs(...) + abs(...)</code>	diferenças negativas cancelam as positivas
<code>if (d <= limiar)</code>	<code>if (d < limiar)</code>	leia se o limite é estrito

```
int tab[4][3] = {{120, 80, -1}, {140, 86, 1},  
                {125, 110, 1}, {115, 75, -1}}, p = 118, g =  
                85, limiar = 25, votos = 0;  
for (int i = 0; i < 4; i++) {  
    int d = abs(p - tab[i][0]) + abs(g - tab[i]  
        ][1]);  
    if (d < limiar) votos += tab[i][2];  
}
```

pressão	glicemia	risco	dist.	votos
120	80	-1	7	-1
140	86	+1	23	0
125	110	+1	32	0
115	75	-1	13	-1

Avaliação Final – 2026.1, Questão II

avaliacao_4_ano_2026_1.pdf

Classificador por vizinhança (distância de Manhattan)

Estratégia:

1. Tabela $M \times 3$: pressão, glicemia e risco (+1 alto, -1 baixo)
2. Distância de Manhattan do novo paciente a cada linha
3. Vizinhos (distância menor que o limiar) votam; o sinal da soma decide

Enunciado completo (com figuras e exemplos) no PDF da prova.

Solução (1/2)

```
1  #include <stdio.h>           // printf e scanf
2  #include <stdlib.h>          // abs
3  #define MAX 100              // a prova usa M = 10
4
5  int main(void) {             // início do programa
6      int m, tab[MAX][3];      // pacientes: pressão, glicemia, risco (+1 ou -1)
7      scanf("%d", &m);         // lê M (10 na prova)
8      for (int i = 0; i < m; i++) // lê a tabela M x 3
9          scanf("%d %d %d", &tab[i][0], &tab[i][1], &tab[i][2]); // pressão, glicemia e risco do
           // paciente i
10     int limiar, pressao, glicemia; // parâmetro e dados do novo paciente
```

Solução (2/2)

```
11     scanf("%d %d %d", &limiar, &pressao, &glicemia); // lê os três valores
12     int votos = 0; // soma dos riscos dos vizinhos (+1 alto, -1 baixo)
13     )
14     for (int i = 0; i < m; i++) { // cada paciente anterior
15         int d = abs(pressao - tab[i][0]) + abs(glicemia - tab[i][1]); // distância de
16         Manhattan
17         if (d < limiar) votos += tab[i][2]; // vizinho (estritamente menor): vota
18     }
19     if (votos > 0) printf("ALTO RISCO\n"); // mais vizinhos de alto risco
20     else if (votos < 0) printf("BAIXO RISCO\n"); // mais vizinhos de baixo risco
21     else printf("INCONCLUSIVO\n"); // empate (inclusive 0 a 0)
22     return 0; // fim sem erro
23 }
```

Avaliação 2 – 2014.2, Questão IV

avaliacao_2_ano_2014_2.pdf

Lista de presença com números repetidos (na prova, 50 números)

Dicas:

- Vetor presentes[] e contador total
- Para cada número da lista, procure em presentes; se não achar, guarde em presentes[total++]
- Imprima o total e os números na ordem em que apareceram

Para conferir: entrada 10 10 20 50 15 50 50 20 10 7 50 → a saída deve conter 5 Alunos

A solução comentada está na página do curso: <https://josedihego.net/logica/#aula-34>