

**Isaac Santos Soares**  
**Matheus Henrique Alves de Souza**  
**Mikael Alves Matias**  
**Wallace Silva Francelino**

## **PLACAR AUTOMÁTICO DE BASQUETE**

Trabalho de Conclusão de Curso apresentado ao  
Curso Técnico em Eletrônica da Etec Gildo Marçal  
Bezerra Brandão, orientado pelo Prof.º Admilson  
Lima Nascimento, como requisito parcial para  
obtenção do título de técnico em Eletrônica.

**São Paulo**  
**2018**

**Isaac Santos Soares**  
**Matheus Henrique Alves de Souza**  
**Mikael Alves Matias**  
**Wallace Silva Francelino**

## **PLACAR AUTOMÁTICO DE BASQUETE**

Trabalho de Conclusão de Curso apresentado ao  
Curso Técnico em Eletrônica da Etec Gildo Marçal  
Bezerra Brandão, orientado pelo Prof.º Admilson  
Lima Nascimento, como requisito parcial para  
obtenção do título de técnico em Eletrônica.

BANCA EXAMINADORA

\_\_\_\_\_/\_\_\_\_/\_\_\_\_

ETEC Gildo Marçal Bezerra Brandão

\_\_\_\_\_/\_\_\_\_/\_\_\_\_

ETEC Gildo Marçal Bezerra Brandão

\_\_\_\_\_/\_\_\_\_/\_\_\_\_

ETEC Gildo Marçal Bezerra Brandão

**Resumo**

O principal objetivo deste trabalho é automatizar o sistema de contagem dos pontos de basquete, mediante conceitos e métodos adquiridos no decorrer do curso de Eletrônica. Através da construção de um protótipo do projeto visamos favorecer na redução de tempo de marcação dos pontos e possibilitar a prevenção de erros graves na contagem dos pontos, pois leva em consideração o cálculo da distância em que a bola foi lançada. Além de facilitar o trabalho dos juízes e árbitros, aproxima cada vez mais a tecnologia ao mundo dos esportes; o que fortalece quatro pilares indispensáveis para o mundo esportivos - controle, tecnologia, treinamento e transparência.

## **Abstract**

The main objective of this work is to automate the counting system of basketball points, using concepts and methods acquired during the course of Electronics. Through the construction of a prototype of the project we aim to favor the reduction of time marking points and enable the prevention of serious errors in the score of points, as it takes into consideration the calculation of the distance in which the ball was launched. In addition to facilitating the work of judges and referees, it increasingly brings technology closer to the world of sports; which strengthens four indispensable pillars for the sports world - control, technology, training and transparency.

## **Lista de ilustrações**

Figura 1 - Contagem de pontos (atualmente)	9
Figura 2 - Placa da barreira óptica Placa da barreira óptica	11
Figura 3 - Sistema mecânico Sistema mecânico	11
Figura 4 - CI 74595 CI 74595	12
Figura 5 - Diagrama do CI 74595 Diagrama do CI 74595	12
Figura 6 - Registradores e displays montados no protoboard	13
Figura 7 - Diagrama da Esp-8266	14
Figura 8 - Diagrama da Esp-01	15
Figura 9 - Diagrama da Esp-12	16
Figura 10 - Diagrama placa de prototipagem NodeMCU	16
Figura 11 - Área de trabalho do VS Code	17
Figura 12 - Código comentado parte 1	18
Figura 13 - Código comentado parte 2	19
Figura 14 - Código comentado parte 3	20
Figura 15 - Código comentado parte 4	21
Figura 16 - Código comentado parte 5	22
Figura 17 - Código comentado parte 6	22
Figura 18 - Código comentado parte 7	23
Figura 19 Transmissor fora da caixa	23
Figura 20 - Transmissores dentro da caixa	24
Figura 21 - NodeMCU com 74595	24
Figura 22 - Tela de trabalho do software de circuito elétrico ISIS Proteus 8.5	25
Figura 23 - Tela de trabalho do software de PCB ARES Proteus 8.5	25
Figura 24 - Circuito Elétrico dos displays feito no proteus	26
Figura 25 - Placa PCB no Proteus	26
Figura 26 - Contador com PIC18f4550 e 4511	27
Figura 27 - Contador com o PIC18f4550 com o sensor da bola e os displays grande.	28
Figura 28 - Contador com o Arduino Mega 2560 e registrador de deslocamento	29
Figura 29 - Contador com variação de pontos NodeMCU	30
Figura 30 - Contador com a variação dos pontos conforme a distância	30
Figura 31 - Solução MikroC Solução MikroC	31
Figura 32 - Comparação de GPIOs entre PIC18f4550 e NodeMCU	32

1 INTRODUÇÃO	7
2 OBJETIVO GERAL	7
2.1 Objetivo específico	7
3 JUSTIFICATIVA	7
4 REFERENCIAL TEÓRICO	8
5 PROCEDIMENTOS METODOLÓGICOS	8
6 DESENVOLVIMENTO	8
6.1 Processo de contagem dos pontos	8
6.2 Elaboração do projeto	9
6.3 Automação do processo	9
6.3.1 Lógica de funcionamento	10
6.3.2 Sensor	10
6.3.3 Registrador de deslocamento	11
6.3.4 Microcontroladores	13
6.3.5 Programação C++	17
6.4 Circuito elétrico	23
6.4.1 Placar	23
6.4.2 Transmissor do jogador	23
6.4.3 Receptor	24
6.4.4 Proteus	24
6.5 Montagem	26
6.5.1 Confeção	27
6.6 Testes e simulações	27
6.7 Problemas e soluções	31
7 CONCLUSÃO	33
8 REFERÊNCIAS	33
9 APÊNDICES	33
9.1 Diagrama em Blocos	33
9.2 Tabela Verdade Display 7 Seguimentos	34
9.3 Tabela de preços sem doação	34
9.4 Tabela de preços sem doação	35



## **1 INTRODUÇÃO**

Atualmente a contagem de pontos do basquete é feita totalmente manual, apenas a parte de exibição no placar para a torcida é feita com o uso de displays e botões para somar ou subtrair 2 pontos, 3 pontos ou 1 ponto no caso de ter ocorrido alguma falta. Por esse motivo pode ocorrer erros de contagem e acabar acontecendo como ocorreu em 2003 no torneio feminino para os jogos Pan Americanos.

No dia 09 de agosto de 2003 a Folha de São Paulo publicou uma matéria sobre um erro dos mesários e da arbitragem na contagem dos pontos da partida de basquete feminino ocorrida no dia 08 de agosto de 2003. Tratava-se do jogo do Brasil contra EUA, que terminou 75 a 69 para os EUA devido o acréscimo de pontos fantasmas à equipe campeã. Este erro custou a chance da equipe de basquete brasileira feminina garantir uma medalha no Pan-Americano de Santo Domingo.

Com a utilização deste sistema de contagem dos pontos proposto visamos favorecer na redução de tempo de marcação dos pontos e possibilitar a prevenção de erros graves na contagem; o protótipo leva em consideração o cálculo da distância em que a bola foi lançada para a soma dos pontos. Tal tecnologia além de facilitar o trabalho dos juízes e árbitros, aproxima cada vez mais a tecnologia ao mundo dos esportes; assim fortalece quatro pilares indispensáveis para o mundo esportivos - controle, tecnologia, treinamento e transparência.

## **2 OBJETIVO GERAL**

Desenvolver um contador de pontos automático para uma partida de basquete com o uso dos conceitos, técnicas e métodos adquiridos ao longo do curso de eletrônica para construção de um protótipo do projeto.

### **2.1 Objetivo específico**

Neste projeto foram utilizados microcontroladores e sensores para criar um contador de pontos automático de basquete, que conforme regras oficiais os pontos variam de acordo com a distância ou por uma situação específica que é a falta. Esse projeto visa reduzir o tempo para a contagem; prevenir erros graves de marcação e aproximar ainda mais a tecnologia para o mundo dos esportes.

## **3 JUSTIFICATIVA**

Este projeto contribui para prevenir ou diminuir a margem de erros da arbitragem, uma vez que o sistema mede a distância em relação à cesta, para a somatória dos pontos. Outrossim, nele se aplica vários conceitos de eletrônica digital e sistemas microprocessador adquiridos no decorrer do curso de eletrônica.



## **4 REFERENCIAL TEÓRICO**

Para este trabalho foram utilizados principalmente conceitos relacionados a eletrônica digital e microcontroladores. Por meio de pesquisas na internet foi observado que ainda não existe ideias similares a tratada neste documento, único projeto que chegou próximo a nossa ideia, foram os mini games de basquete no shopping, na qual conta os pontos, porém nestas não há variação no valor da cesta de acordo com a distância conforme determina a regra oficial.

## **5 PROCEDIMENTOS METODOLÓGICOS**

Para a realização deste trabalho primeiramente foram utilizados métodos de pesquisas, para se ter um melhor embasamento sobre o tema escolhido: facilitar na hora da computação dos pontos de basquete. Sendo assim, percebemos que a melhor forma de facilitar seria automatizando a contagem de acordo com a variação da distância, foi feito um questionário com os integrantes do grupo, de como seria feito essa contagem da distância e se seria viável.

Posteriormente também foram feitas pesquisas sobre assuntos relacionados:

- Como é feita a contagem atualmente;
- Tamanho da quadra;
- Métodos de calcular a distância digitalmente;
- Regras do basquete.

## **6 DESENVOLVIMENTO**

### **6.1 Processo de contagem dos pontos**

Por meio de pesquisas na internet observamos que a contagem de pontos atualmente é feita manual, desde a medição da distância que é feita por imagens, para ver onde o pé do jogador se encontra no momento em que a cesta foi efetuada, até a hora de demonstrar nos displays de 7 segmentos, onde contém um botão para a marcação de 3 pontos, 2 pontos e 1 ponto.

