UNIVERSIDADE DE RIBEIRÃO PRETO

ENGENHARIA DE SOFTWARE

DISCIPLINA: LÓGICA E CRIATIVIDADE – LM122A

PROFESSOR: PABLO RODRIGO SANCHES

TechParking

Murilo Ijanc - 834125

João Victor Tiezzi de Jesus - 834376

Ranniery Lucas de Carvalho Moreira - 834761

Miguel Silvério - 833895

Ribeirão Preto - SP

2020

Semestre 2020/1

SUMÁRIO

1. Introdução ……....4
1.1. Objetivo geral ……....4
1.2. Objetivos específico ……..….4
1.3. Justificativa ……5
2. Material e Métodos ……..5
3. Resultados e Discussões ……...7
3.1 O Protótipo ……..……7
3.2 Projeto Integrador ……...35
4. Conclusão ……..35
5. Referências ……....36

LISTA DE ILUSTRAÇÕES

Fotografia 1- Início da montagem……...7
Fotografia 2 - Placa ESP8266…….7
Fotografia 3 - Sensor Ultra-sônico……..7
Fotografia 4 - Medidas na caixa para o protótipo……...8
Fotografia 5 - Encapamento da caixa…….8
Fotografia 6 - Teste de leds com o servidor (vermelho desconectado).....9
Fotografia 7 - Teste de leds com o servidor (verde conectado).....9
Fotografia 8 - Caixa protetora do protótipo encapada.…….10
Fotografia 9 - Componentes conectados e colocados na caixa…….10
Fotografia 10 - Componentes conectados a energia (bateria externa).....11
Fotografia 11 - Protótipo pronto (Desconectado do servidor).....11
Fotografia 12 - Protótipo pronto (Conectado do servidor).....11
Fotografia 13 - Computador com monitoramento durante os testes…….....12
Fotografia 14 - Protótipo pronto e Posicionado para o teste final (Monitoramento).....12
Fotografia 15 - Protótipo pronto e Posicionado para o teste final (Ambiente residencial).....12
Fotografia 16 - Ícone da página Web no Smartfone……13
Fotografia 17 - Interface da página Web no Smartfone……....13

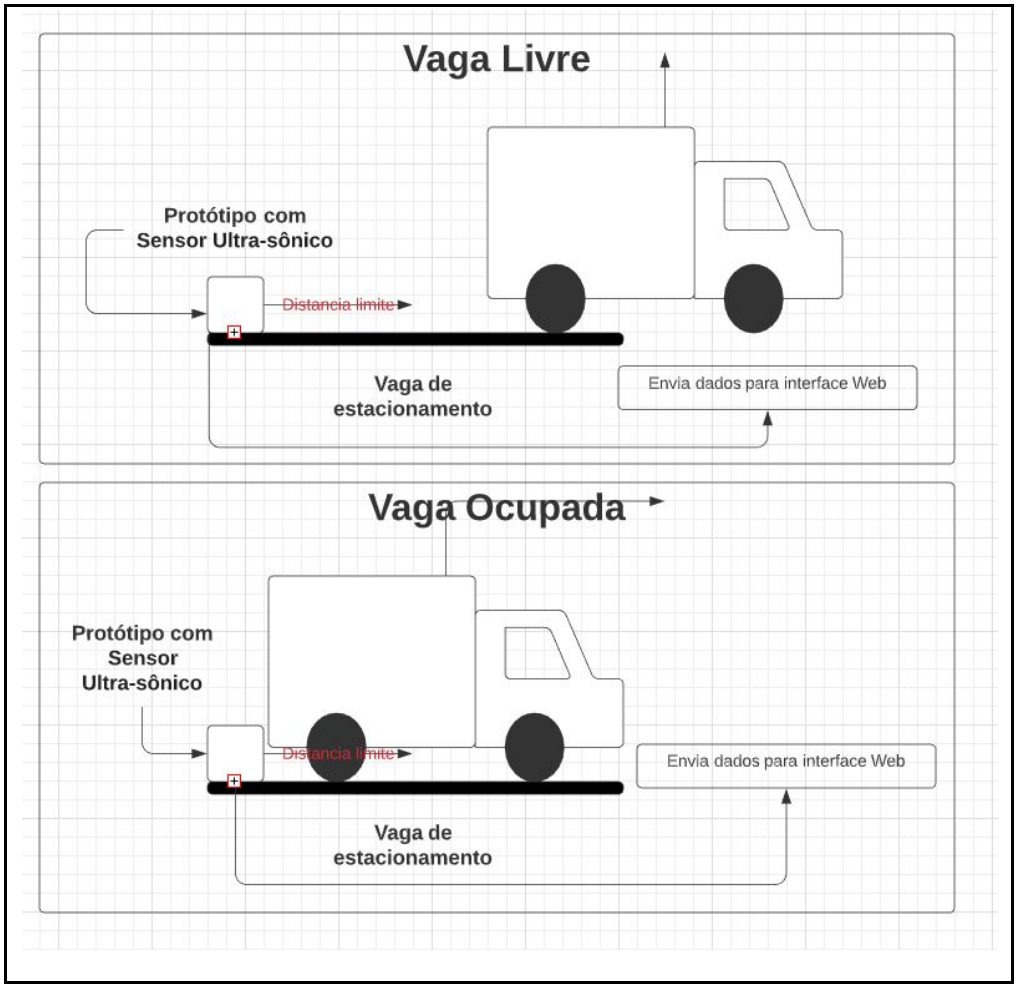
Fase 1 – Descoberta

PERSONAS

CIdadãos com pressa	Portadores de deficiência	Controle de Estacionamentos

Fase 2 – Interpretação

LEMBRETE VISUAL



Fase 3 – Ideação

TechParking

Fases 4 e 5 – Experimentação e Evolução

1. Introdução

A ideia adotada para o desenvolvimento deste projeto foi a implementação de sensores ultrassônicos que , teriam como função, a identificação da presença de veículos automotivos em vagas de estacionamento. Onde, através de uma página web, seria apresentado aos usuários a configuração da disponibilidade de vagas naquele ambiente. A motivação por trás dessa ideia é oriunda da incerteza com a qual as pessoas têm que lidar na hora de estacionarem seus veículos. Tendo que dispor de muito tempo e paciência na hora de procurar vagas em um estacionamento lotado, podendo perder compromissos e afins que venham a possuir um tempo pré estabelecido para acontecer.

