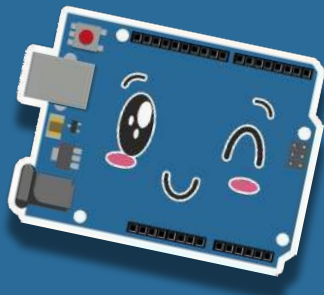


# [T2] Microcontrolador - Desenvolvimento de SW embarcado -

Portar uma aplicação para outra plataforma com mais restrições

Equipe: João Gabriel & Alyson Noronha





```
void hungarian() {
    memset(&pairV, 0, sizeof pairV);

    for(int i = 1, j0 = 0; i <= n; i++) {
        pairV[0] = i;
        memset(&minv, INF, sizeof minv);
        memset(&used, false, sizeof used);

        do {
            used[j0] = true;
            int i0 = pairV[j0], delta = INF, j1;
            for(int j = 1; j <= m; j++) {
                int cur = cost[i0][j] - pu[i0] - pv[j];
                if (used[j]) continue;
                if (cur < minv[j])
                    minv[j] = cur, way[j] = j0;
                if (minv[j] < delta)
                    delta = minv[j], j1 = j;
            }
            for(int j = 0; j <= m; j++) {
                if (used[j]) {
                    pu[pairV[j]] += delta, pv[j] -= delta;
                } else {
                    minv[j] -= delta;
                }
            }
            j0 = j1;
        } while(pairV[j0] != 0);
        do {
            int j1 = way[j0];
            pairV[j0] = pairV[j1];
            j0 = j1;
        } while(j0);
    }
}
```

# ALGORITMO - Descrição

## \* em linguagem C



- Comentários mais detalhados no código enviado na atividade!



# ALGORITMO - Descrição - Estruturas de dados \* em linguagem C

```
// Variáveis utilizadas pelo método húngaro
int pu[MAXN], pv[MAXN];
int way[MAXM], minv[MAXM];
bool used[MAXM];

// Array resultante das arestas usadas no emparelhamento
int pairV[MAXN];
```

```
#define n 44
#define m 44

// Estrutura para do Grafo Bipartido
int cost[MAXN][MAXM]
```



- Comentários mais detalhados no código enviado na atividade!

# >> Nova Plataforma - Arduino UNO - Desafios e Dificuldades

- Familiarização com a nova plataforma.
- Testar/Validar modificações.
- Incompatibilidades e Restrições.

```
1 /* Courier Prime (Alan Dague-Greene)
2  *
3  * http://tinyurl.com/q62n8sn
4  */
5
6 #include <stdio.h>
7
8 int main(void)
9 {
10     for (int i = 99; i >= 1; i--) {
11         if (i == 1) {
12             printf("1 bottle of beer on the wall,\n");
13             printf("1 bottle of beer.\n");
14             printf("You take one down, pass it around,\n");
15             printf("No more bottles of beer on the wall.\n");
16         } else {
17             printf("%d bottles of beer on the wall,\n", i);
18             printf("%d bottles of beer.\n", i);
19             printf("You take one down, pass it around,\n");
20             printf("%d bottles of beer on the wall.\n", i);
21             printf("\n");
22         }
23     }
24 }
```



```
sketch_jun18a
void setup() {
    Serial.begin(9600);
}

void loop() {
    Serial.println("hello littleWorld");
}
```

Uploading ...

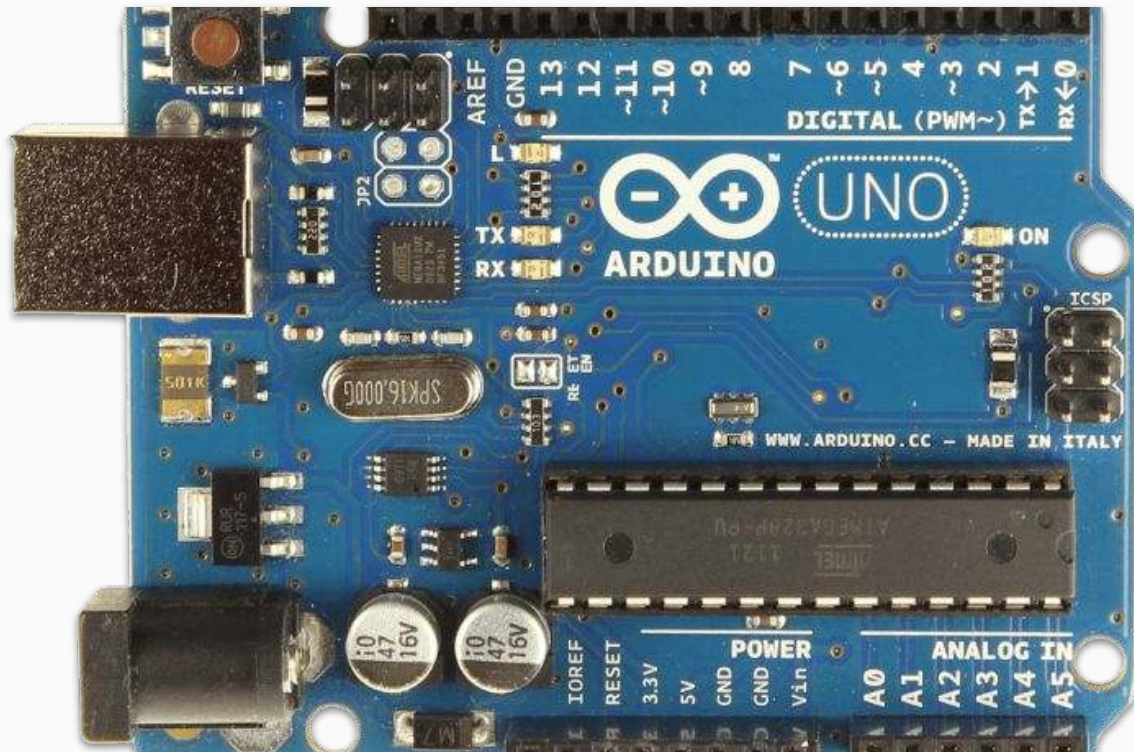
Global variables use 169 bytes memory, leaving 2,391 bytes for the sketch. Maximum is 2,560 bytes.