Abaixo está uma imagem ilustrativa de como é feita a contagem:

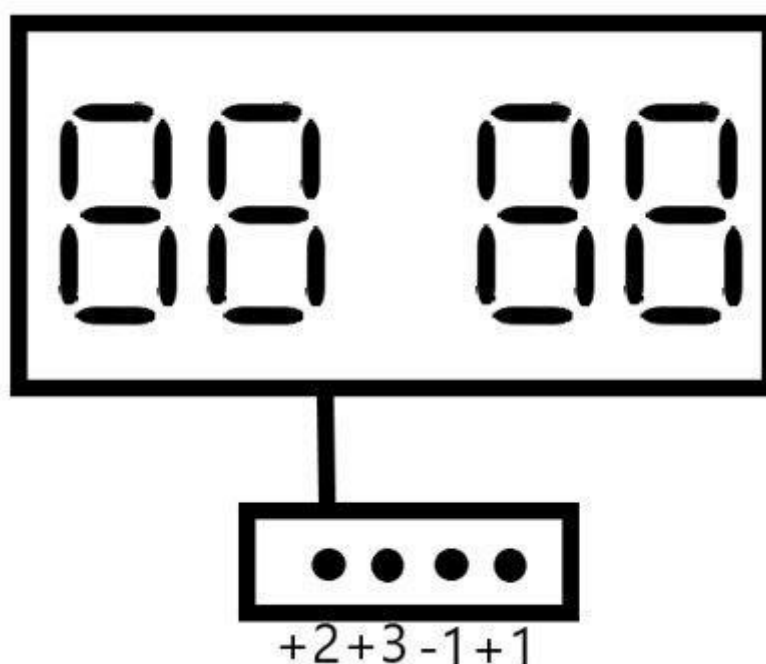


Figura 1 - Contagem de pontos (atualmente)

## 6.2 Elaboração do projeto

A ideia deste trabalho foi constituída no intuito de automatizar o processo de contagem da distância entre o jogador e a cesta e assim determinar o valor do ponto, para efeito de aplicabilidade dos conceitos adquiridos no curso e contribuição da eletrônica para a facilitação e diminuição de erros de arbitragem.

Com a automatização da variação da contagem dos pontos se obteve as seguintes vantagens:

- Diminuição do tempo gasto pelo árbitro: em uma partida de basquete quando um ponto é efetuado os árbitros devem consultar nas imagens capturadas pelas câmeras, se o pé do jogador se encontra dentro ou fora da linha de três pontos no momento do arremesso da bola ao encontro da cesta.
- Redução na margem de erros: na forma de contagem dos pontos atualmente, foram encontrados erros na hora de determinar o valor do ponto de acordo com a distância em que foi arremessada a bola, com o projeto isso foi reduzido.
- Facilitação no trabalho dos árbitros: com a utilização deste projeto reduzirá o trabalho dos árbitros no que tange a necessidade de rever repetidas vezes as imagens dos lances para certificar-se das distâncias de lançamento da bola.

## 6.3 Automação do processo

O objetivo didático deste projeto é montar um protótipo para a contagem de pontos de acordo com a variação da distância em relação ao jogador com a cesta, composto principalmente por microcontroladores com acesso e roteamento WiFi, registrador de deslocamento, barreira óptica.

### 6.3.1 Lógica de funcionamento

O placar terá três microcontroladores, dois como transmissores e o restante como receptor. Para iniciar o processo de verificação da distância o sensor da cesta tem que ser acionado e os transmissores devem estar ativos, caso contrário o placar não computará os pontos feitos pelos jogadores. Além do sensor da cesta ativo e os transmissores funcionando, um computador será necessário para informar qual jogador efetuou a cesta.

O placar terá uma lógica de funcionamento relacionada aos seguintes passos:

- Fazer a cesta;
- Verificar se os transmissores dos jogadores estão ativos;
- Identificar o jogador que efetuou a cesta;
- Verificar se o jogador se encontra dentro ou fora da linha de 3 pontos;
- Somar o respectivo ponto feito ao placar antigo;
- Demonstrar os placares atuais no display.

### 6.3.2 Sensor

Foi usado sensor do tipo barreira óptica neste protótipo.

O sensor de barreira óptica foi utilizado para monitorar se o jogador conseguiu fazer a cesta. Na lógica da programação ele atua como gatilho de funcionamento, pois o microcontrolador espera o momento em que o sensor será ativado para iniciar processo de verificação.

A seguir o sensor utilizado em dele e posteriormente fixado na tabela:

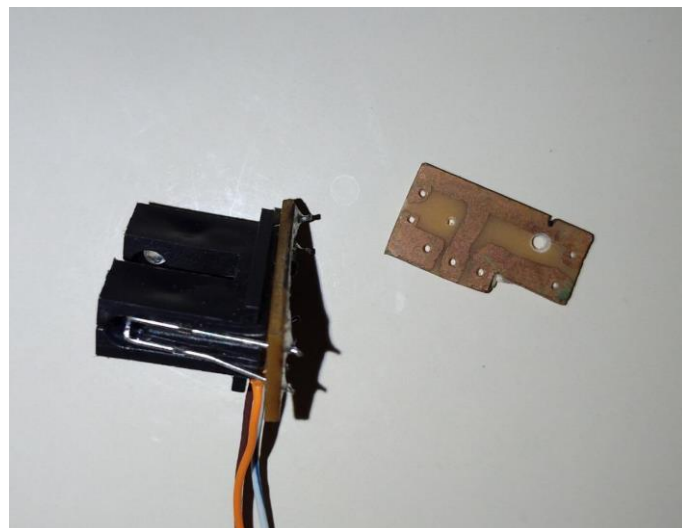


Figura 2 - Placa da barreira óptica Placa da barreira óptica



Figura 3 - Sistema mecânico Sistema mecânico

### 6.3.3 Registrador de deslocamento

Foram utilizados 4 registradores de deslocamento 74595, que é um registrador de deslocamento em série, arranjados em cascata. Portanto ele necessita da entrada de sinal de dados e de sinal de clock.

Eles foram utilizados para comandar os displays dos placares, sendo um registrador para cada display, distribuídos da seguinte maneira: dois para unidade e dezena do jogador 1 e os outros dois para o jogador 2.