1.1. Objetivo geral

O objetivo geral do projeto é identificar a disponibilidade de vagas em um determinado estacionamento, com antecedência, através de dispositivos eletrônicos.

1.2. Objetivos específico

As estratégias utilizadas para o desenvolvimento deste projeto foram discutidas através de reuniões feitas entre os integrantes do grupo, nelas foram analisados as seguintes ideias:

- Busca de um público alvo para o projeto.
- Determinar as funções feitas para obter um melhor resultado.
- Obtenção de uma melhora no acesso a vagas de um determinado estacionamento.
- Maior controle em tempo real de quantidade de veículos estacionados.
- Promover um aumento na acessibilidade de pessoas com algum tipo de deficiência ou urgência na hora de estacionar seus veículos.
- Mostrar em tempo real ao usuário o status prévio de um local de estacionamento.
- Desenvolvimento do código e montagem do protótipo.

1.3. Justificativa

Devido à rapidez em que o mundo hodierno está inserido, é preciso cada vez mais objetificar ações para que as mesmas disponham do menor tempo possível para serem realizadas. Sabendo disso, foi criado com base nesse preceito, um dispositivo que, através de sensores ultrassônicos, identifica a disponibilidade ou não de uma vaga de um estacionamento. Onde, através de uma página web, é possível identificar de forma rápida e tranquila a quantidade de vagas livres para ocupação. Evitando assim, imprevistos envolvendo tempo e transições de grandes distâncias, o que é muitas vezes um problema em casos de pessoas com deficiência.

2. Material e Métodos

Materiais e componentes utilizados/ para a realização do protótipo:

-1 Placa ESP8266.

-1 Sensor ultrassônico.

- Jumpers.

-2 Leds.

-2 Resistores 220 ohms.

-1 Caixa de acrílico.

- Fita isolante

-Bateria externa de 6.5 mAh (Para poder utilizar o protótipo em qualquer lugar)

Softwares utilizados

- Openbsd (Sistema operacional, utilizado para programar).
- Makefile.
- Arduino GCC.
- Conjunto de bibliotecas do ESP8266 (Wifi, Web Server).
- GIT .
- Websocket Javascript.
- HTML, CSS .
- VI.
- Python.

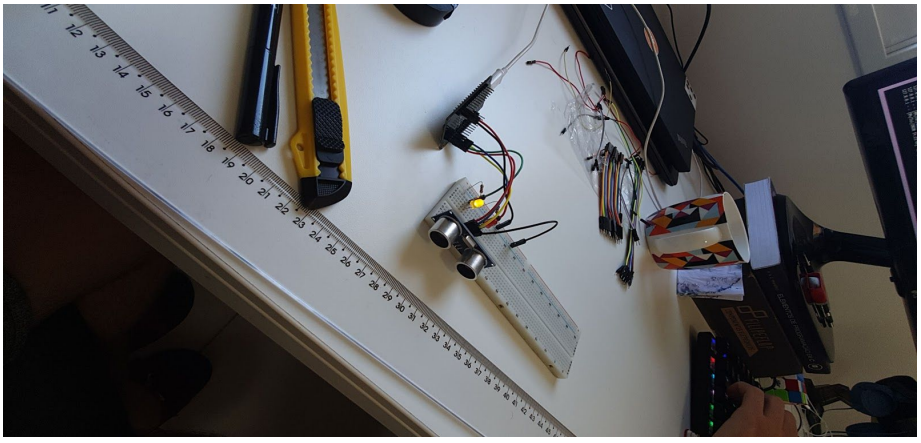
Etapas de Desenvolvimentos: Iniciamos o projeto realizando um brainstorm na qual foram levantadas as ideias para a realização do resultado final, o processo foi desenvolvido utilizando as seguintes etapas:

1. Criamos um grupo através do whatsapp para nos comunicarmos.
2. Discutimos os meios para a codificação e documentação.
3. Começamos a codificação na linguagem c/c++
    3.1. Fizemos alguns testes para verificar a funcionalidade da placa ESP8266.
    3.2. Testamos a conexão das leds, conexão wifi e conexão com o sensor.
4. Criamos e testamos a conexão com o servidor. (levantamos um servidor HTTP e realizamos requisições em um determinada url.)
5. Desenvolvemos um website no qual possui websocket para realizar as conexões com o servidor e obter as informações do status do estacionamento.
6. Fizemos alguns ajustes em determinados erros e melhoramos a interface web, criando icone e melhorando o layout.
7. Demos início na montagem do protótipo.
8. Acrescentamos leds indicadores de conectividade ao servidor.
9. Finalizamos a montagem do projeto e realizamos alguns testes de funcionalidade no ambiente residencial.

