፥



IFSC – EMI/TDS – PROGRAMAÇÃO IV

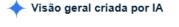
Em programação com Arduino(c) se faz necessário armazenar dados para posterior manipulação dos mesmos. Até agora foi visto uso de Array e Matriz.

Cada um dos exemplos acima listados pode-se armazenar apenas um tipo de dado: inteiro, float ou

texto. A partir de agora estudaremos um conceito mais abrangente onde

é possível unir números com letras. Este recurso é o

Struct.



Ouvir

Em Arduino, uma struct (estrutura) é um tipo de dado definido pelo utilizador que agrupa várias variáveis de diferentes tipos sob um mesmo nome, permitindo organizar e manipular dados relacionados como uma única entidade. Para usar uma struct em Arduino, defina-a usando a palavrachave struct seguida do nome da estrutura e, dentro de chaves, declare os membros que a compõem, e depois crie variáveis desse tipo para aceder e modificar os dados.

Declaração de **struct**:

Nas linhas 1 a 4 serão definidos as colunas e tipo de dados para armazenamento; Na linha 5 é definido o tamanho desta struct[10] posições(gavetas);

```
1 struct abc type {
    int a;
3
    float b;
4
   String c;
5 | abc array[10];
```

```
A seguir a repetição for vai
popular a matriz com os
seguintes dados:
Coluna a: valor de i;
Coluna b: i*(random)
Coluna c: texto+valor da
coluna b;
```

```
9 for (int i = 0; i < 10; i++)
10 {
11
    abc_array[i].a =i;
12
    abc array[i].b =i*random(1,5);
13
    abc array[i].c = "b*10%: "+String((abc_array[i].b)*0.10);
14 }
```

Para exibir os dados gerados usa-se a repetição for:

```
18 for (int i = 0; i < 10; i++)
19 {
   Serial.print("a: "+String(abc array[i].a));
21 Serial.print(" - b: "+String(abc array[i].b));
    Serial.print(" - c: ")+Serial.println(abc array[i].c);
23
   delay(1000);
24 }
```

```
COM7
a: 0 - b: 0.00 - c: b*10%: 0.00
a: 1 - b: 2.00 - c: b*10%: 0.20
a: 2 - b: 4.00 - c: b*10%: 0.40
a: 3 - b: 9.00 - c: b*10%: 0.90
a: 4 - b: 12.00 - c: b*10%: 1.20
a: 5 - b: 5.00 - c: b*10%: 0.50
a: 6 - b: 6.00 - c: b*10%: 0.60
a: 7 - b: 21.00 - c: b*10%: 2.10
a: 8 - b: 32.00 - c: b*10%: 3.20
a: 9 - b: 18.00 - c: b*10%: 1.80
a: 0 - b: 0.00 - c: b*10%: 0.00
a: 1 - b: 2.00 - c: b*10%: 0.20
a: 2 - b: 4.00 - c: b*10%: 0.40
```