# < Familiarização / Arduíno UNO >

## O que é -

- Placa microcontroladora baseada no microcontrolador ATmega328;
- Open-Source (código aberto);
- Desenvolvida pela Arduino.cc:
  - - Interação com outros aparelhos;
  - - Integração com outros dispositivos da própria companhia ou não;

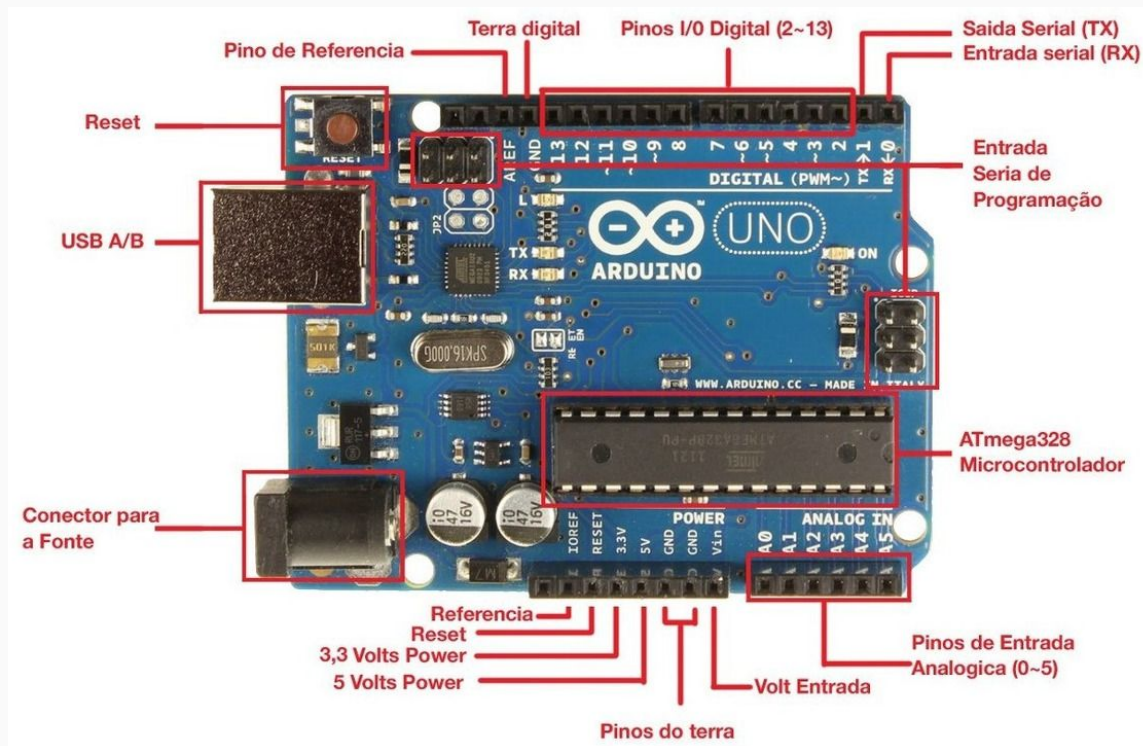




# < Familiarização / Arduíno UNO >

## Características -

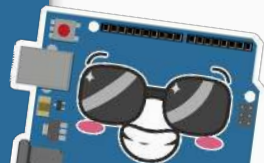
- Esquemas e Referências;
- Alimentação da Placa;
- Memória;
- Entradas e Saídas;
- Comunicação;
- Programação;



# < Familiarização / Arduíno IDE >

## Características -

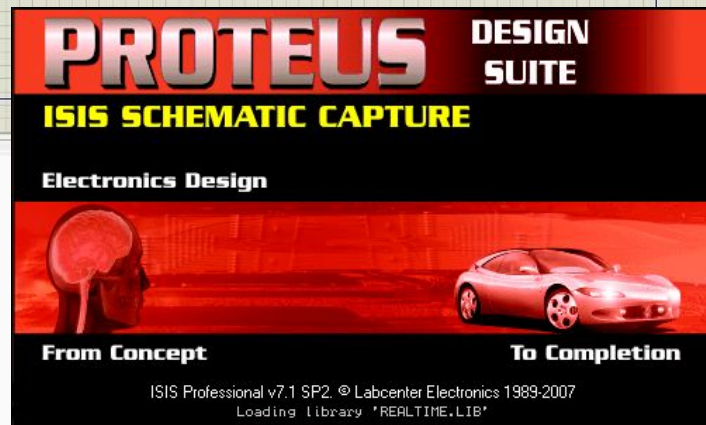
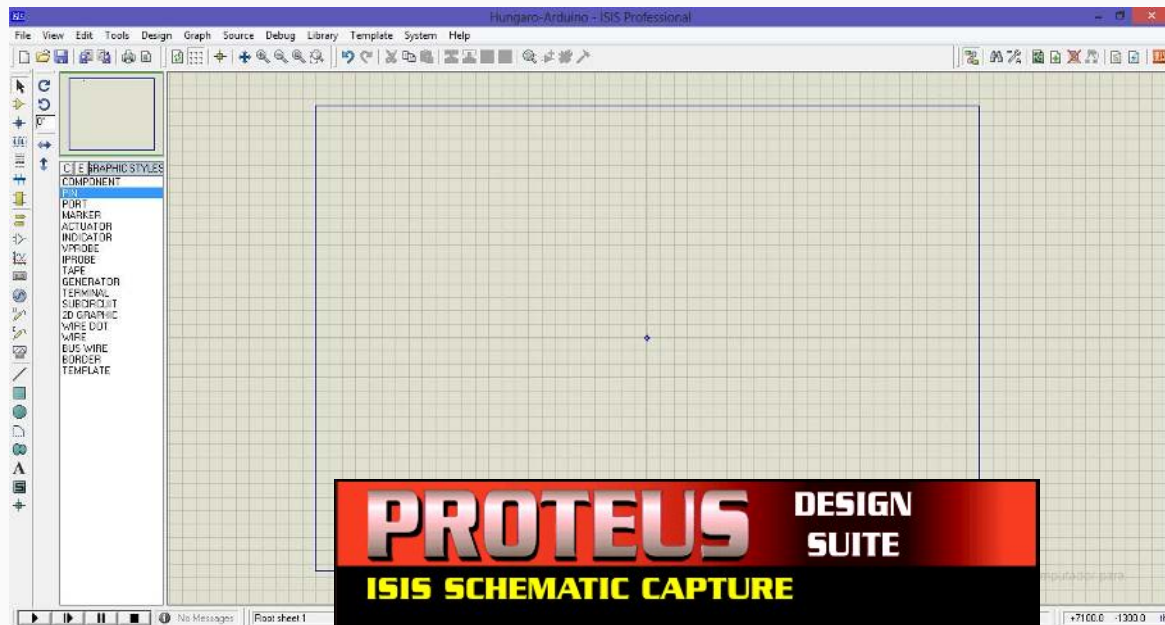
- Interface;
- Ferramentas;
- Linguagem;



## < Testes & Validações / em PC >

### Ambiente de simulação - "Proteus Design Suite" (Proteus ISIS)

- Software de desenho e simulação para criação de projetos eletrônicos;
- Captura esquemática;
- Simulação de microcontroladores;
- Projetos de PCB;
- Verificação 3D;
- E muito mais! ...