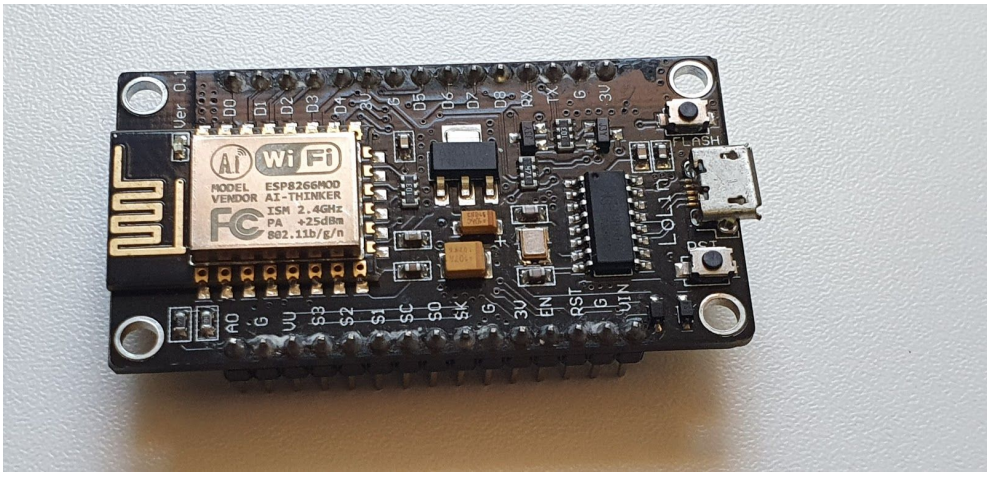
3. Resultados e Discussões

3.1 O Protótipo

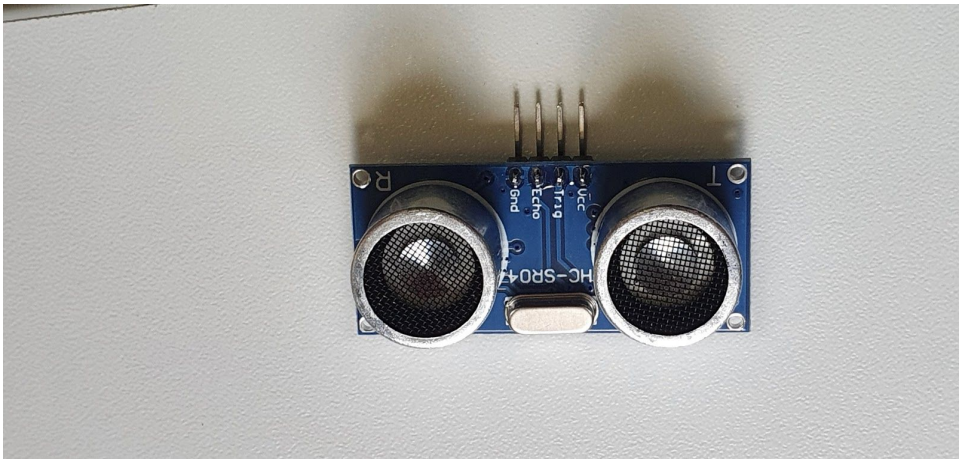
Fotografia 1- Início da montagem.



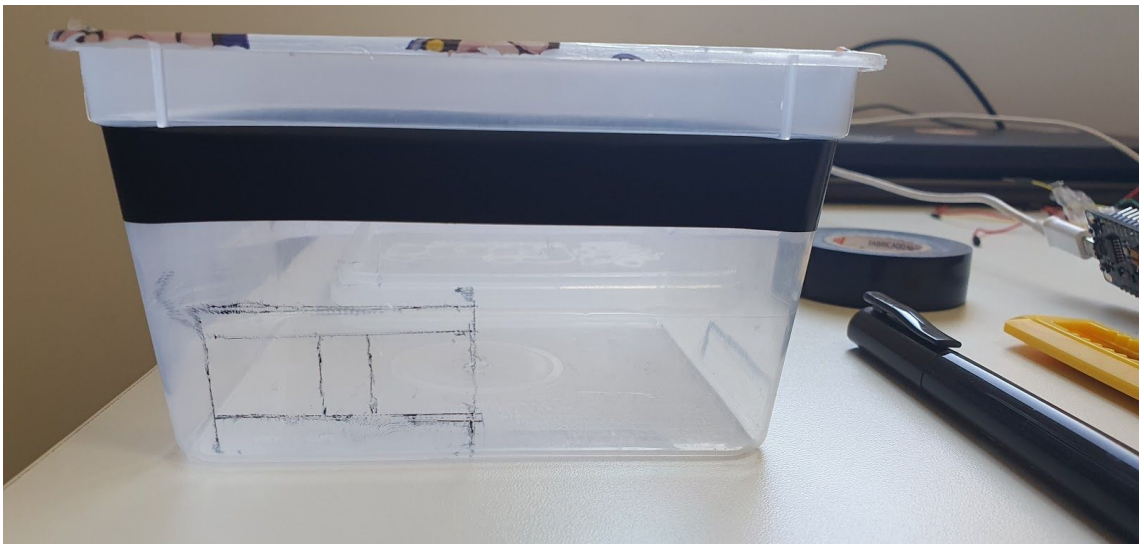
Fotografia 2 - Placa ESP8266.



Fotografia 3 - Sensor Ultra-sônico.

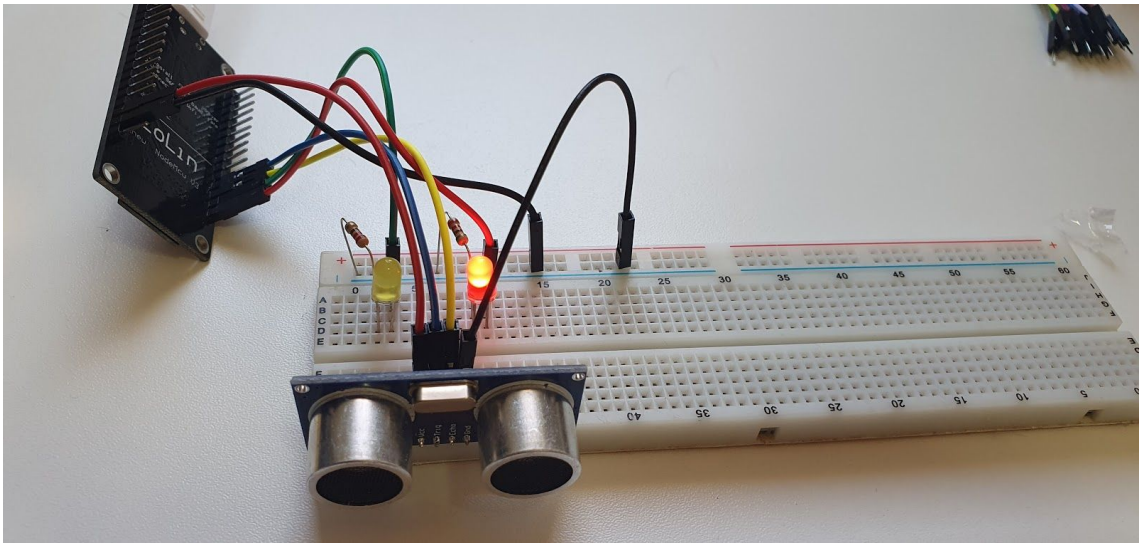


Fotografia 4 - Medidas na caixa para o protótipo.