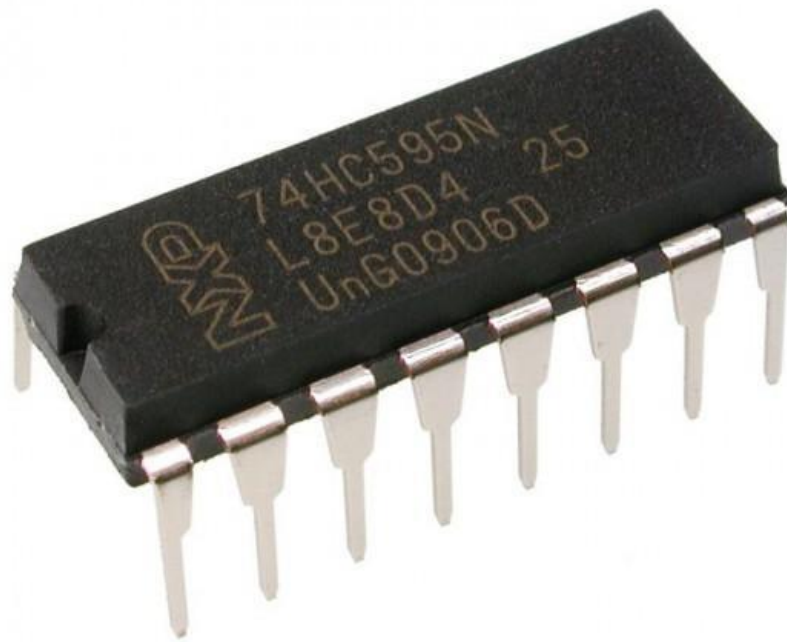


Figura 4 - CI 74595 CI 74595

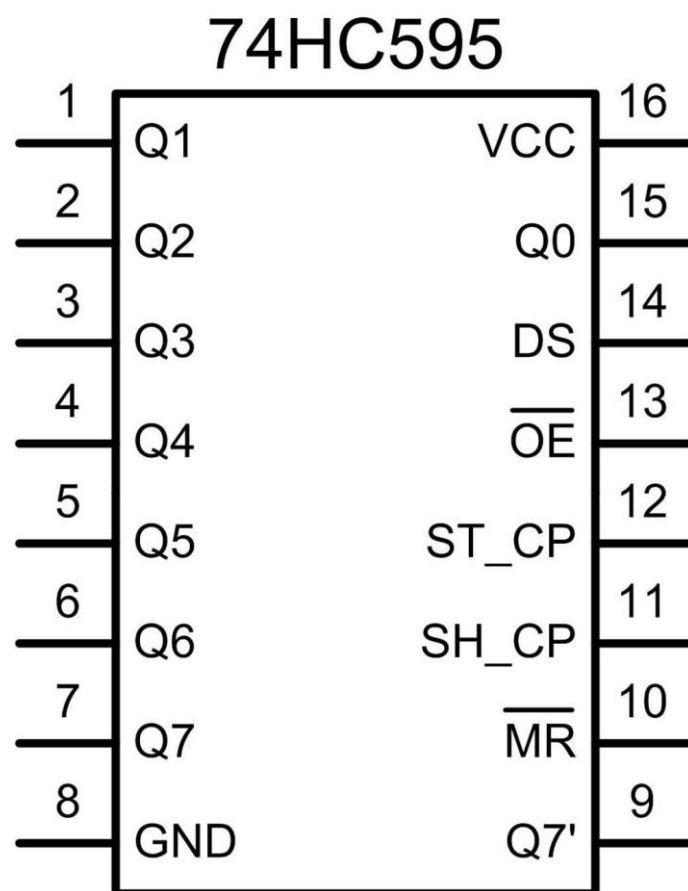


Figura 5 - Diagrama do CI 74595 Diagrama do CI 74595

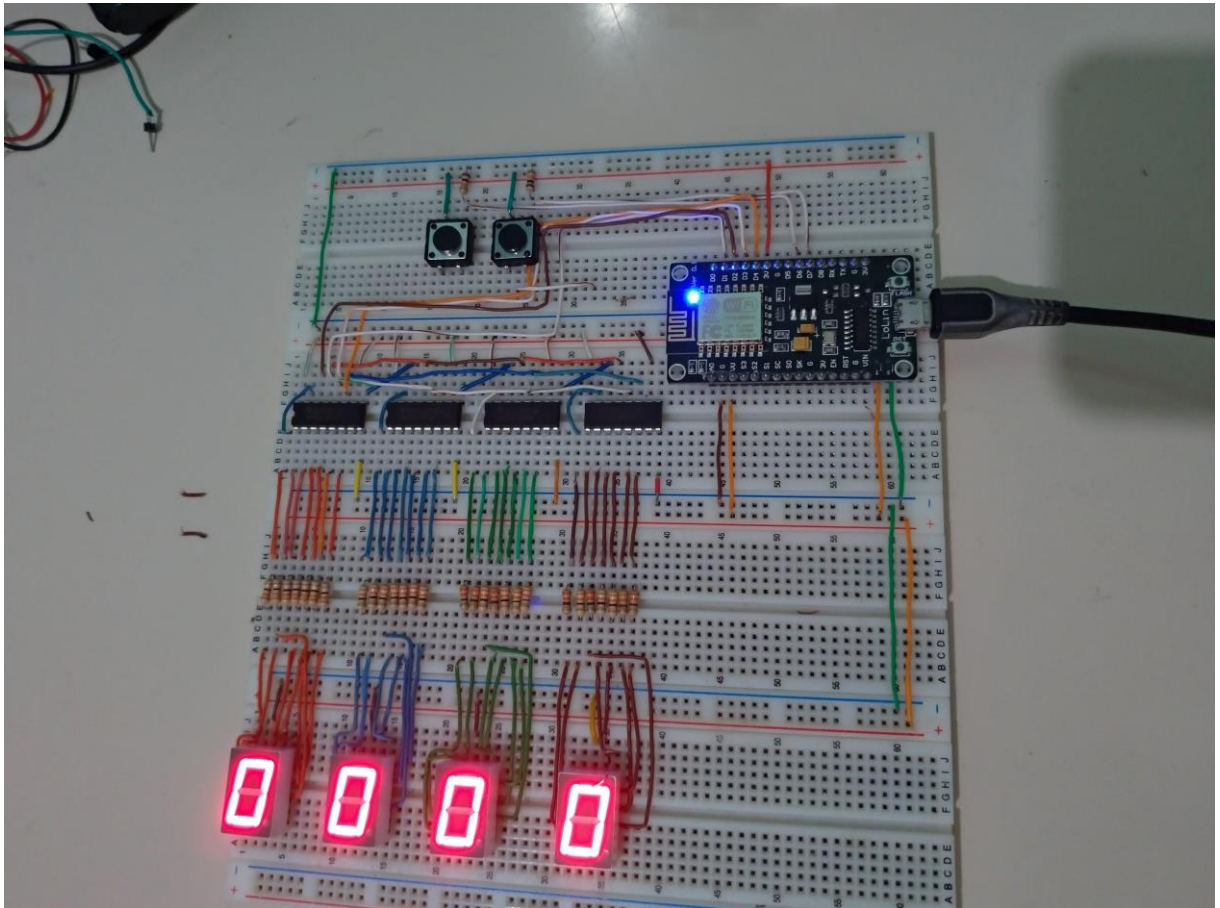


Figura 6 - Registradores e displays montados no protoboard

#### 6.3.4 Microcontroladores

Microcontrolador é um pequeno computador num único circuito integrado o qual contém um núcleo de processador, memória e periféricos programáveis de entrada e saída. A memória de programação pode ser RAM, NOR flash ou PROM a qual, muitas vezes, é incluída no chip.

O microcontrolador utilizado foi o Esp-8266 da fabricante Espressif. Esse microcontrolador pode ser encontrado com diferentes quantidades de entrada e saída, foi utilizado o Esp-01 que possui 4 Entradas / Saídas para Uso Geral (GPIO) e o Esp-12 que possui 17 GPIOs na placa de prototipagem NodeMCU.

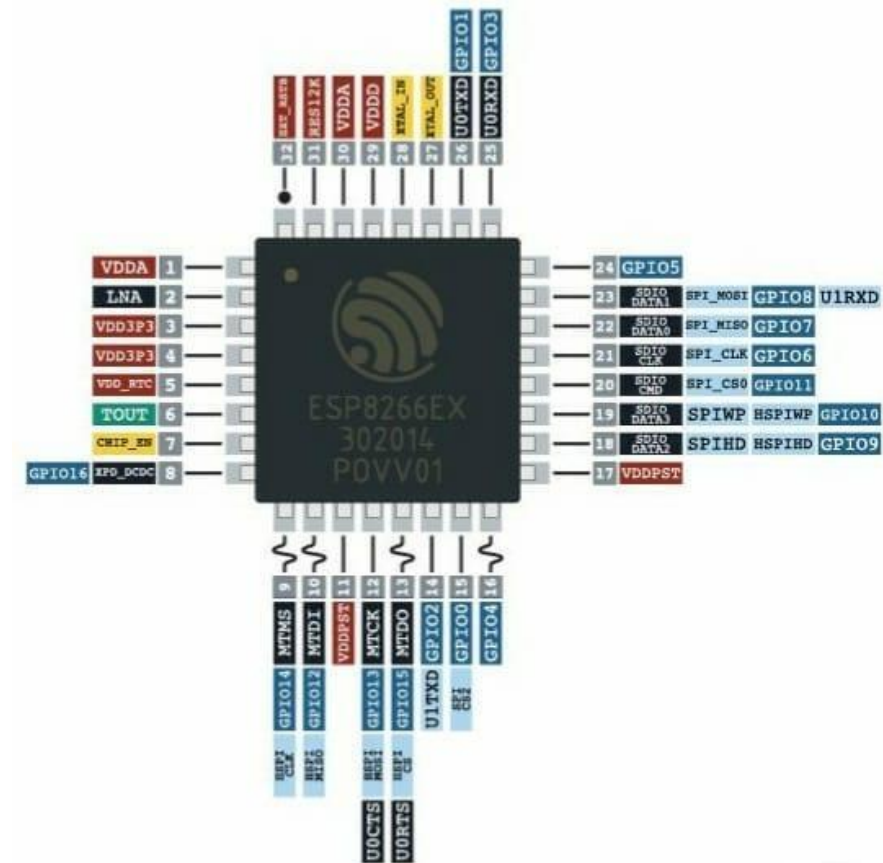


Figura 7 - Diagrama da Esp-8266

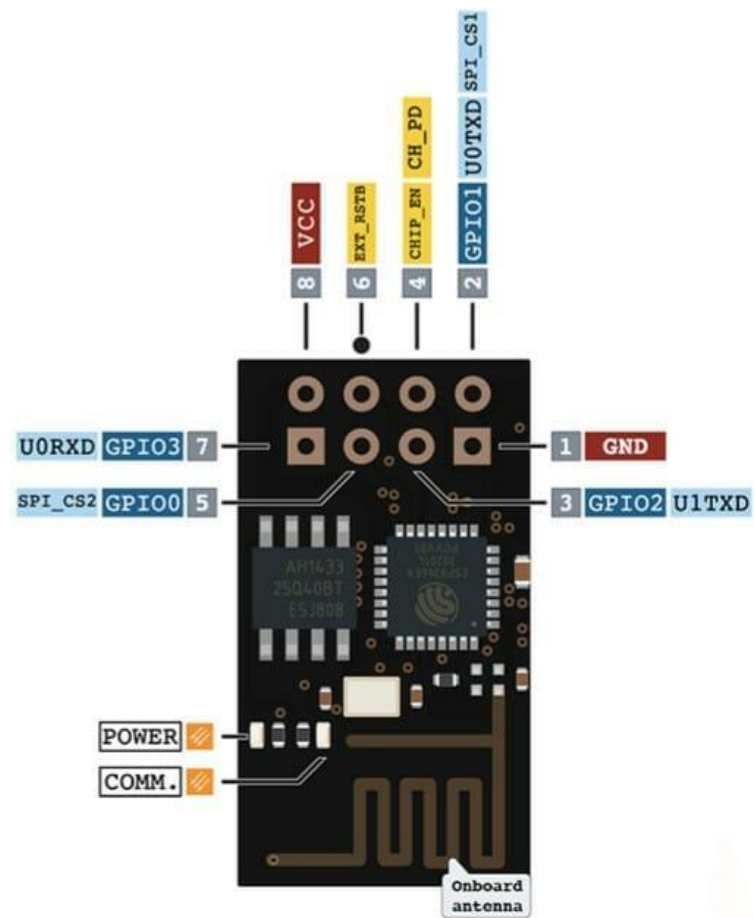
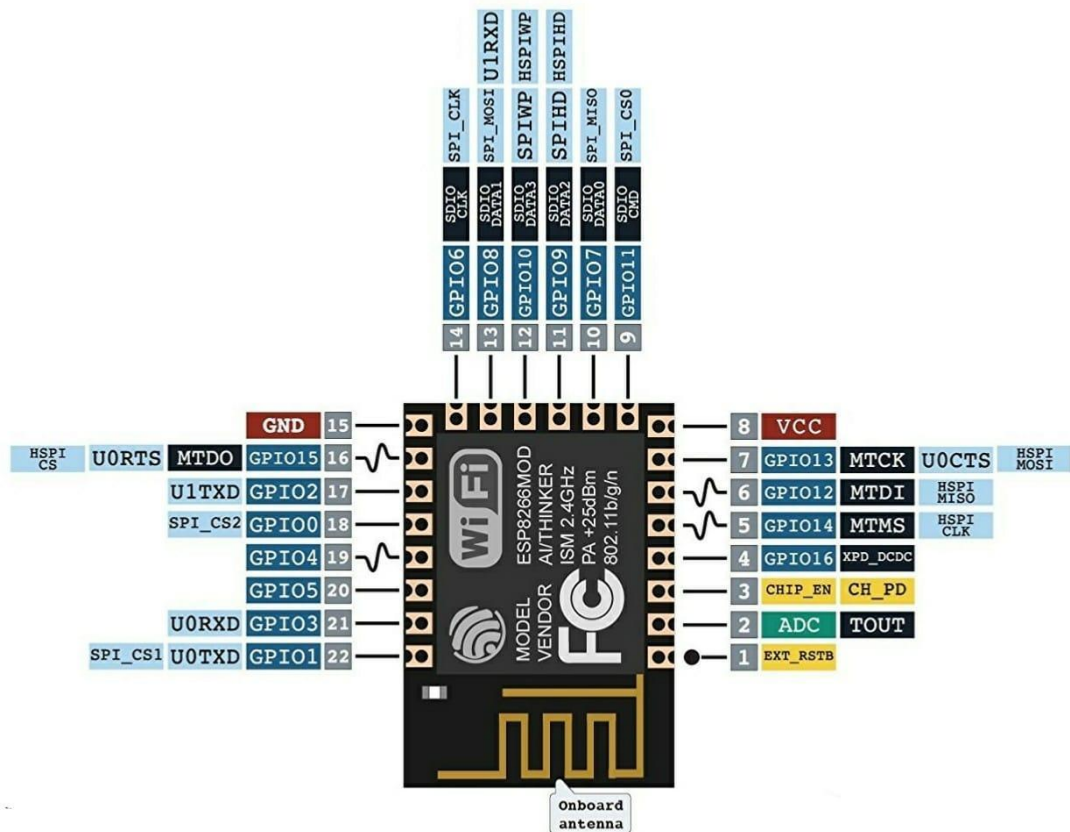