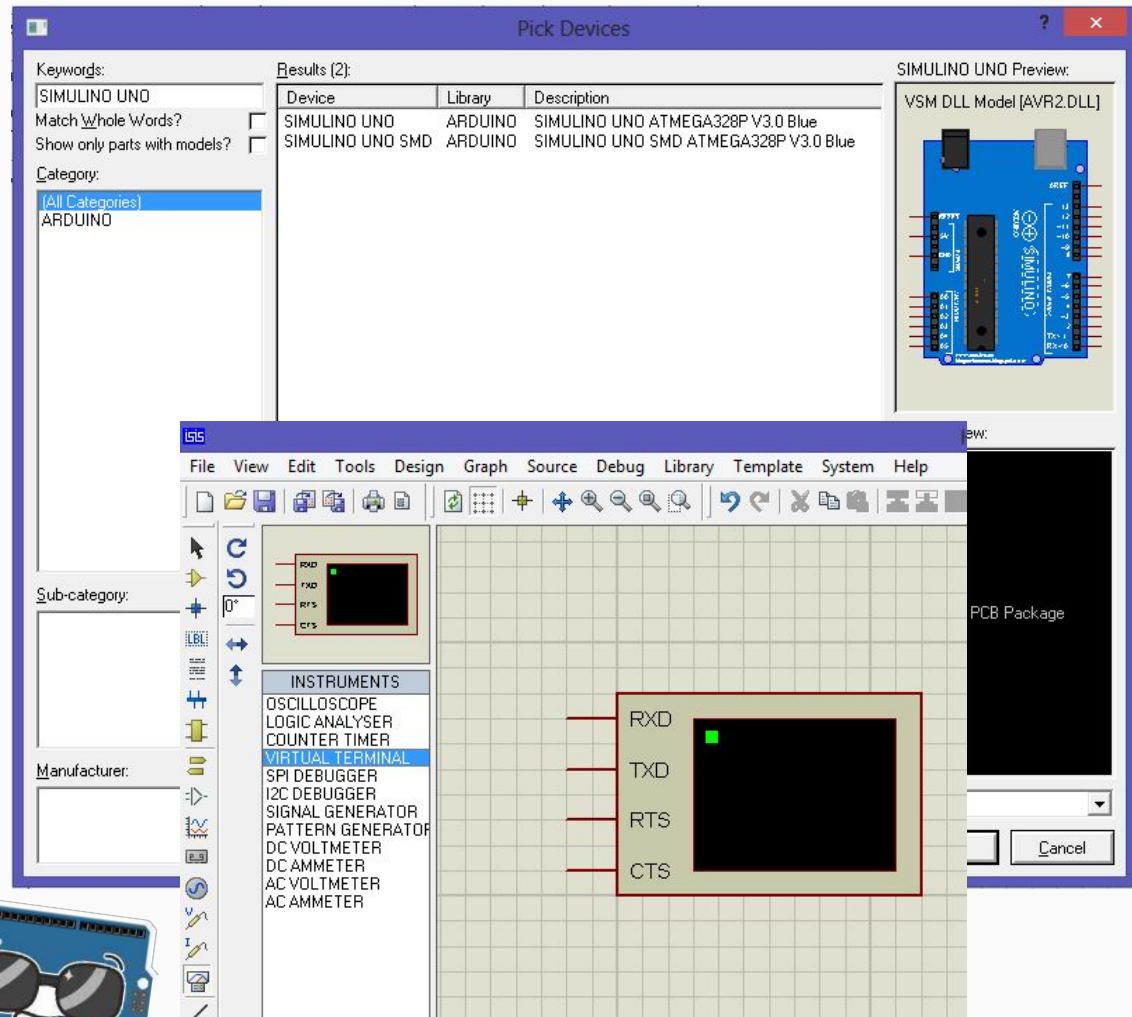




# < Testes & Validações / em PC >

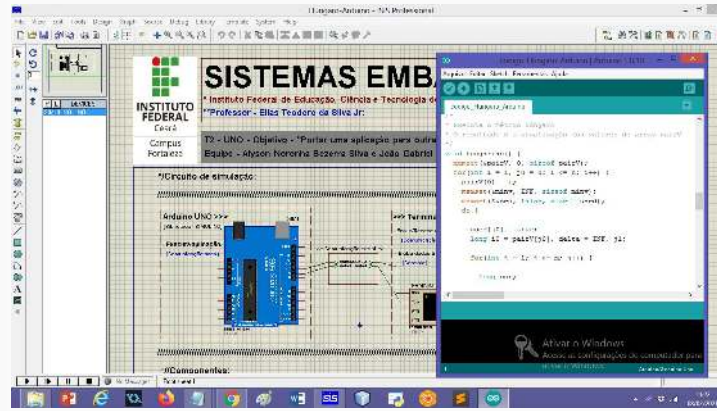
## Ferramentas de simulação -

- Placa Arduino UNO;  
(Proteus - Biblioteca SIMULINO)
- Terminal Virtual;