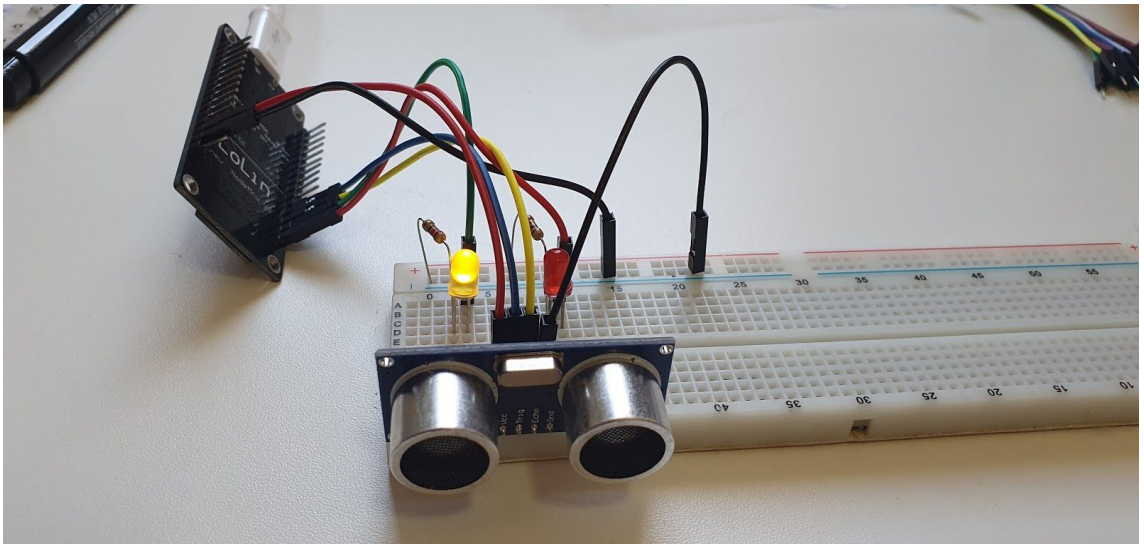


Fotografia 5 - Encapamento da caixa .

Fotografia 6 - Teste de leds com o servidor (vermelho desconectado).

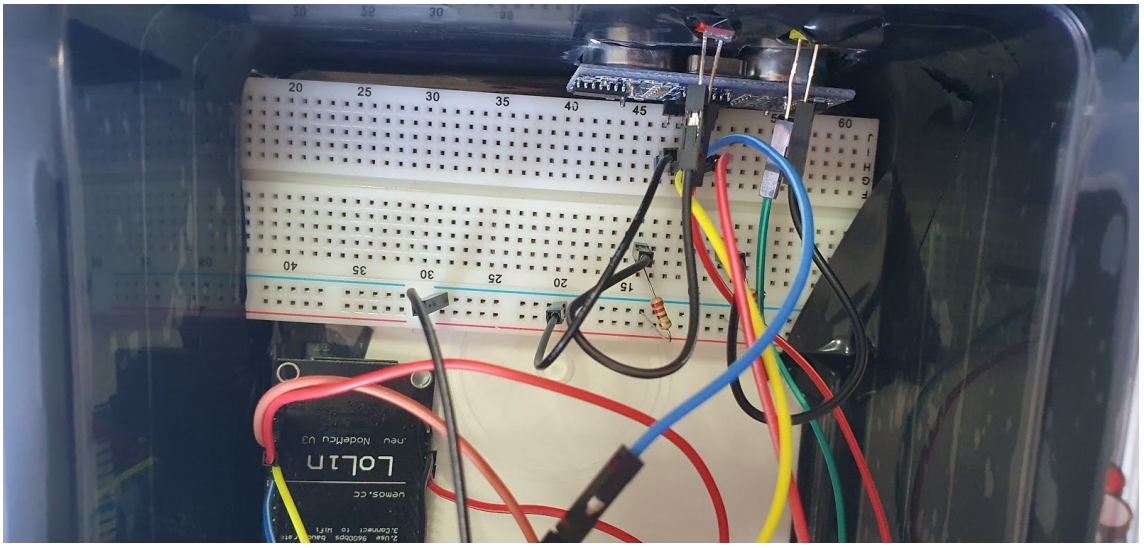


Fotografia 7 - Teste de leds com o servidor (verde conectado).

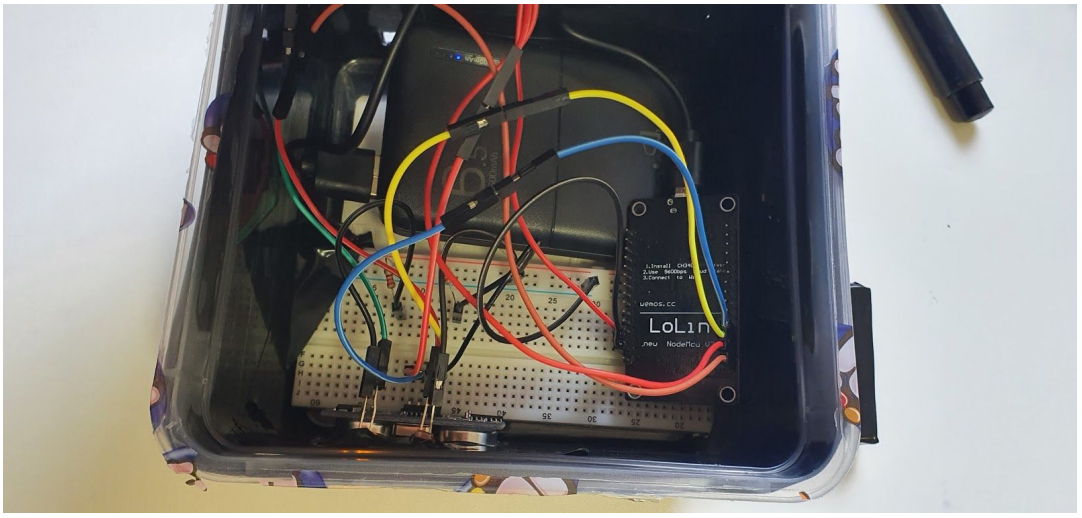


Fotografia 8 - Caixa protetora do protótipo encapada.

Fotografia 9 - Componentes conectados e colocados na caixa.