Figura 8 - Diagrama da Esp-01





### 6.3.5 Programação C++

O microcontrolador Esp-8266 pode ser programado em Lua, Python e C++. Escolhemos a linguagem de programação C++, porque foi a que tivemos contato durante o curso, além de ser a linguagem mais poderosa do mundo.

#### 6.3.5.1 Microsoft Visual Studio Code

Tivemos também que escolher um Ambiente Integral de Desenvolvimento (IDE). Existem milhares de editores de textos e IDEs para se utilizar, nós escolhemos o editor de texto da Microsoft Visual Studio Code, um software gratuito e open source.

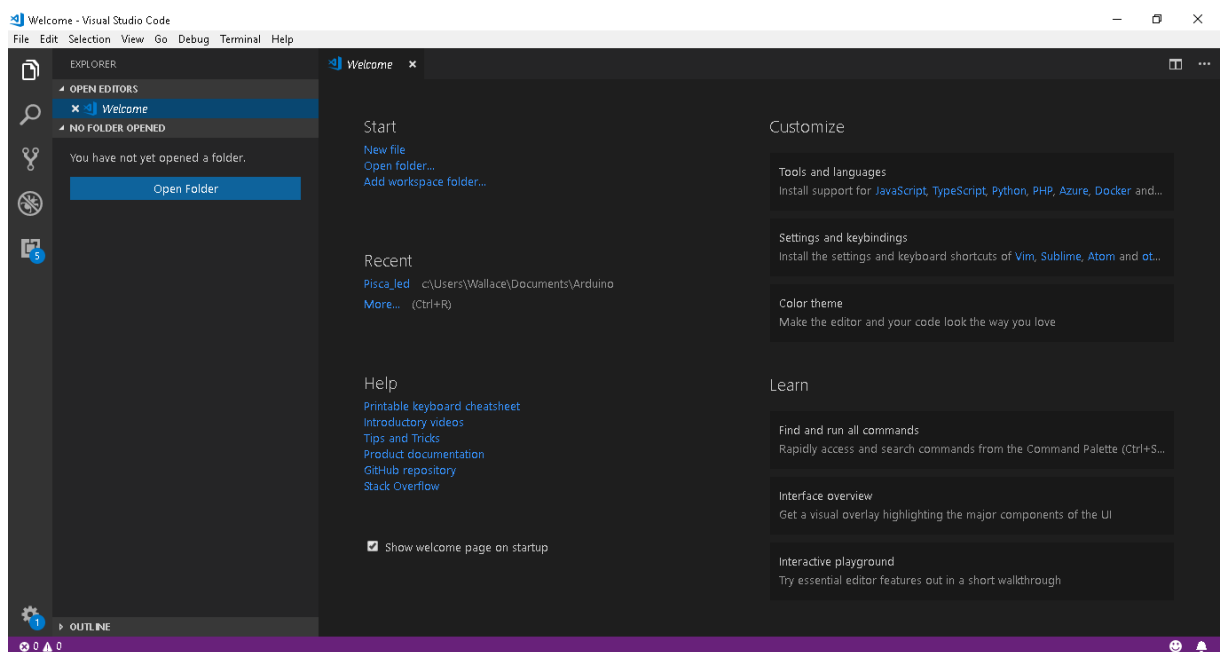


Figura 11 - Área de trabalho do VS Code

### 6.3.5.2 Código comentado

```

1
2  #include "ESP8266WiFi.h"// INCLUE BIBLIOTECA WIFI PARA ESP8266
3
4  const unsigned char PIN_CESTA = 13;//DEFINE O A PORTA DO SENSOR OPTICO DA CESTA
5  const unsigned char PIN_RESET = 12;//DEFINE A PORTA DO BOTAO RESET DO PLACAR
6
7
8  const unsigned char NDISPLAYS = 4; //DEFINE NUMERO DE DISPLAYS DE 7 SEGMENTOS
9  const unsigned char N_JOGADORES = 10; // DEFINE O NUMERO DE JOGADORES MAXIMO
10
11  const int32_t dist3 = -74;//VARIABEL QUE DENOMINA DE QUAL ATENUAÇÃO SERA O PONTO
12
13  const unsigned char PIN_CLK = 2;// DEFINE O PINO CLOCK
14  const unsigned char PIN_DT = 4;//DEFINE O PINO DE DADOS
15  const unsigned char PIN_ENB = 0;// DEFINE O PINO ENABLE
16  const unsigned char PIN_OE = 5;//DEFINE O PINO ENABLE OUTPUT
17
18
19
20  volatile int state_Cesta;//CRIAR UMA VARIABEL DA CESTA
21  volatile int state_Reset;//CRIAR UMA VARIABEL DO RESET
22
23  void inline handle_PIN_CESTA(void)// FUNÇÃO DE MUDANÇA DA VARIABEL CESTA
24  {
25
26      state_Cesta = 1;
27
28  }
29
30  void inline handle_PIN_RESET(void)// FUNÇÃO DE MUDANÇA DA VARIABEL RESET
31  {
32
33      state_Reset = 1;
34
35  }
36
37
38  const unsigned char tdelay1 = 500;// DEFINE VALOR DO TDELAY 1
39  const unsigned char tdelay2 = 200;// DEFINE VALOR DO TDELAY 2
40

```

Figura 12 - Código comentado parte 1

```
41
42 int dcb(char num)// FUNCAO DA CONVERSAO DE NUMERO DECIMAL PARA BINARIO PARA O CI 74595
43 {
44     switch (num)
45     {
46
47         case 0:
48             return 0b11111010;
49         case 1:
50             return 0b00110000;
51         case 2:
52             return 0b11011100;
53         case 3:
54             return 0b01111100;
55         case 4:
56             return 0b00110110;
57         case 5:
58             return 0b01101110;
59         case 6:
60             return 0b11101110;
61         case 7:
62             return 0b00111000;
63         case 8:
64             return 0b11111110;
65         case 9:
66             return 0b01111110;
67         default:
68             return 0;
69     }
70 }
71
72 void escreve(int nt){ //FUNCAO QUE ENVIA OS DADOS PARA O CI 74595
73     char i = 0;
74
75     digitalWrite(PIN_ENB, LOW);
76     while (nt)
77     {
78         shiftOut(PIN_DT, PIN_CLK, LSBFIRST, dcb(nt % 10));
79         nt /= 10;
80     }
```

Figura 13 - Código comentado parte 2

```

80     i++;
81 }
82 for (; i < NDISPLAYS; i++)
83 {
84     shiftOut(PIN_DT, PIN_CLK, LSBFIRST, dcb(0));
85 }
86 digitalWrite(PIN_ENB, HIGH);
87 }
88
89
90
91
92 void apaga(void){// FUNÇÃO QUE FAZ O DISPLAY PISCAR
93 #ifndef efeito
94     digitalWrite(PIN_ENB, LOW);
95 #endif
96 for (char i=0; i < NDISPLAYS; i++)
97 {
98     shiftOut(PIN_DT, PIN_CLK, LSBFIRST, dcb(-1));
99 #ifndef efeito
100     delay(100);
101 #endif
102 }
103 #ifndef efeito
104     digitalWrite(PIN_ENB, HIGH);
105 #endif
106 }
107
108 void Contador(int v1, int v2, int pisca){// FUNÇÃO QUE INFORMA O VALOR PARA DEMONSTRAR NO DISPLAY
109
110     int n=100 * v1 + v2;
111
112     char i = 0;
113     escreve(n);
114     while(pisca--){
115         apaga();
116         delay(tdelay1);
117         escreve(n);
118         delay(tdelay2);
119     }