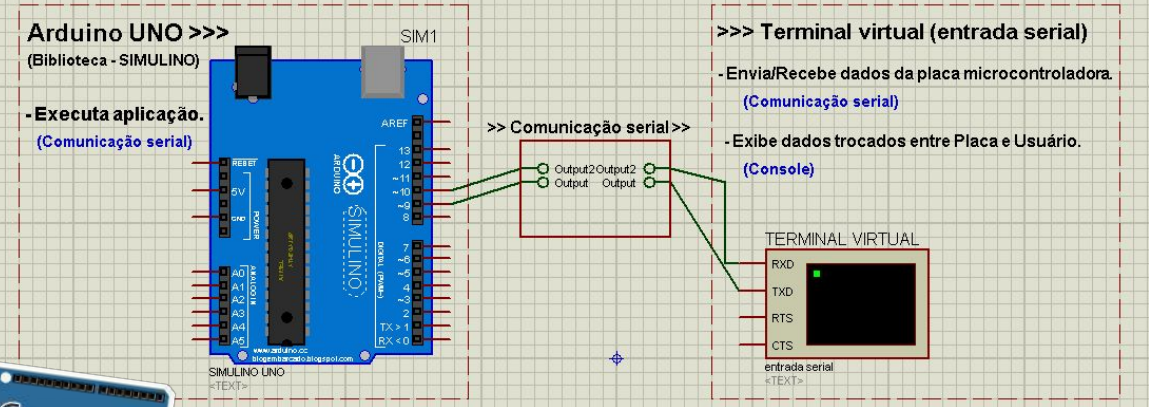


## Modo de simulação -

- ARDUINO IDE + Proteus ISIS;
- Comunicação serial;



**\*Circuito de simulação:**



# < Incompatibilidades & Restrições / >

## Memória - Arduino UNO -

{ - SRAM facilmente excedível; }

- Inviabilidade ao armazenar a entrada teste de  $\sim 2K$  elementos na SRAM da placa;

- Operação "Serial.print();" copia string da FLASH para a SRAM preenchendo o seu espaço;

Flash 32k bytes (of which .5k is used for the bootloader)  
SRAM 2k bytes  
EEPROM 1k byte

```
Custo maximo de: 42613
0 [44][44]
0 [43][44]
0 [44][43]

...Program finished with exit code 0
Press ENTER to exit console.
```

**<< Print dos "Elementos Fora do Escopo" de uma matriz de entrada válida em C**

```
Virtual Terminal - TERMINAL VIRTUAL

Custo maximo da matrix solicitada:
-23455
-8032
0
-8175

[44][44]
[43][44]
[44][43]
```

**<< Print dos "Elementos Fora do Escopo" de uma matriz de entrada válida na linguagem do ARDUINO**

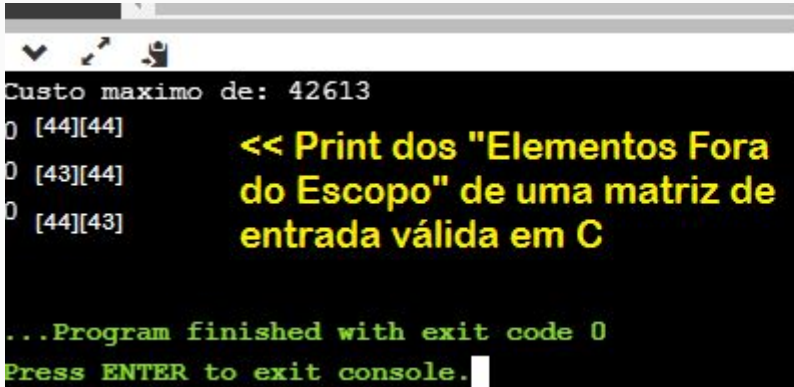


# < Incompatibilidades & Restrições / >

## Memória - Arduino UNO -

- Compiladores da linguagem C reconhecem endereços de memória fora do espaço atribuído a uma variável como sendo ZERO; //
- Compiladores da linguagem do Arduino reconhecem endereços de memória fora do espaço atribuído a uma variável como um inteiro aleatório (lixo); //
- Olhar print dos terminais destacados ao lado para ver essa comparação;

Flash 32k bytes (of which .5k is used for the bootloader)  
SRAM 2k bytes  
EEPROM 1k byte

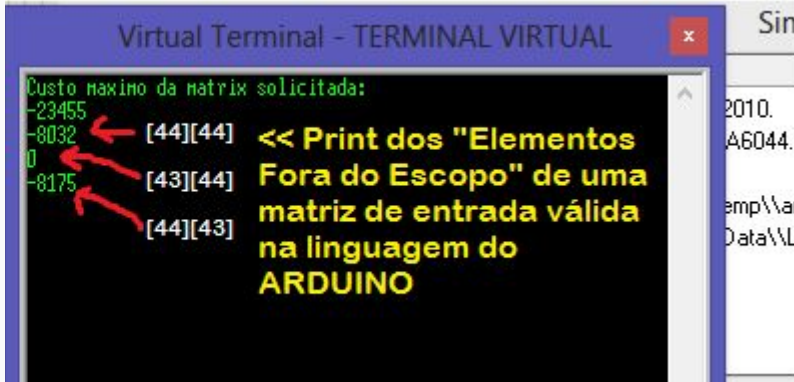


Custo maximo de: 42613

0 [44][44]  
0 [43][44]  
0 [44][43]

**<< Print dos "Elementos Fora do Escopo" de uma matriz de entrada válida em C**

...Program finished with exit code 0  
Press ENTER to exit console.



Virtual Terminal - TERMINAL VIRTUAL

Custo maximo da matrix solicitada:

-23455  
-8032  
0  
-8175

[44][44]  
[43][44]  
[44][43]

**<< Print dos "Elementos Fora do Escopo" de uma matriz de entrada válida na linguagem do ARDUINO**



# < Incompatibilidades & Restrições / >

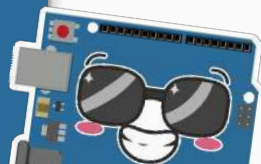
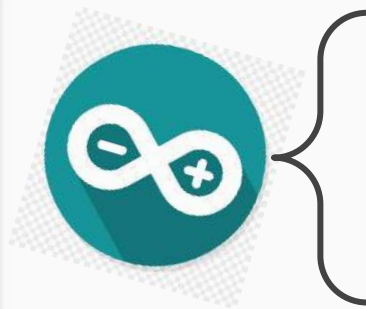
## Bibliotecas -

- p/ Funcionamento do código em C:

- #include <string.h>
- #include <stdio.h>
- #include <stdbool.h>

- p/ Funcionamento do código na plataforma Arduino UNO:

- #include <avr/pgmspace.h>
- #include <SoftwareSerial.h>





# < Incompatibilidades & Restrições / >

## Estrutura dos códigos -

- Na estrutura de uma aplicação C:
  - “int main()”: Executa a aplicação e permite seu controle.
- Na estrutura de uma aplicação no Arduino:
  - “void setup()”: Inicializa e define os valores iniciais uma única vez.
  - “void loop()”: Executa a aplicação e permite o seu controle através de loops sucessivos.



```
sketch_feb08a
void setup() {
  // put your setup code here, to run once:
}

void loop() {
  // put your main code here, to run repeatedly:
}
```



```
#include <stdio.h>

int main() {
  /* my first program in C */
  printf("Hello, World! \n");

  return 0;
}
```