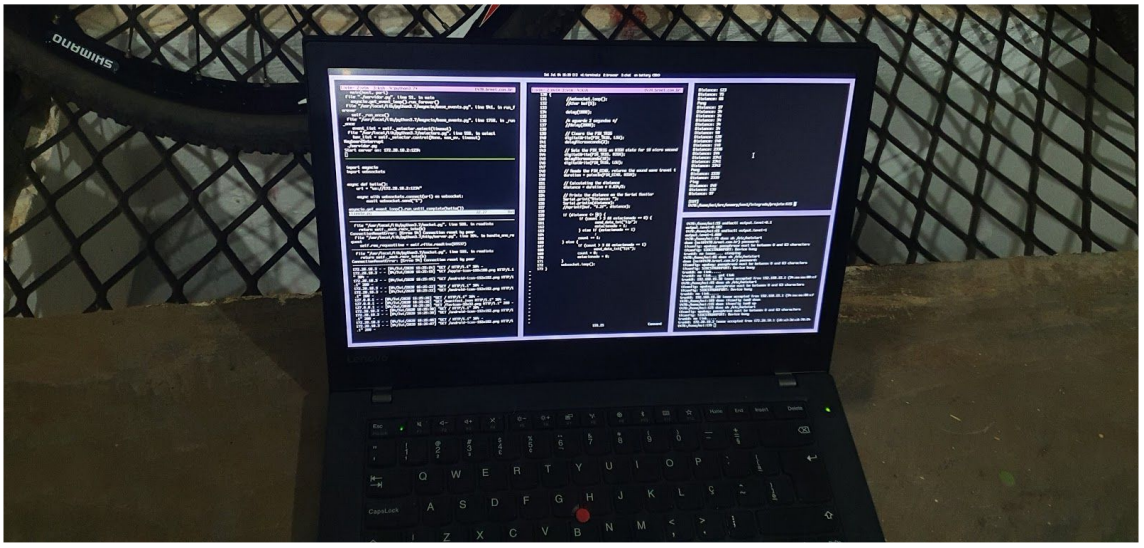
Fotografia 10 - Componentes conectados a energia (bateria externa).



Fotografia 11 - Protótipo pronto (Desconectado do servidor).

Fotografia 12 - Protótipo pronto (Conectado ao servidor).

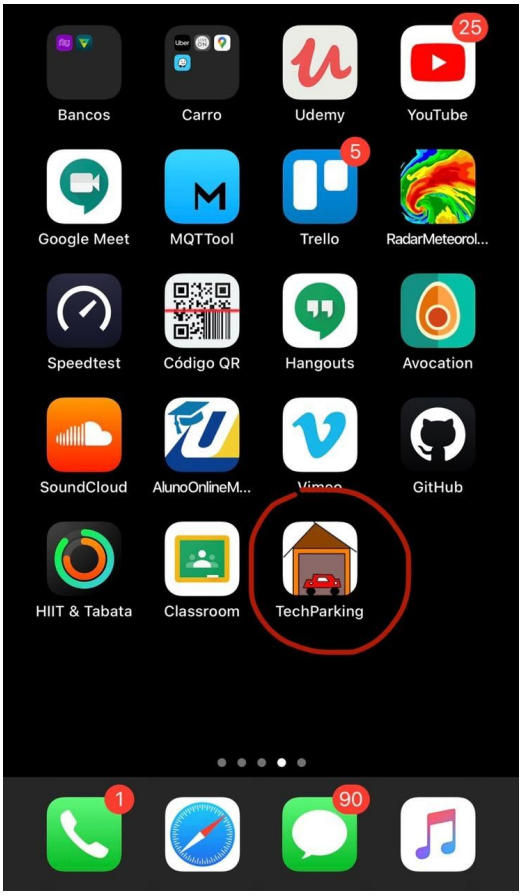
Fotografia 13 - Computador com monitoramento durante os testes.



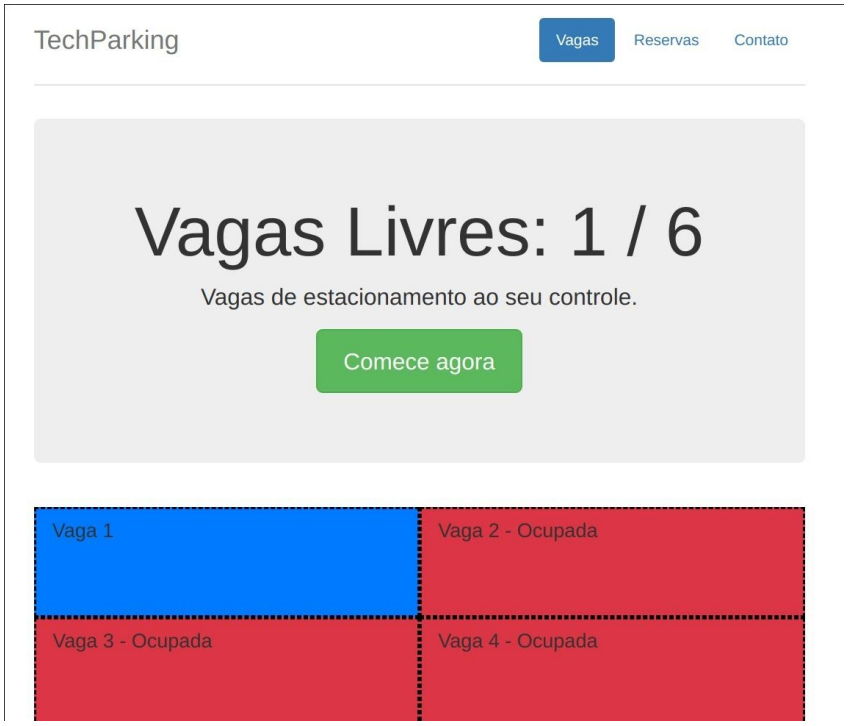
Fotografia 14 - Protótipo pronto e Posicionado para o teste final (Monitoramento).

Fotografia 15 - Protótipo pronto e Posicionado para o teste final (Ambiente residencial).

Fotografia 16 - Ícone da página Web no Smartfone.



Fotografia 17 - Interface da página Web no Smartfone



Fonte de Todas as Imagens: Elaborada pelo autor.