```

Figura 14 - Código comentado parte 3

```

120 }
121
122
123 void placartime(int valora, int valorb, int pisca)// FUNÇÃO QUE DEFINE O PLACAR DO TIME NO SERIAL MONITOR
124 {
125     Serial.print("Time A: ");
126     Serial.print(valora);
127     Serial.print(" x Time B: ");
128     Serial.println(valorb);
129     Contador(valora, valorb, pisca);
130 }
131
132 void setup(void)// FUNÇÃO DE SETUP
133 {
134     Serial.begin(115200);// DEFINE A VELOCIDADE DA PORTA SERIAL
135
136
137     pinMode(PIN_CESTA, INPUT);// DEFINE O PINO CESTA COMO ENTRADA
138     pinMode(PIN_CLK, OUTPUT);// DEFINE O PINO CLOCK COMO SAIDA
139     pinMode(PIN_DT, OUTPUT);//DEFINE O PINO DATA COMO SAIDA
140     pinMode(PIN_ENB, OUTPUT);//DEFINE O PINO ENABLE COMO SAIDA
141     pinMode(PIN_OE, OUTPUT);//DEFINE O PINO ENABLE OUTPUT COMO SAIDA
142     digitalWrite(PIN_OE, LOW);// DEFINE PINO OE COMO BAIXO
143
144     attachInterrupt(PIN_CESTA, handle_PIN_CESTA, FALLING);// FUNÇÃO DE INTERRUPÇÃO EM FUNÇÃO DO PINO CESTA
145     attachInterrupt(PIN_RESET, handle_PIN_RESET, FALLING);// FUNÇÃO DE INTERRUPÇÃO EM FUNÇÃO DO PINO RESET
146     Wifi.mode(WIFI_STA);// DEFINE O WIFI DA ESP COMO STATION
147     Wifi.disconnect();// DESCONECTA DE QUALQUER WIFI QUE A ESP POSSA TER SE CONECTADO
148     delay(100);// DA UM DE DA UM DELAY DE 100ms
149
150     Serial.println("Setup done");// ESCRIVE NO SERIAL MONITOR SETUP DONE
151     placartime(0, 0, 0);// ZERA O PLACAR
152 }
153
154 static int timea = 0, timeb = 0;// INICIA O PLACAR DO TIME A E TIME B COM 0
155
156 void loop(void)// FUNÇÃO DE LOOP
157 {
158
159     if (state_Cesta)// VERIFICA ESTADO DA VARIÁVEL CESTA

```

Figura 15 - Código comentado parte 4

```

160 {
161
162     int n = Wifi.scanNetworks();// NUMERO DE WIFI ENCONTRADOS
163     int njog = 0;//IDENTIFICAÇÃO PARA OS VETORES
164     Serial.println("Cesta!");// ESCRIBE CESTA! NO SERIAL MOTNITOR
165
166     String jog[N_JOGADORES];// CRIA UM VETOR DE STRINGS
167     int32_t dist[N_JOGADORES];// CRIA UM VETOR DE NUMEROS INTEIROS
168
169     for (int nt = 0; nt < n && nt < N_JOGADORES; nt++)// ENQUANTO NT FOR MENOR QUE O NUMERO DE JOGADORES
170     {
171         if (Wifi.SSID(nt)[0] == '$' || Wifi.SSID(nt)[0] == '#')// IDENTIFICA OS JOGADORES DE ACORDO COM OS CARACTERES $ #
172         {
173             jog[njog] = Wifi.SSID(nt);// ADICIONA NO VETOR DE STRING OS JOGADORES IDENTIFICADOS
174             dist[njog] = Wifi.RSSI(nt);// ADICIONA NO VETOR DE NUMEROS INTEIROS AS ATENUAÇÕES DOS JOGADORES
175             njog++;
176         }
177     }
178
179     if (njog == 0)// CASO NUMERO DE JOGADORES FOR 0
180     {
181         Serial.println("Não há jogadores por perto");// ESCRIBE NO SERIAL MONITOR QUE NÃO HÁ JOGADORES POR PERTO!
182     }
183     else // CASO SEJA DIFERENTE DE 0
184     {
185
186         int flag;// VARIÁVEL FLAG
187         do// CONDIÇÃO WHILE DO
188         {
189             flag = 0;// INICIA FLAG COM 0
190             for (int i = 0; i < njog; i++)// ENQUANTO I FOR MENOR QUE NUMERO DE JOGADORES
191             {
192                 Serial.println(String("Jogador[" + i + String("): ") + jog[i] + " Distancia: " + dist[i]);/* ESCRIBE NO
193                 SERIAL MONITOR OS JOGADORES E SUAS ATENUAÇÕES
194                 */
195             }
196
197             Serial.println("Quem fez a cesta? Digite o número:");// PERGUNTA NO SERIAL MONITOR QUE JOGADOR FEZ CESTA
198

```

Figura 16 - Código comentado parte 5

```

199     while (!Serial.available());// FICA ESPERANDO UMA RESPOSTA DE QUEM FEZ A CESTA
200
201     int menujogador = Serial.parseInt();// ESCRIBE O A RESPOSTA NA VARIÁVEL MENUJOGADOR
202
203     if (menujogador >= 0 && menujogador < njog)// VERIFICA SE A RESPOSTA É VÁLIDA
204     {
205         int32_t atenuacao = dist[menujogador];// PEGA A ATENUAÇÃO DO JOGADOR ESCOLIDO
206         int tres = (atenuacao < dist3) ? 3 : 2;// VERIFICA SE A ATENUAÇÃO CORRESPONDE A UMA CESTA DE 3 PONTOS
207         switch (jog[menujogador][0])// VERIFICA SE O JOGADOR ESCOLIDO É DO TIME A OU B
208         {
209             case '$':
210             {
211                 timea += tres;// SOMA 3 PONTOS NO TIME A
212             }
213             break;
214             case '#':
215             {
216                 timeb += tres;// SOMA 3 PONTOS NO TIME B
217             }
218             break;
219         }
220     }
221     else // CASO SEJA UMA RESPOSTA INVÁLIDA
222     {
223         flag = 1;// DEFINE FLAG COMO 1
224         Serial.println("Opcao invalida");// ESCRIBE NO SERIAL MONITOR OPÇÃO INVÁLIDA!
225     }
226 } while (flag);// CASO FLAG SEJA VERDADEIRO VOLTA NO DO
227
228 placartime(timea, timeb, 3);// EXIBE PLACAR ATUAL NO DISPLAY
229
230
231
232 }
233 Serial.println("");// PULA UMA LINHA NO SERIAL MONITOR
234 state_Cesta = 0;// DEFINE ESTADO DA CESTA COMO 0
235 }
236
237 if (state_Reset)// SE RESET FOR VERDADEIRO
238 {

```

Figura 17 - Código comentado parte 6

```

239     timea = 0; // ZERRA O PLACAR TIME A
240     timeb = 0; // ZERRA O PLACAR TIME B
241     placartime(0, 0, 3); // EXIBE PLACAR ATUAL NO DISPLAY
242     delay(100); // DA UM DELAY DE 100ms
243     state_Reset = 0; // DEFINE ESTADO DA RESET COMO 0
244 }
245 }

```

Figura 18 - Código comentado parte 7

## 6.4 Circuito elétrico

Para projetar o circuito elétrico levamos em consideração a quantidade de GPIOs da placa de prototipagem NodeMCU e a sua utilização de um baixo nível de tensão, pois usa apenas 3.3v como alimentação e em suas GPIOs.

### 6.4.1 Placar

O placar é composto por quatro displays catodo comum (D4001A/B) e 28 transistores BC328 do tipo PNP chaveando 12v para os displays.

Divididos esse conjunto em duas placas separadas, contando com dois displays e 14 transistores em cada.

### 6.4.2 Transmissor do jogador

Foi utilizado uma Esp-01 no modo access point (ponto de acesso ou roteador) para cada jogador

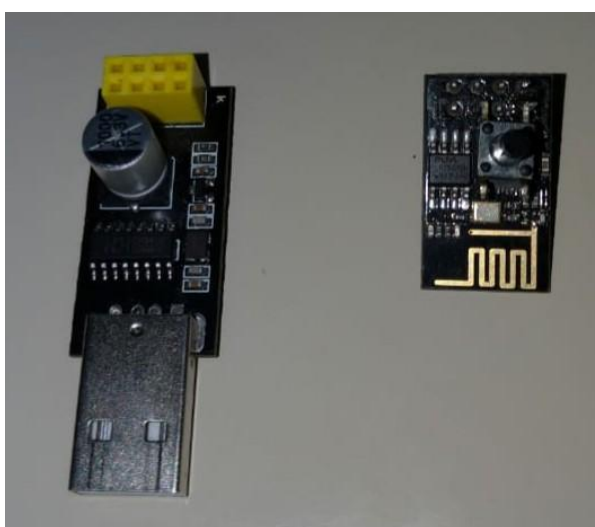


Figura 19 Transmissor fora da caixa





Figura 20 - Transmissores dentro da caixa

### 6.4.3 Receptor

Utilizamos a NodeMcu no modo Station (estação ou receptor WiFi) para medir a atenuação do WiFi transmitido pelas Esp-01. A nodeMCU também ficou responsável pela parte da lógica de contagem e a de enviar para o registrador de deslocamento para demonstrar o placar no display.