# < Incompatibilidades & Restrições / >

## Estrutura dos códigos - Função - **Void loop ()** -

- Por permitir que o controle da aplicação na plataforma Arduino seja através de loops sucessivos sem o devido tratamento essa função pode acabar repetindo operações, declarações e funções desnecessariamente, gastando a memória para uso da aplicação.



```
sketch_feb08a
void setup() {
  // put your setup code here, to run once:
}

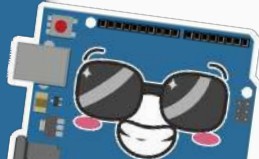
void loop() {
  // put your main code here, to run repeatedly:
}
```



```
#include <stdio.h>

int main() {
  /* my first program in C */
  printf("Hello, World! \n");

  return 0;
}
```



# < Incompatibilidades & Restrições / >

## Tamanhos dos Dados - Linguagem do Arduino

- **Tamanho dos dados** na linguagem do Arduino **muito pequeno** se comparados aos da linguagem C;

- **Realizar muitas operações** com **muitos elementos sucessivamente** resulta em rápidos gastos de memória e de alcançar o limite de valor que esses dados podem guardar (principalmente para uma entrada de  $\sim 2K$  elementos);



| C Basic Data Types | 32-bit CPU   |   | 64-bit CPU   |   |
|--------------------|--------------|---|--------------|---|
|                    | Size (bytes) | Range   | Size (bytes) | Range   |
| char               | 1            | -128 to 127   | 1            | -128 to 127   |
| short              | 2            | -32,768 to 32,767                                   | 2            | -32,768 to 32,767                                   |
| int                | 4            | -2,147,483,648 to 2,147,483,647                     | 4            | -2,147,483,648 to 2,147,483,647                     |
| long               | 4            | -2,147,483,648 to 2,147,483,647                     | 8            | 9,223,372,036,854,775,808-9,223,372,036,854,775,807 |
| long long          | 8            | 9,223,372,036,854,775,808-9,223,372,036,854,775,807 | 8            | 9,223,372,036,854,775,808-9,223,372,036,854,775,807 |
| float              | 4            | 3.4E +/- 38   | 4            | 3.4E +/- 38   |
| double             | 8            | 1.7E +/- 308  | 8            | 1.7E +/- 308  |

| Arduino Data Types           | Size in Bytes | Can contain:  |
|------------------------------|---------------|---|
| boolean                      | 1             | true (1) or false (0)   |
| char                         | 1             | ASCII character or signed value between -128 and 127  |
| unsigned char, byte, uint8_t | 1             | ASCII character or unsigned value between 0 and 255   |
| int, short                   | 2             | signed value between -32,768 and 32,767   |
| unsigned int, word, uint16_t | 2             | unsigned value between 0 and 65,535   |
| long                         | 4             | signed value between -2,147,483,648 and 2,147,483,647   |
| unsigned long, uint32_t      | 4             | unsigned value between 0 and 4,294,967,295  |
| float, double                | 4             | floating point value between -3.4028235E+38 and 3.4028235E+38 (Note that double is the same as a float on this platform.) |

# < Incompatibilidades & Restrições / >

## Sintaxes/Versões das Linguagens -

- Linguagem de programação utilizada pelo Arduino **derivada das linguagens C/C++** (com pequenas modificações);

- Na aplicação em C os dados são “printados” por:  
- “printf( );”

- Na aplicação do Arduino os dados são “printados” por:  
- “Serial.print( );”  
- “Serial.println( );”



# < Incompatibilidades & Restrições / >

## Sintaxes/Versões das Linguagens -

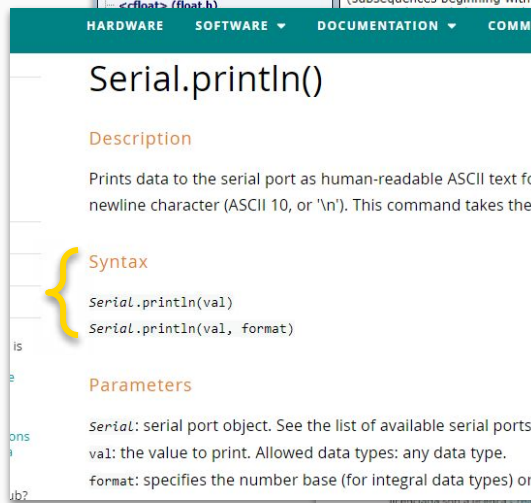
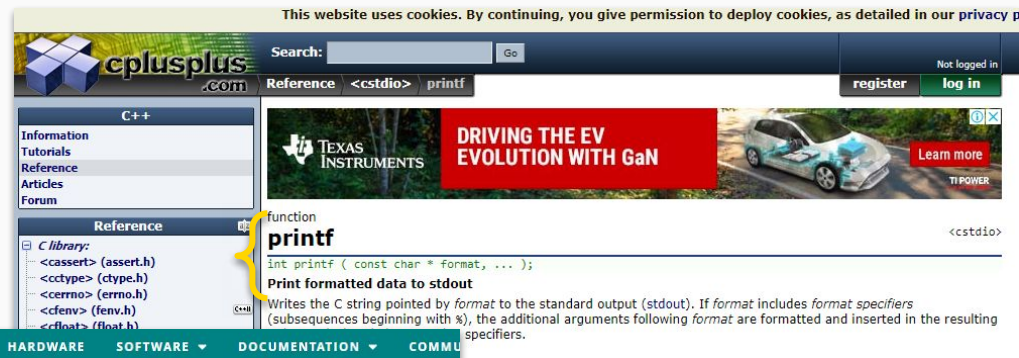
- Linguagem de programação utilizada pelo Arduino **derivada das linguagens C/C++** (com pequenas modificações);

- Na aplicação em C os dados são “printados” por:

- “printf ();”

- Na aplicação do Arduino os dados são “printados” por:

- “Serial.print ();”
- “Serial.println ();”





# < Incompatibilidades & Restrições / >

## Sintaxes/Versões das Linguagens -

- Na aplicação do Arduino antes de se “printar” os dados via comunicação serial é necessário declarar os pinos onde se irá ocorrer essa comunicação:

- “SoftwareSerial mySerial(pino x, pino y);”

- Na aplicação do Arduino precisa-se também definir o “Baud Rate” da comunicação:

- “mySerial.begin(9600);”
- “Serial.begin(9600);”



PROFESSIONAL EDUCATION STORE

Search on Arduino

HARDWARE SOFTWARE DOCUMENTATION COMMUNITY BLOG ABOUT

Esta página também está disponível em outros 2 idiomas. Mudar idioma Português (Brasil)

Reference > Language > Functions > Communication > Serial > Begin

### Serial.begin()

Sintaxe

```
Serial.begin(speed) Serial.begin(speed, config)
```

Parâmetros

**Serial:** objeto porta serial. Veja a lista de portas serials disponíveis em cada placa no [Serial - Página](#)

**speed:** a taxa de transmissão em bits per second (baud rate) - (long)

**config:** configuração do número de bits por transmissão, paridade, e stop bits. Opções válidas são:

- SERIAL\_5N1
- SERIAL\_6N1
- SERIAL\_7N1
- SERIAL\_8N1 (o padrão)
- SERIAL\_5N2

PROFESSIONAL EDUCATION STORE

Hardware Software Documentation Community Blog About

### SoftwareSerial(rxPin, txPin, inverse\_logic)

Description

SoftwareSerial is used to create an instance of a SoftwareSerial object, whose name you need to provide as in the example below. The `inverse_logic` argument is optional and defaults to false. See below for more details about what it does. Multiple SoftwareSerial objects may be created, however only one can be active at a given moment.

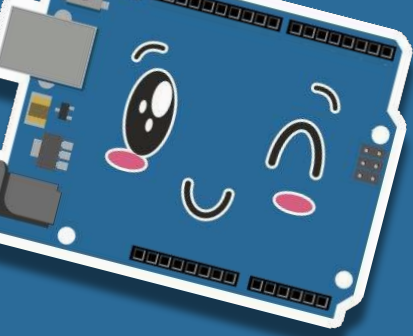
You need to call `SoftwareSerial.begin()` to enable communication.

Parameters

**rxPin:** the pin on which to receive serial data

**txPin:** the pin on which to transmit serial data

**inverse\_logic:** is used to invert the sense of incoming bits (the default is normal logic). If set, SoftwareSerial treats a LOW (0 volts on the pin, normally) on the Rx pin as a 1-bit (the idle state) and a HIGH (5 volts on the pin, normally) as a 0-bit. It also affects the way that it writes to the Tx pin. Default value is false.



# ALGORITMO - Resultados obtidos

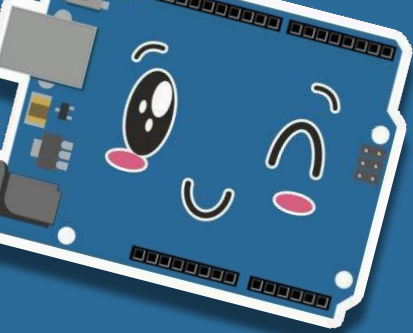
## \* na linguagem do arduino

### \* Modificador de variáveis - PROGMEM

```
// Código da aplicação no Arduino -  
// Matriz de entrada:  
#define linhas 44  
#define colunas 44  
const int Matriz_entrada[linhas][colunas] PROGMEM = {  
  
    { 0,-1,-2}, //  
    {-3,-4,-5}, //  
    {-6,-7,-8}, //  
  
};
```

```
// Código da aplicação em C -  
// Matriz de entrada:  
int linhas, colunas;  
int Matriz_entrada[linhas][colunas];
```





# ALGORITMO - Resultados obtidos

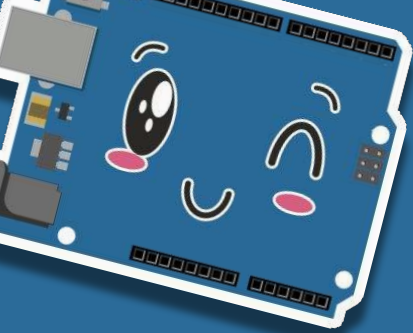
## \* na linguagem do arduino

### \* Modificador de variáveis - PROGMEM - Leitura de dados

```
// Código da aplicação em C -  
// trecho de código que acessa  
int custo_maximo = 0;  
metodo_hungaro();  
for(int j = 1; j <= m; j++) {  
    if (pesos_max[j] == 0) continue;  
    custo_maximo += Matriz_entrada[pesos_max[j]][j];  
}
```




```
// Código da aplicação no Arduino -  
// trecho de código que acessa a matriz de entrada (Matriz_entrada):  
long custo_maximo = 0;  
metodo_hungaro();  
for(int j = 1; j <= m; j++) {  
    if (pesos_max[j] == 0) continue;  
    if ((j < m) || (pesos_max[j] < m)) custo_maximo += (int)pgm_read_word_near(&Matriz_entrada[pesos_max[j]][j]);  
}
```



# ALGORITMO - Resultados obtidos

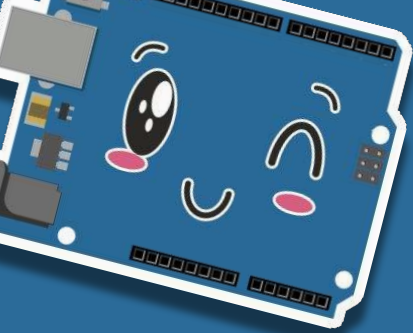
## \* na linguagem do arduino

\* Biblioteca - SoftwareSerial - Possibilitar Comunicação Serial



```
// Código da aplicação em C -  
// print do resultado final calculado pelo método (custo máx)  
printf("Custo máximo de: %d\n", -custo_maximo); /  
printf("\n");
```

```
// Código da aplicação no Arduino -  
// print do resultado final calculadopelo método (custo máx):  
mySerial.println(F("Custo maximo da matrix solicitada: "));  
mySerial.print(custo_maximo);  
mySerial.println(F("\n"));
```