Códigos Fontes

Código Hardware ESP8266

```
/*

    * Copyright (c) 2020 Murilo Ijanc' <mbsd@m0x.ru>
    *
    * Permission to use, copy, modify, and distribute this
    software for any
    * purpose with or without fee is hereby granted,
    provided that the above
    * copyright notice and this permission notice appear in
    all copies.
    *
    * THE SOFTWARE IS PROVIDED "AS IS" AND THE AUTHOR
    DISCLAIMS ALL WARRANTIES
    * WITH REGARD TO THIS SOFTWARE INCLUDING ALL IMPLIED
    WARRANTIES OF
    * MERCHANTABILITY AND FITNESS. IN NO EVENT SHALL THE
    AUTHOR BE LIABLE FOR
    * ANY SPECIAL, DIRECT, INDIRECT, OR CONSEQUENTIAL
    DAMAGES OR ANY DAMAGES
    * WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR
    PROFITS, WHETHER IN AN
    * ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS
    ACTION, ARISING OUT OF
    * OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS
    SOFTWARE.
    */


    #include <ESP8266WiFi.h>
```

```
#include <Hash.h>
#include <WebSocketsClient.h>

/* wifi ssid e senha */
const char     *wifi_ssid = "iPhone";
const char     *wifi_password = "1234567890";

/* leds */
const int  PIN_VERM = 16;          /* D0 */
const int  PIN_AMAR = 5;       /* D1 */

/* Sensor ultrasônico */
const int  PIN_ECHO = 4;           /* D2 */
const int  PIN_TRIG = 0;           /* D3 */

/* BaudRate 1152000 */
const int       BRD = 115200;

/* configurações do servidor websocket */
const char     *ip_servidor = "172.20.10.2";
const char     *rota = "/";
const int  porta = 1234;

/* instância global websocket */
WebSocketsClient websocket;
```

```
long duration;
int distance;
int count = 0;
int estacionado = 0;

static void      fn_recebe_evento(WStype_t,  uint8_t  *,
size_t);
static void     send_data_txt(const char *);

/*
* Função Setup
*/
void
setup(void)
{
     /* inicializo o serial */
     Serial.begin(BRD);

     /* inicializamos os leds */
     pinMode(PIN_VERM, OUTPUT);
     pinMode(PIN_AMAR, OUTPUT);

     /* inicializamos o sensor ultrasônico */
     pinMode(PIN_ECHO, INPUT);
     pinMode(PIN_TRIG, OUTPUT);
```

```
/* inicializar wifi modo estação (cliente) */
WiFi.mode(WIFI_STA);
WiFi.begin(wifi_ssid, wifi_password);

while (WiFi.status() != WL_CONNECTED) {
   digitalWrite(PIN_VERM, HIGH);
   delay(500);
   digitalWrite(PIN_VERM, LOW);
   Serial.println("Não conectado.");
}
digitalWrite(PIN_VERM, LOW);
digitalWrite(PIN_AMAR, HIGH);

/* imprime o ssid da rede wifi */
Serial.print("Conectado Wifi: ");
Serial.println(wifi_ssid);
Serial.println("");

/* retorna o ip da placa */
Serial.print("IP: ");
Serial.print(WiFi.localIP());

/* conectar ao servidor websocket */;
websocket.begin(ip_servidor, porta, rota);
/* qualquer evento */
websocket.onEvent(fn_recebe_evento);
/* intervalo de reconexão */
```

```
    websocket.setReconnectInterval(5000);
    /* heart beat */
    websocket.enableHeartbeat(15000, 3000, 2);
}


static void
send_data_txt(const char *msg)
{
    websocket.sendTXT(msg);
}


static void
fn_recebe_evento(WStype_t  tipo,  uint8_t  *dado,  size_t
tamanho)
{
    switch(tipo) {
        case WStype_DISCONNECTED:
          Serial.println("Desconectado Websocket");
          break;
        case WStype_CONNECTED:
          Serial.println("Conectado no Websocket");
          break;
        case WStype_PING:
          Serial.println("Ping");
          break;
        case WStype_PONG:
          Serial.println("Pong");
          break;
```

```
	}
}


/*
* Loop principal
*/
void
loop(void)
{
	//websocket.loop();
	//char buf[5];

	delay(1000);

	/* aguarde 2 segundos */
	//delay(2000);

	// Clears the PIN_TRIG
	digitalWrite(PIN_TRIG, LOW);
	delayMicroseconds(2);

	// Sets the PIN_TRIG on HIGH state for 10 micro seconds
	digitalWrite(PIN_TRIG, HIGH);
	delayMicroseconds(10);
	digitalWrite(PIN_TRIG, LOW);
```

```
      // Reads the PIN_ECHO, returns the sound wave
travel time in microseconds
      duration = pulseIn(PIN_ECHO, HIGH);


      // Calculating the distance
      distance = duration * 0.034/2;


      // Prints the distance on the Serial Monitor
      Serial.print("Distance: ");
      Serial.println(distance);
      //sprintf(buf, "%.2f", distance);


      if (distance <= 40) {
         if (count > 3 && estacionado == 0) {
           send_data_txt("1|p");
           estacionado = 1;
         } else if (estacionado == 1)
           ;
         count += 1;
      } else {
         if (count > 3 && estacionado == 1)
           send_data_txt("1|l");
         count = 0;
         estacionado = 0;
      }
      websocket.loop();}
```

Código do Servidor em python

```
#!/usr/bin/e
nv python3
                #
                # Copyright (c) 2020 Murilo Ijanc'
                <mbsd@m0x.ru>
                #
                # Permission to use, copy, modify, and
                distribute this software for any
                # purpose with or without fee is hereby
                granted, provided that the above
                # copyright notice and this permission notice
                appear in all copies.
                #
                # THE SOFTWARE IS PROVIDED "AS IS" AND THE
                AUTHOR DISCLAIMS ALL WARRANTIES
                # WITH REGARD TO THIS SOFTWARE INCLUDING ALL
                IMPLIED WARRANTIES OF
                # MERCHANTABILITY AND FITNESS. IN NO EVENT
                SHALL THE AUTHOR BE LIABLE FOR
                # ANY SPECIAL, DIRECT, INDIRECT, OR
                CONSEQUENTIAL DAMAGES OR ANY DAMAGES
                # WHATSOEVER RESULTING FROM LOSS OF USE, DATA
                OR PROFITS, WHETHER IN AN
                # ACTION OF CONTRACT, NEGLIGENCE OR OTHER
                TORTIOUS ACTION, ARISING OUT OF
                # OR IN CONNECTION WITH THE USE OR
                PERFORMANCE OF THIS SOFTWARE.


                import os
                import asyncio
```

```
import websockets

USUARIOS = set()


async def register(websocket):
    USUARIOS.add(websocket)


async def unregister(websocket):
    USUARIOS.remove(websocket)


async def notificar_usuarios(msg, websocket):
    if USUARIOS:
        await asyncio.wait([u.send(msg) for u
in USUARIOS if u != websocket])


async def hello(websocket, path):
    await register(websocket)

    try:
        async for msg in websocket:
```

```
                await notificar_usuarios(msg,
websocket)
    finally:
        await unregister(websocket)



def main(host, port):
      start_server = websockets.serve(hello,
host, port)

asyncio.get_event_loop().run_until_complete(s
tart_server)
     print("Start server on: %s:%d" % (host,
port))
    asyncio.get_event_loop().run_forever()



if __name__ == '__main__':
           host    =    os.environ.get("HOST",
"192.168.15.38")
    port = int(os.environ.get("PORT", 1234))


    main(host, port)
```