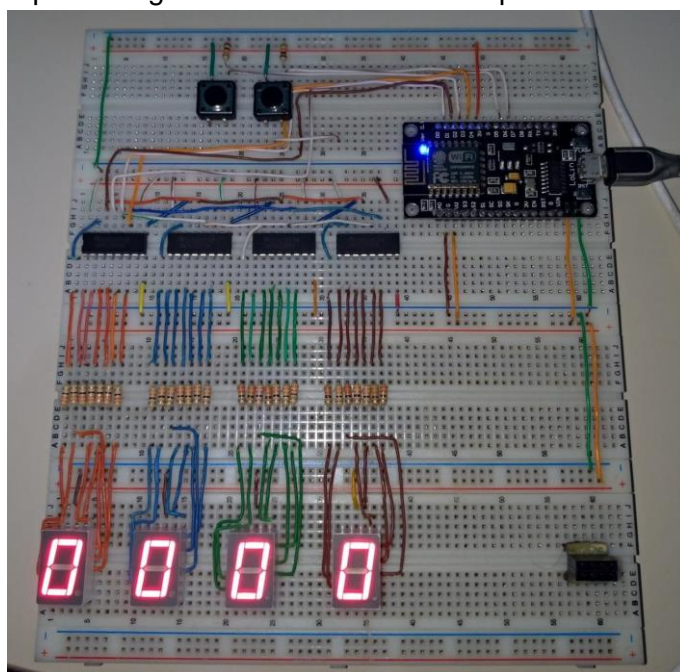


Figura 21 - NodeMCU com 74595

### 6.4.4 Proteus

Para desenhar o circuito elétrico e confeccionar o PCB deste projeto foi utilizado o software Proteus na versão 8.5. O Proteus é utilizado para desenhar e simular projetos eletrônicos, possuem ferramentas que possibilitam a impressão do circuito e pode ser usado para confecção de PCB (Placas de Circuito Impresso). Abaixo a demonstração da tela de trabalho do software de desenho do circuito elétrico e de placa de circuito impresso. Posteriormente o circuito feito para o placar automático de basquete.