# ALGORITMO - Resultados obtidos

## \* na linguagem do arduino

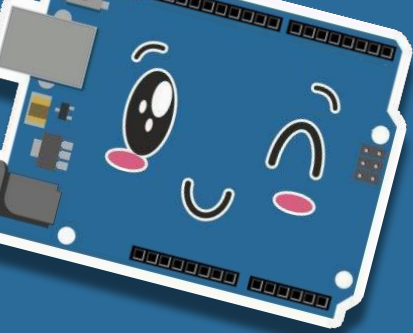
### \* Biblioteca - Restrições de memória -

```
// Código da aplicação no Arduino -  
// trecho de código que acessa a matriz de entrada (Matriz_entrada):  
long custo_maximo = 0;  
metodo_hungaro();  
for(int j = 1; j <= m; j++) {  
    if (pesos_max[j] == 0) continue;  
    if ((j < m) or (pesos_max[j] < m)) custo_maximo += (int)pgm_read_word_near(&Matriz_entrada[pesos_max[j]][j]);  
}
```



Evita que a IDE do ARDUINO acesse  
endereços fora do ESCOPO  
da MATRIZ DE ENTRADA e adicione  
valores aleatórios





# ALGORITMO - Resultados obtidos

## \* na linguagem do arduino

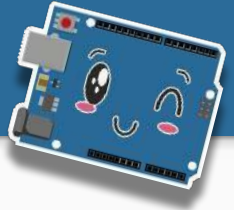
\* Biblioteca - Restrições de memória - Evitar “memory leak”

```
if ((i0 == m) or (j == m)) { // Evitar acesso a índices FORA  
    cur = 0 - pu[i0] - pv[j]; // do limite de memória da MATRIZ DE ENTRADA  
}else{  
    cur = (long)pgm_read_word_near(&cost[i0][j]) - pu[i0] - pv[j];  
}
```



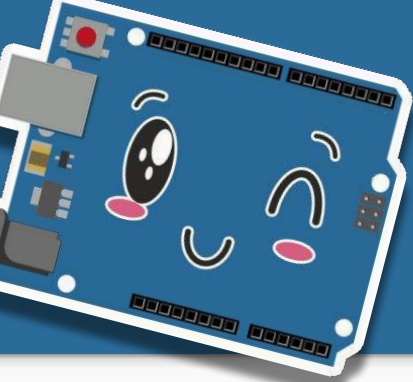
# SEMB - Contextualizando

## Plataformas Microcontroladas



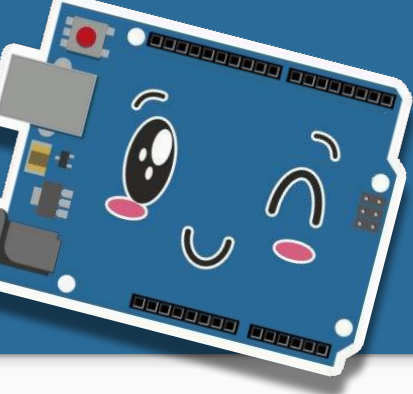
## Plataformas Microcontroladas

- (Recursos limitados): A natureza “limitada” dos microcontroladores torna a programação nestes mais restritiva e limitada quanto ao uso dos recursos que essa pode disponibilizar como memória, energia, e etc.
- (Maior conhecimento e estudo requeridos): Suas muitas limitações acabam por englobar tanto áreas de software quanto de hardware, pedindo do seu desenvolvedor conhecimentos de sistemas digitais, componentes eletrônicos, projeto de hardware, montagem e muito além do que só a programação em si.



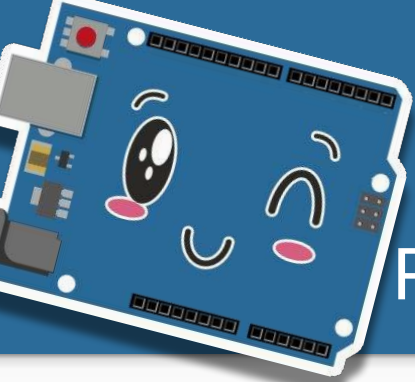
# REFERÊNCIAS - Artigos online

- [https://en.wikipedia.org/wiki/Hungarian\\_algorithm](https://en.wikipedia.org/wiki/Hungarian_algorithm)
- <https://maestrovirtuale.com/metodo-hungaro-o-que-e-exemplo/>
- <https://www.filipeflop.com/blog/o-que-e-arduino/>
- <https://www.embarcados.com.br/arduino-uno/>
- [https://en.wikipedia.org/wiki/Arduino\\_Uno](https://en.wikipedia.org/wiki/Arduino_Uno)
- [https://pt.wikipedia.org/wiki/Proteus\\_\(programa\\_de\\_computador\)](https://pt.wikipedia.org/wiki/Proteus_(programa_de_computador))
- [https://en.wikipedia.org/wiki/Proteus\\_Design\\_Suite](https://en.wikipedia.org/wiki/Proteus_Design_Suite)
- <https://www.embarcados.com.br/clp-versus-microcontrolador/#:~:text=As%20grandes%20vantagens%20na%20utiliza%C3%A7%C3%A3o,aplica%C3%A7%C3%B5es%20caseiras%20ou%20mais%20simples.>
- [https://www.tutorialspoint.com/cprogramming/c\\_program\\_structure.htm](https://www.tutorialspoint.com/cprogramming/c_program_structure.htm)
- [https://www.tutorialspoint.com/arduino/arduino\\_program\\_structure.htm#:~:text=Arduino%20programs%20can%20be%20divided,any%20syntax%20or%20compilation%20error.](https://www.tutorialspoint.com/arduino/arduino_program_structure.htm#:~:text=Arduino%20programs%20can%20be%20divided,any%20syntax%20or%20compilation%20error.)
-



## REFERÊNCIAS - Sites

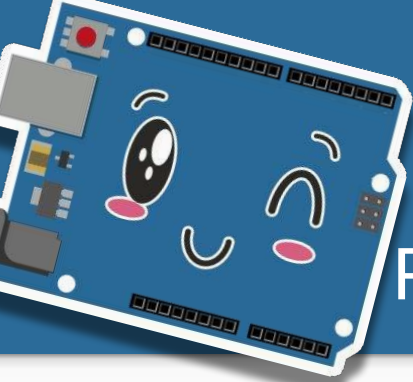
- [site: http://www.hungarianalgorithm.com/index.php](http://www.hungarianalgorithm.com/index.php)
- <https://forum.arduino.cc/index.php?topic=218231.0>
- <https://forum.arduino.cc/index.php?topic=177842.0>
- <https://www.baudaeletronica.com.br/arduino-uno-r3.html#:~:text=O%20Arduino%20Uno%20R3%20C3%A9,e%20um%20bot%C3%A3o%20de%20reset.>
- <https://gist.github.com/juniorcesarabreu/ff81bac6dd8510bcd1a816f270223168>
- <https://www.labcenter.com/simulation/>
- [https://stackoverflow.com/questions/23096366/how-to-stop-a-loop-arduino#:~:text=The%20closest%20you%20can%20do,stop%20processing%20until%20it's%20reset.&text=void%20loop\(\)%20%7B%20%2F%2F%20All,required%20to%20prevent%20compile%20error.%20%7D](https://stackoverflow.com/questions/23096366/how-to-stop-a-loop-arduino#:~:text=The%20closest%20you%20can%20do,stop%20processing%20until%20it's%20reset.&text=void%20loop()%20%7B%20%2F%2F%20All,required%20to%20prevent%20compile%20error.%20%7D)
- [https://www.nongnu.org/avr-libc/user-manual/group\\_avr\\_pgmspace.html](https://www.nongnu.org/avr-libc/user-manual/group_avr_pgmspace.html)