Código do Website

```
#!/usr/bin/en
v python3
#
#   Copyright   (c)   2020   Murilo   Ijanc'
<mbsd@m0x.ru>
#
#  Permission  to  use,  copy,  modify,  and
distribute this software for any
#  purpose  with  or  without  fee  is  hereby
granted, provided that the above
# copyright notice and this permission notice
appear in all copies.
#
# THE SOFTWARE IS PROVIDED "AS IS" AND THE
AUTHOR DISCLAIMS ALL WARRANTIES
# WITH REGARD TO THIS SOFTWARE INCLUDING ALL
IMPLIED WARRANTIES OF
#  MERCHANTABILITY AND FITNESS. IN NO EVENT
SHALL THE AUTHOR BE LIABLE FOR
#   ANY   SPECIAL,   DIRECT,   INDIRECT,   OR
CONSEQUENTIAL DAMAGES OR ANY DAMAGES
# WHATSOEVER RESULTING FROM LOSS OF USE, DATA
OR PROFITS, WHETHER IN AN
#  ACTION  OF  CONTRACT,  NEGLIGENCE  OR  OTHER
TORTIOUS ACTION, ARISING OUT OF
#   OR   IN   CONNECTION   WITH   THE   USE   OR
PERFORMANCE OF THIS SOFTWARE.


import asyncio
import websockets
```

```
async def hello():
    uri = "ws://172.20.10.2:1234"

        async with websockets.connect(uri) as
    websocket:
        await websocket.send("1")

    asyncio.get_event_loop().run_until_complete(h
    ello())
```

```
<!DOCT
YPE
html>
<html lang="en">
 <head>
   <meta charset="utf-8">
        <meta http-equiv="X-UA-Compatible"
content="IE=edge">
                <meta name="viewport"
content="width=device-width, initial-scale=1">
   <meta name="description" content="">
   <meta name="author" content="">

    <link rel="apple-touch-icon" sizes="57x57"
href="/apple-icon-57x57.png">
<link rel="apple-touch-icon" sizes="60x60"
href="/apple-icon-60x60.png">
```

```
<link      rel="apple-touch-icon"      sizes="72x72"
href="/apple-icon-72x72.png">
<link      rel="apple-touch-icon"      sizes="76x76"
href="/apple-icon-76x76.png">
<link     rel="apple-touch-icon"     sizes="114x114"
href="/apple-icon-114x114.png">
<link     rel="apple-touch-icon"     sizes="120x120"
href="/apple-icon-120x120.png">
<link     rel="apple-touch-icon"     sizes="144x144"
href="/apple-icon-144x144.png">
<link     rel="apple-touch-icon"     sizes="152x152"
href="/apple-icon-152x152.png">
<link     rel="apple-touch-icon"     sizes="180x180"
href="/apple-icon-180x180.png">
<link  rel="icon"  type="image/png"  sizes="192x192"
href="/android-icon-192x192.png">
<link   rel="icon"   type="image/png"   sizes="32x32"
href="/favicon-32x32.png">
<link   rel="icon"   type="image/png"   sizes="96x96"
href="/favicon-96x96.png">
<link   rel="icon"   type="image/png"   sizes="16x16"
href="/favicon-16x16.png">
<link rel="manifest" href="/manifest.json">
<meta                name="msapplication-TileColor"
content="#ffffff">
<meta                name="msapplication-TileImage"
content="/ms-icon-144x144.png">
<meta name="theme-color" content="#ffffff">


   <title>TechParking</title>


   <!-- Bootstrap core CSS -->
```

```
                                                    <link
href="https://getbootstrap.com/docs/3.3/dist/css/bo
otstrap.min.css" rel="stylesheet">


     <style>
        /* Space out content a bit */
body {
 padding-top: 20px;
 padding-bottom: 20px;
}


/* Everything but the jumbotron gets side spacing
for mobile first views */
.header,
.marketing,
.footer {
 padding-right: 15px;
 padding-left: 15px;
}


/* Custom page header */
.header {
 padding-bottom: 20px;
 border-bottom: 1px solid #e5e5e5;
}
/* Make the masthead heading the same height as the
navigation */
.header h3 {
```

```
 margin-top: 0;
 margin-bottom: 0;
 line-height: 40px;
}

/* Custom page footer */
.footer {
 padding-top: 19px;
 color: #777;
 border-top: 1px solid #e5e5e5;
}

/* Customize container */
@media (min-width: 768px) {
 .container {
   max-width: 730px;
 }
}
.container-narrow > hr {
 margin: 30px 0;
}

/* Main marketing message and sign up button */
.jumbotron {
 text-align: center;
 border-bottom: 1px solid #e5e5e5;
}
```

```
.jumbotron .btn {
 padding: 14px 24px;
 font-size: 21px;
}

/* Supporting marketing content */
.marketing {
 margin: 40px 0;
}
.marketing p + h4 {
 margin-top: 28px;
}

/* Responsive: Portrait tablets and up */
@media screen and (min-width: 768px) {
 /* Remove the padding we set earlier */
 .header,
 .marketing,
 .footer {
   padding-right: 0;
   padding-left: 0;
 }
 /* Space out the masthead */
 .header {
   margin-bottom: 30px;
 }
  /* Remove the bottom border on the jumbotron for
visual effect */
```

```
 .jumbotron {
   border-bottom: 0;
 }
}
          .estacionamento {
               width: 100%;
          }
          .vaga {
               min-height: 100px;
               background-color: #007bff;
               border-color: black;
               border-style: dashed;
          }
          .preenchida {
               background-color: #dc3545;
          }
     </style>
 </head>


 <body>


   <div class="container">
     <div class="header clearfix">
       <nav>
         <ul class="nav nav-pills pull-right">
                              <li   role="presentation"
class="active"><a href="#">Vagas</a></li>
```

```
                        <li  role="presentation"><a
href="#">Reservas</a></li>
                        <li  role="presentation"><a
href="#">Contato</a></li>
          </ul>
        </nav>
        <h3 class="text-muted">TechParking</h3>
      </div>


      <div class="jumbotron">
        <h1 id="info-vagas">Vagas Livres: 1 / 6</h1>
          <p class="lead">Vagas de estacionamento ao
seu controle.</p>
             <p><a  class="btn  btn-lg  btn-success"
href="#" role="button">Comece agora</a></p>
      </div>

      <div class="row marketing">
        <div class="col-lg-6 vaga" id="1">
          <h4>Vaga 1</h4>
        </div>


            <div  class="col-lg-6  vaga  preenchida"
id="2">
          <h4>Vaga 2 - <span>Ocupada</span></h4>
        </div>
            <div  class="col-lg-6  vaga  preenchida"
id="3">
          <h4>Vaga 3 - <span>Ocupada</span></h4>
```

```
        </div>

            <div  class="col-lg-6  vaga  preenchida"
id="4">
          <h4>Vaga 4 - <span>Ocupada</span></h4>
        </div>
            <div  class="col-lg-6  vaga  preenchida"
id="5">
          <h4>Vaga 5 - <span>Ocupada</span></h4>
        </div>

            <div  class="col-lg-6  vaga  preenchida"
id="6">
          <h4>Vaga 6 - <span>Ocupada</span></h4>
        </div>
      </div>

      <footer class="footer">
           <p>&copy; 2020 TechParking Eng. Software,
Inc.</p>
      </footer>

    </div>
```

```
<!--

        <div class="estacionamento">
          <div class="vaga" id="1">01</div>
          <div        class="vaga        preenchida"
id="2">02</div>
        </div>
-->
       <script>
          /* host do servidor websocket */
          const host = "ws://172.20.10.2:1234/";
           const ws = new WebSocket(host);
          var entrou = 0;


          /*
           * Função altera a cor da div
           */
          var altera_vaga = (id_vaga) => {
               /* obtenho o elemento vaga pelo id*/
               const        class_preenchida        =
"preenchida";
               const            el_vaga            =
document.getElementById(id_vaga);
               if  (el_vaga  !=  undefined  &&  el_vaga
!= null) {

el_vaga.classList.toggle(class_preenchida);
               } else {
                    console.log("Não   encontrado   o
elemento de id: ", id_vaga);
```

```
                }
                if (entrou === 0) {

document.getElementById("info-vagas").innerHTML="Va
gas Livres: 2 / 6";
                    entrou = 1;
                } else {

document.getElementById("info-vagas").innerHTML="Va
gas Livres: 1 / 6";
                    entrou = 0;
                }
            };


            ws.onmessage = function (event) {
                const data = (event.data);
                console.log("Recebido: ", data);
                const _id = data.split("|");
                altera_vaga(data[0]);
            };
        </script>
    </body>
</html>
```