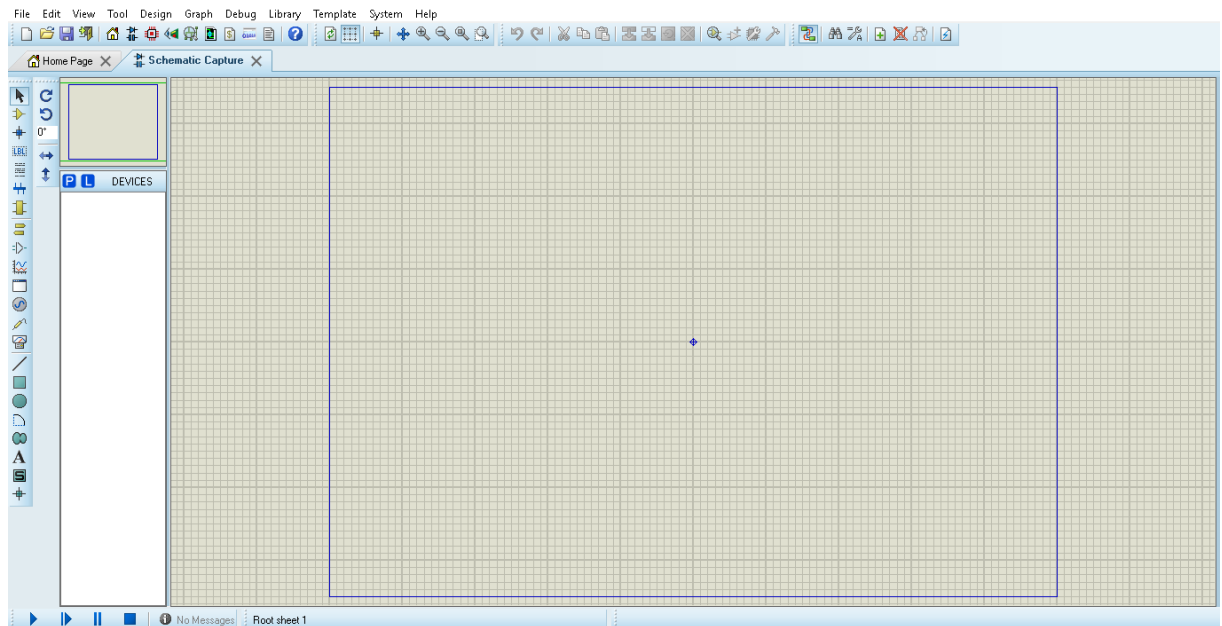


Figura 22 - Tela de trabalho do software de circuito elétrico ISIS Proteus 8.5

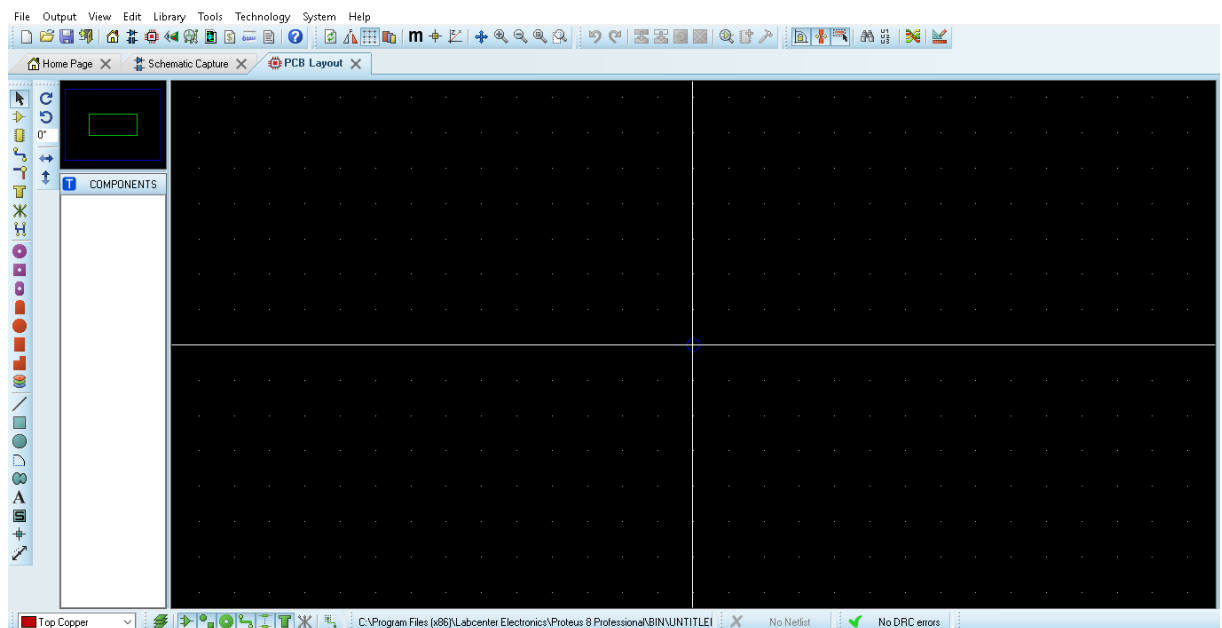


Figura 23 - Tela de trabalho do software de PCB ARES Proteus 8.5

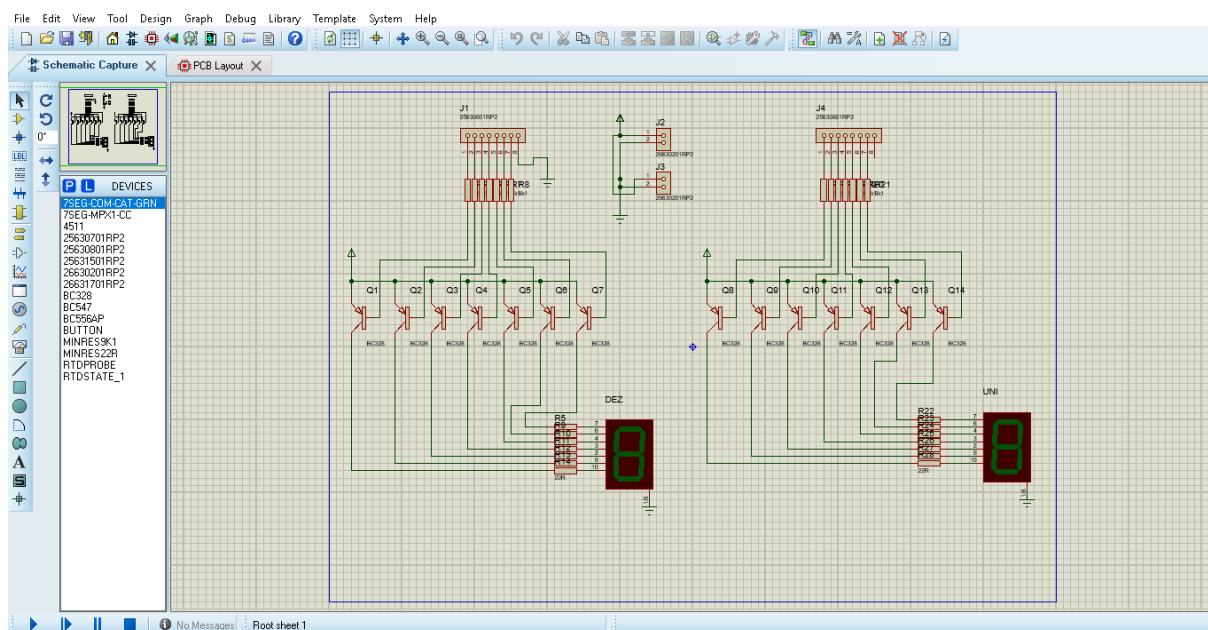


Figura 24 - Circuito Elétrico dos displays feito no proteus

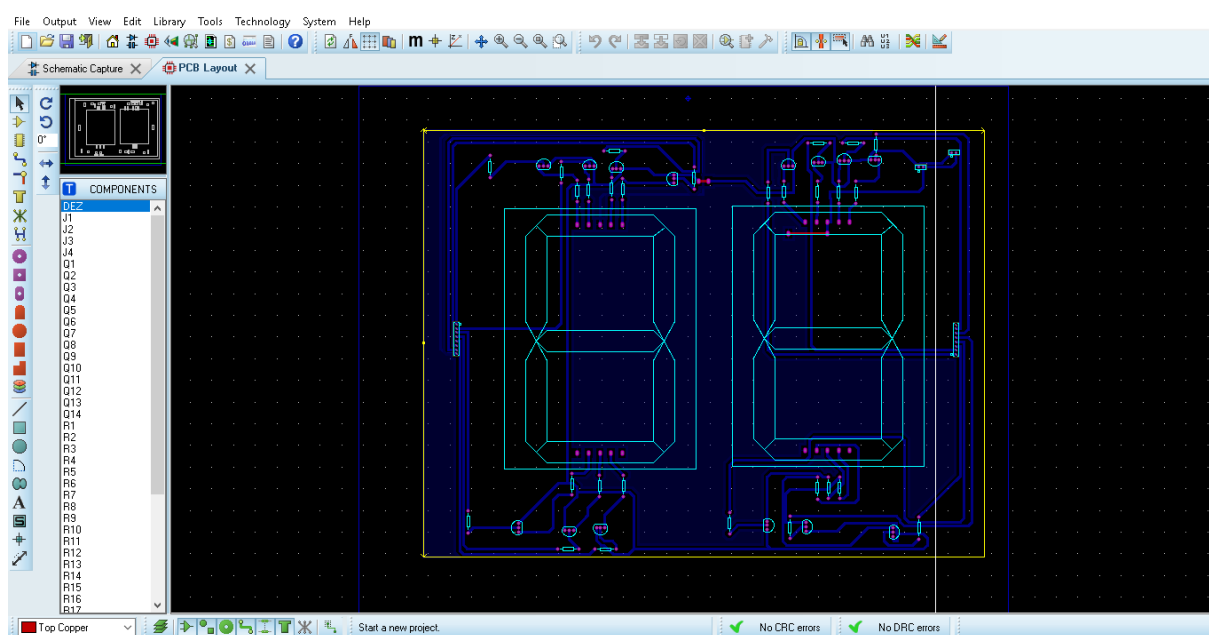


Figura 25 - Placa PCB no Proteus

## 6.5 Montagem

Na montagem geral do projeto foram utilizados materiais baratos e de fácil acesso, devido ao objetivo de realizar apenas um protótipo do projeto.

### 6.5.1 Confeção

Para executar a montagem da estrutura foi utilizado madeira, barra de ferro e uma base de ferro. As ferramentas utilizadas foram: serra tico-tico e furadeira.

## 6.6 Testes e simulações

Primeiro foi realizado teste de um contador de 0 até 99 utilizando o PIC 18f4550.

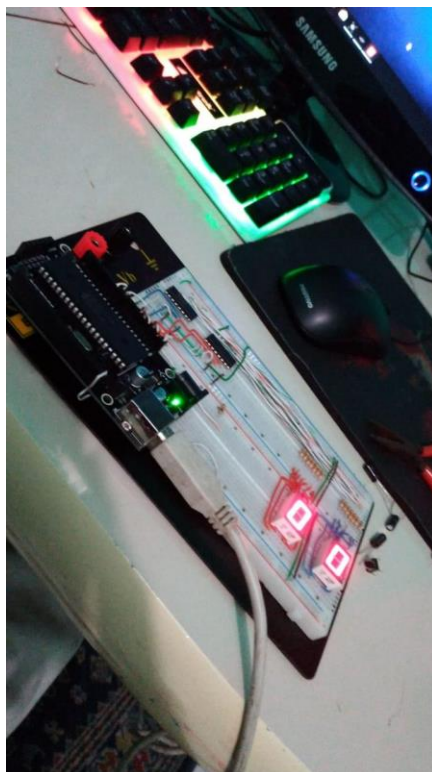


Figura 26 - Contador com PIC18f4550 e 4511

Segundo foi um contador que utilizava o sensor de bola como pulso para a acreção dos números e com o display grande.

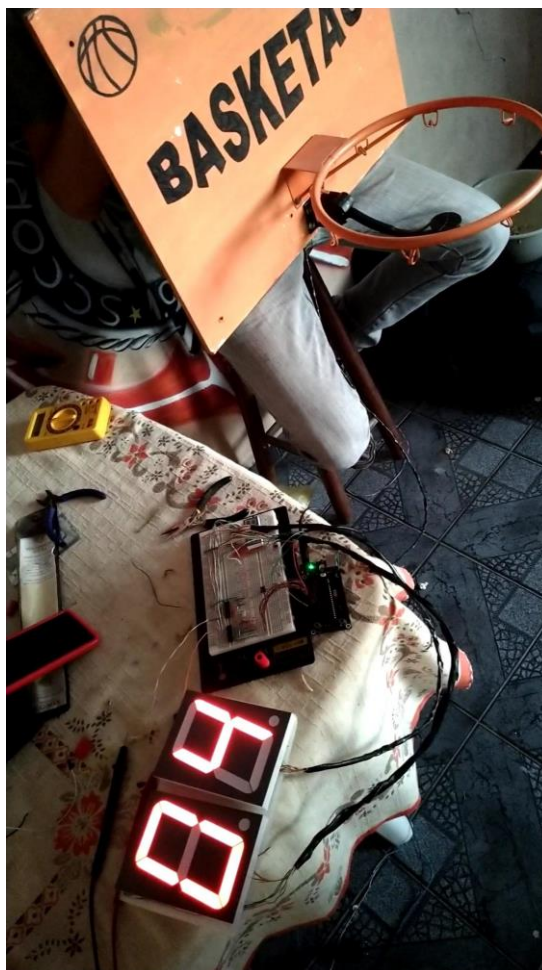


Figura 27 - Contador com o PIC18f4550 com o sensor da bola e os displays grande.

Terceiro foi um contador de 0 até 999 com o registrador de deslocamento e o Arduino Mega 2560.



Figura 28 - Contador com o Arduino Mega 2560 e registrador de deslocamento



Quarto foi um contador com variação do valor dos pontos conforme era apertado o botão e o sensor da bola era acionado, foi utilizado a NodeMCU.

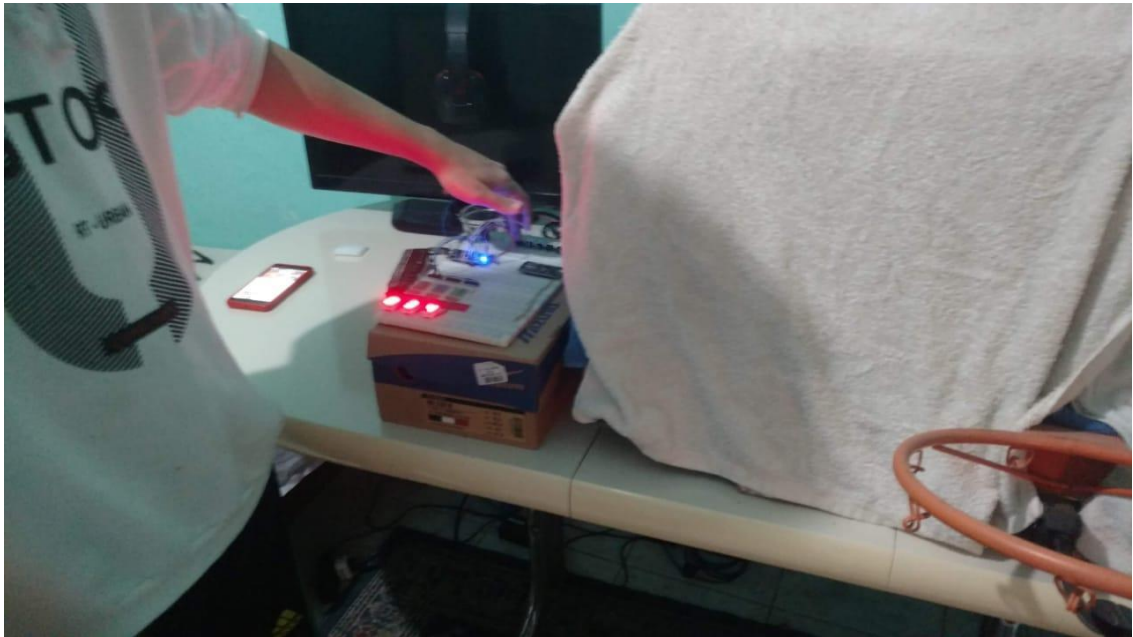


Figura 29 - Contador com variação de pontos NodeMCU

Quinto foi um contador com a variação dos pontos conforme a distância dos jogadores.