# REFERÊNCIAS - Sites - Linguagem C

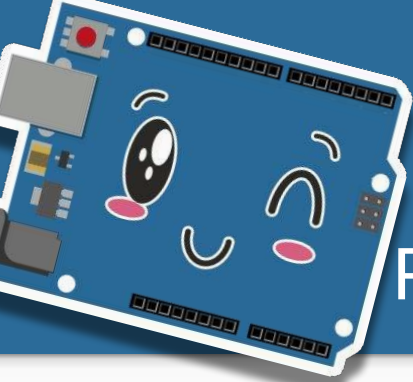
- <https://en.cppreference.com/w/c/language>
- <http://www.cplusplus.com/reference/cstring/memset/>





## REFERÊNCIAS - Sites - Arduino.cc

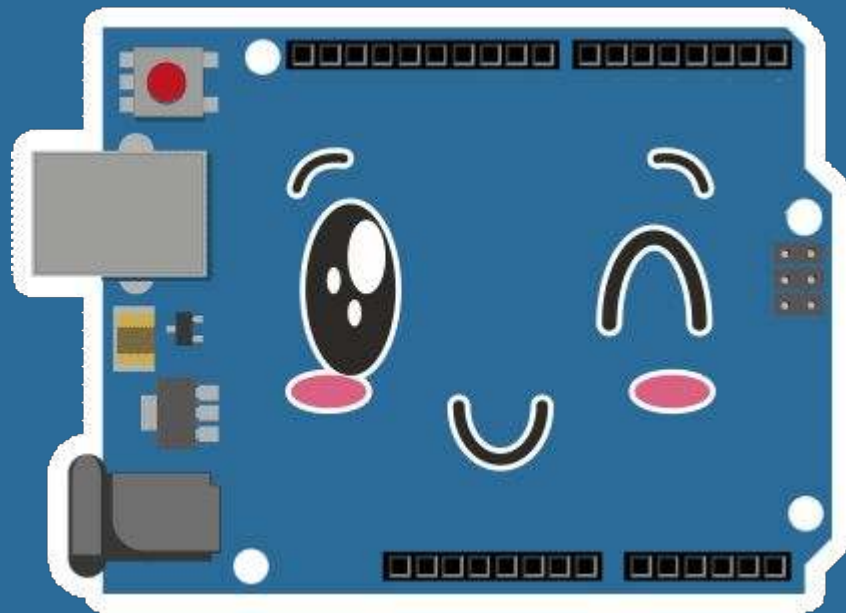
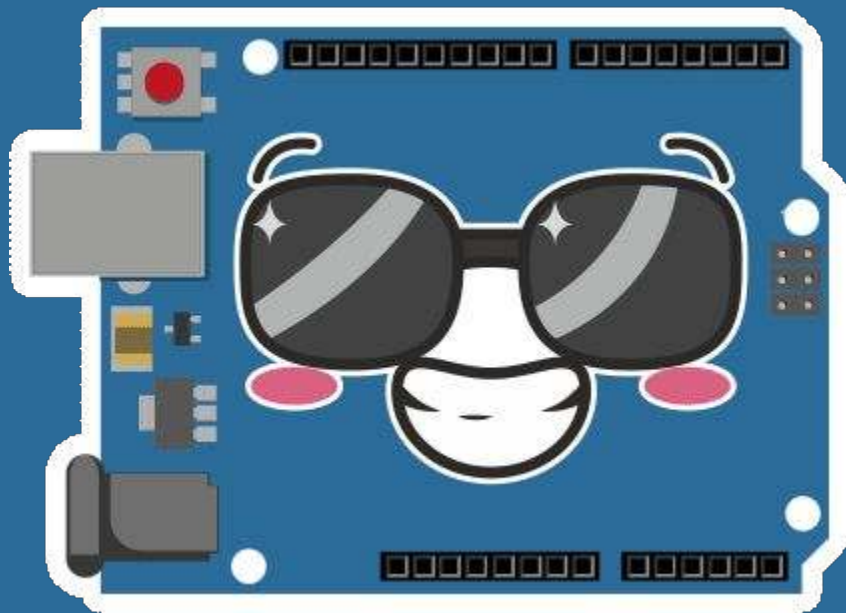
- <https://store.arduino.cc/usa/arduino-uno-rev3>
- <https://www.arduino.cc/en/uploads/Main/arduino-uno-schematic.pdf>
- <https://www.arduino.cc/en/software/>
- <https://www.arduino.cc/reference/pt/>
- <https://www.arduino.cc/en/Reference/HomePage&xid=17259,15700021,15700105,15700124,15700149,15700168,15700173,15700201>
- <https://www.arduino.cc/en/Reference/softwareSerial>
- <https://www.arduino.cc/reference/pt/language/variables/utilities/progmem/>
- <https://www.arduino.cc/reference/pt/language/functions/communication/serial/begin/>
- <https://www.arduino.cc/reference/en/language/functions/communication/serial/println/>
- <https://www.arduino.cc/reference/pt/language/functions/communication/serial/print/>



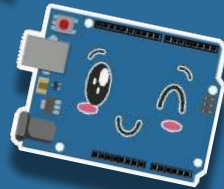
## REFERÊNCIAS - Sites - Arduino.cc

- <https://www.arduino.cc/reference/en/language/structure/sketch/setup/>
- <https://www.arduino.cc/reference/en/language/structure/sketch/loop/>
- <https://www.arduino.cc/en/Reference/SoftwareSerialConstructor>

\*Dúvidas?



Obrigado !



{ Fim }