3.2 **Projeto Integrador**

Tendo em vista à realização do projeto Integrador, foi usado como base da confecção do projeto: As estruturas de código aprendidas na matéria de “Laboratório de Programação I”; como lidar e programar em cima de um microcontrolador sem sistema operacional, analisado na matéria de “Lógica e Criatividade”; a documentação dos requisitos do projeto, aprendido na matéria de “Engenharia de Software”; a organização e confecção textual, abordado na matéria de “Práticas de Leitura, Interpretação e Produção de Textos” e o design e estruturação da interface presente na página web, que tivemos como base de atuação os conceitos examinados na matéria de “Interação Humano Computador”.

4. **Conclusão**

Tendo em vista o objetivo inicial do trabalho, que consiste em identificar a disponibilidade de vagas em um determinado estacionamento e suas especificidades, pode-se observar um satisfatório retornos dos esforços dispostos neste projeto. No que se refere ao resultado geral do produto, tem-se que as funcionalidades estão atuando de maneira correta, o tempo de resposta entre a ocupação de uma vaga e o preenchimento da mesma na página web, apresenta um desempenho rápido e responsivo, algo que excedeu as expectativas definidas para esse parâmetro. As limitações identificadas neste projeto consistem em: a quantidade de material disponibilizada para a confecção de um maior número de dispositivos e a dificuldade de encontros presenciais para a construção do mecanismo, devido a atual política de distanciamento social. No que diz respeito às melhorias que poderiam ser implementadas em versões futuras do dispositivo atual, temos que conseguiriam ser realizados aperfeiçoamentos na interface da página web, poderia ser realizado a extensão dessa função à aplicativos destinados para dispositivos móveis e uma maior compactação do protótipo físico, utilizando materiais mais resistentes, dispor de uma maior organização dos componentes integrados (PCB), e em relação a segurança seria necessário a implementação de uma autenticação para o acesso ao gerenciamento da placa.

5. Referências

<https://www.arduino.cc/>

HTML

<https://developer.mozilla.org/en-US/docs/Web/HTML>

Websocket

<https://developer.mozilla.org/en-US/docs/Web/API/WebSockets_API>

Datasheet ESP8266

<https://www.espressif.com/sites/default/files/documentation/0a-esp8266ex_datasheet_en.pdf>

Datasheet Sensor ultrassônico
<https://www.filipeflop.com/blog/sensor-ultrassonico-hc-sr04-ao-arduino/>

Datasheet Leds

<https://category.alldatasheet.com/index.jsp?sSearchword=Led%20datasheet&gclid=EAIaIQobChMI3vfAuJa16gIVCA-RCh03wwRhEAAYASAAEgK-K_D_BwE>

CONTRIBUIÇÕES DOS ALUNOS

Murilo Ijanc - 834125 : Codificação e montagem do protótipo.

João Victor Tiezzi de Jesus - 834376 : Montagem do protótipo e documentação.

Ranniery Lucas de Carvalho Moreira - 834761 : Documentaçao.

Miguel Silvério - 833895 : Apresentação.