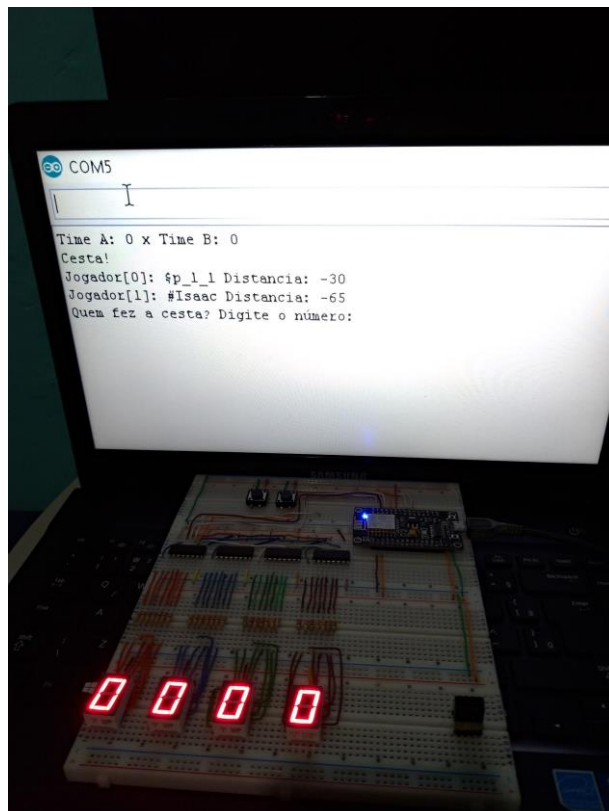


Figura 30 - Contador com a variação dos pontos conforme a distância

## 6.7 Problemas e soluções

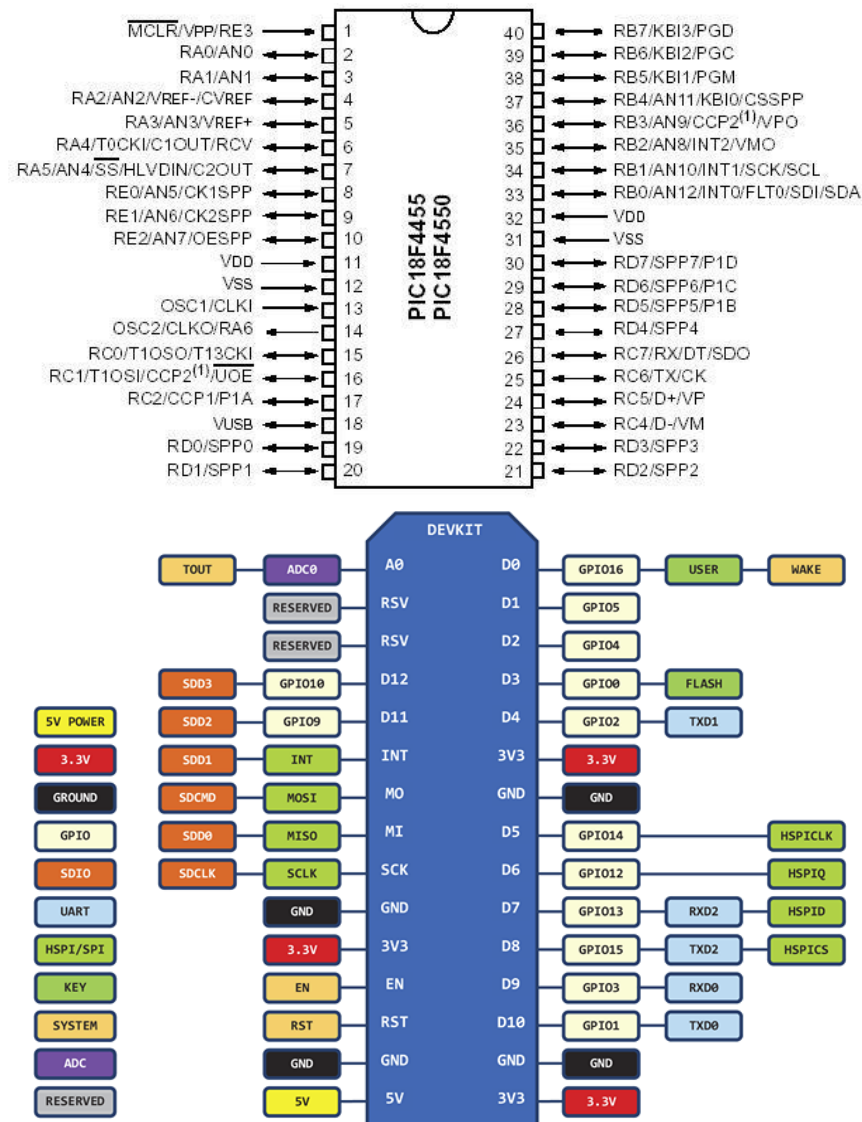
Em um dos testes foi observado que o sensor da bola, por ser mecânico na hora que voltava rebatia e dava mais de um pulso para o microcontrolador. Como solução foi colocar um delay de 1 segundo para a próxima verificação de pulso do sensor.

```
while (1)
{
    DISPLAYUNI();
    DISPLAYDEZ();
    if (sen)
    {
        VALORDCESTA();
        uni=uni+val_ces;
        DISPLAYUNI();
        DISPLAYDEZ();
        delay_ms(1000);
    }
}
```

Figura 31 - Solução MikroC Solução MikroC



Também foi observado que a NodeMCU tinha uma quantidade de GPIOs bem menor do que o PIC18f4550, por isso resolvemos utilizar o registrador de deslocamento 74595 ao invés do decodificador 4511.



*D0(GPIO16) can only be used as gpio read/write, no interrupt supported, no pwm/i2c/ow supported.*

Figura 32 - Comparação de GPIOs entre PIC18f4550 e NodeMCU

## 7 CONCLUSÃO

Dessa forma este projeto contribui para a diminuição da margem de erros da somatória de pontos e do tempo gasto pelo árbitro para execução deste serviço, também estamos contribuindo para a revolução na maneira que os esportes são praticados, trazendo a eletrônica para o mundo dos esportes. Além disso, nele se aplica vários conceitos adquiridos durante o curso com o efeito de fixação do aprendizado.

## 8 REFERÊNCIAS

PONTO fantasma no basquete, São Paulo, [S.l.:s.n.], 2003

Disponível em:

<<https://www1.folha.uol.com.br/fsp/esporte/fk0908200308.htm>> Acesso em: 18/09/2018 - 20h30min

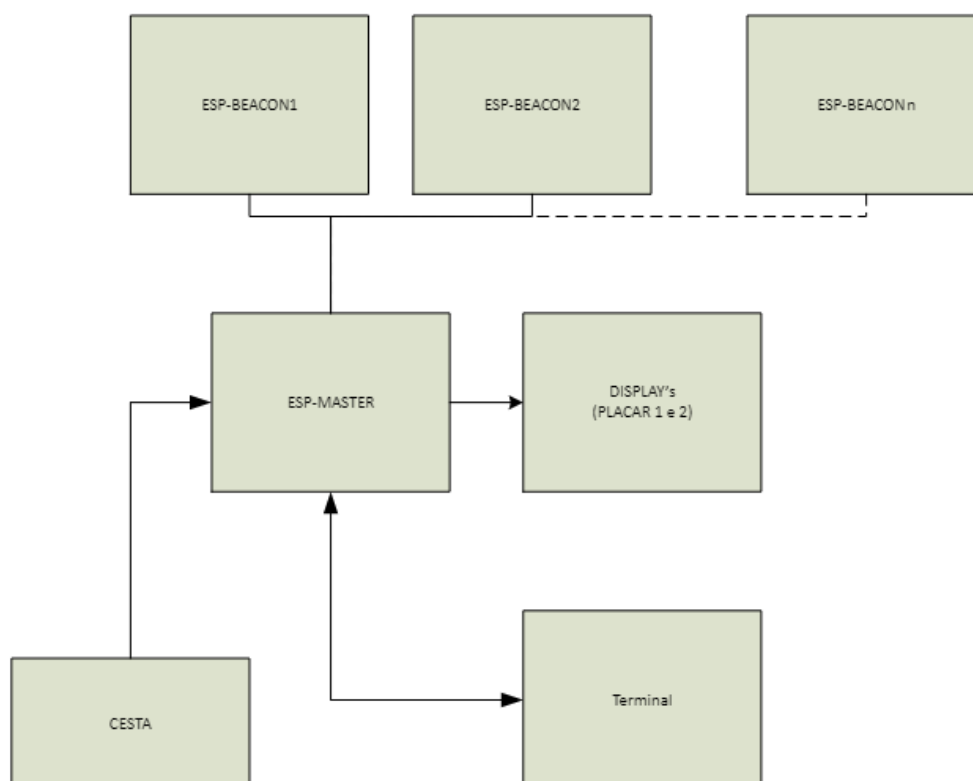
REGRAS oficiais basquetebol, Brasil, [S.l.:s.n.], 2017

Disponível em:

<<http://www.cbb.com.br/comum/code/MostrarArquivo.php?C=MzM0>> Acesso em: 18/09/2018 - 21h

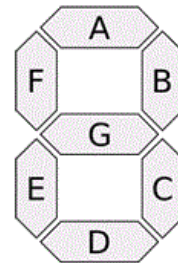
## 9 APÊNDICES

### 9.1 Diagrama em Blocos



## 9.2 Tabela Verdade Display 7 Seguimentos

DECECIMAL	A	B	C	D	E	F	G
0	1	1	1	1	1	1	0
1	0	1	1	0	0	0	0
2	1	1	1	1	0	0	1
3	1	1	1	1	0	0	1
4	0	1	1	0	0	1	1
5	1	0	1	1	0	1	1
6	1	0	1	1	1	1	1
7	1	1	1	0	0	0	0
8	1	1	1	1	1	1	1
9	1	1	1	1	0	1	1



## 9.3 Tabela de preços sem doação

TABELA DE CUSTO ESTIMADO		
ITEM	QUANTIDADE	PREÇO
NomdeMCU	2	R\$ 20,60
Esp01	2	R\$ 35,63
Display	4	R\$ 120
Placa de Fenolite	3	R\$ 30,00
Transistor	30	R\$ 9,00
CI	4	R\$ 6,00
Tabela	1	R\$ 40,00
Caixa Esp01	2	R\$ 5,00
Base de Ferro	1	Doação
Tubo de Ferro Aço inox	1	Doação
	<b>TOTAL</b>	<b>R\$ 266,23</b>

## 9.4 Tabela de preços sem doação

TABELA DE CUSTO ESTIMADO		
ITEM	QUANTIDADE	PREÇO
NomdeMCU	2	R\$ 20,60
Esp01	2	R\$ 35,63
Display	4	R\$ 120
Placa de Fenolite	3	R\$ 30,00
Transistor	30	R\$ 9,00
CI	4	R\$ 6,00
Tabela	1	R\$ 40,00
Caixa Esp01	2	R\$ 5,00
Base de Ferro	1	R\$ 400,00
Tubo de Ferro Aço inox	1	R\$ 140,00
	<b>TOTAL</b>	<b>R\$ 806,23</b>