

**INSTITUTO FEDERAL DE EDUCAÇÃO E TECNOLOGIA DE MINAS  
GERAIS *CAMPUS* FORMIGA  
Técnico Integrado em Informática**

**HENE KYANNE BUENO DE OLIVEIRA**

**SMART PARKING  
PROTÓTIPO DE ESTACIONAMENTO INTELIGENTE**

Formiga MG  
2018

HENE KYANNE BUENO DE OLIVEIRA

SMART PARKING  
PROTÓTIPO DE ESTACIONAMENTO INTELIGENTE

Trabalho de conclusão de curso apresentado à Instituição Federal de educação e tecnologia de Minas Gerais Campus Formiga IFMG como sendo um requisito para que se possa obter o diploma de Técnico em informática.

Orientador: Prof. Dr. Otávio de Souza Martins Gomes.

# AGRADECIMENTOS

Agradeço primeiramente a Deus por estar presente em minha vida tornando todos meus sonhos possíveis. Ao meu orientador Otávio Gomes, por sua dedicação e prontidão para orientar e ajudar a concluir o projeto.

Agradeço aos meus familiares, principalmente a minha mãe por me ajudar e apoiar, se fazendo presente em todos momentos de minha vida. Ao meu namorado Gabriel por estar ao meu lado me apoiando em todas as decisões.

A toda instituição, principalmente a equipe de professores pelos conhecimentos transferidos ao longo desses 4 anos de curso.

Aos meus amigos que estiveram comigo durante o curso, ajudando nesta caminhada nos momentos bons e ruins.

## RESUMO

Este trabalho descreve a implementação de um simulador de um protótipo de estacionamento inteligente, tanto para o setor público, quanto para o privado. Por meio dele é possível simular o funcionamento de um estacionamento automatizado que visa auxiliar os condutores na procura por vagas nos estabelecimentos. Com intuito de melhorar esse cenário, desenvolveu-se uma simulação que gerencia e faz o monitoramento das vagas por meio das plataformas Arduino, *Processing* e *AppInventor*.

**Palavras chaves:** Protótipo de estacionamento Inteligente. Arduino. *Processing*.

## LISTA DE FIGURAS

Figura 1 Exemplo de estacionamento nas ruas .....	5
Figura 2 Representação das luzes do sistema adotado pelo MorumbiShopping.....	6
Figura 3 Ilustração dos Painéis do MorumbiShopping. ....	6
Figura 4 Estacionamento subterrâneo de bicicletas. ....	7
Figura 5 Estacionamento subterrâneo de automóveis. ....	7
Figura 6 Placa Arduino Uno. ....	8
Figura 7 Representação dos LEDs. ....	9
Figura 8 Cálculo para medir a resistência de um LED.....	10
Figura 9 Exemplo da diferença entre os resistores. ....	11
Figura 10 Representação do LDR. ....	11
Figura 11 Micro Servo 9g TowerPro.....	12
Figura 12 Display LCD 16x2. ....	13
Figura 13 Protoboard. ....	14
Figura 14 Ilustração do potenciômetro. ....	14
Figura 15 Módulo Bluetooth HC-06. ....	16
Figura 16 Logo da ferramenta TinkerCad. ....	16
Figura 17 Logo da ferramenta AppInventor.....	17
Figura 18 Metodologia .....	19
Figura 19 Certificado do curso Aprenda Arduino com uso de simulador. ....	21
Figura 20 Certificado do curso Arduino- Curso completo e prático.....	21
Figura 21 Maquete do estacionamento (visão superior). ....	24
Figura 22 Maquete do estacionamento (visão lateral).....	24
Figura 23 Ilustração dos leds que representam uma vaga. ....	25
Figura 24 Teste com o módulo Bluetooth. ....	25
Figura 25 Circuito eletrônico do estacionamento.....	26
Figura 26 Circuito eletrônico em funcionamento.....	27
Figura 27 Permissão para ativar o Bluetooth. ....	28
Figura 28 Bluetooth do smartphone ativado. ....	29
Figura 29 Lista dos dispositivos com bluetooth ativado. ....	30
Figura 30 Aplicativo conectado ao módulo. ....	30
Figura 31 Circuito do módulo Bluetooth. ....	31
Figura 32 Mensagem de erro. ....	32
Figura 33 Ilustração das vagas ocupadas.....	32
Figura 34 Codificação das vagas. ....	33
Figura 35 Ilustração das vagas desabilitadas.....	34
Figura 36 Ilustração das vagas habilitadas. ....	35
Figura 37 Vagas ocupadas.....	36

## **LISTA DE TABELAS**

Tabela 1 Características da placa Arduino. ....	8
Tabela 2 Descrição dos componentes do circuito. ....	18

# LISTA DE ABREVEATURAS E SIGLAS

ABNT - Associação Brasileira de Normas Técnicas

GND – *Ground*

IBGE - Instituto Brasileiro de Geografia e Estatística

IDE - *Integrated Development Environment*

LCD - *Liquid Crystal Display*"

LDR - *Light Dependent Resistor*

LED - *Light Emitting Diode*

MIT- *Massachusetts Institute of Technology*

# SUMÁRIO

<b>1 INTRODUÇÃO</b> .....	1
1.1 Objetivos Gerais .....	2
1.2 Objetivos Específicos.....	2
1.2 Motivação e Justificativa.....	3
<b>2 REFERENCIAL TEÓRICO</b> .....	4
2.1 Sistemas de estacionamentos .....	4
2.2 Arduino .....	7
2.2.1 LEDs .....	9
2.2.2 Resistor.....	9
2.2.3 LDR.....	11
2.2.4 Servo Motor.....	12
2.2.5 <i>Display LCD</i> .....	13
2.2.6 <i>Protoboard</i> .....	13
2.2.7 Potenciômetro .....	14
2.3 <i>Bluetooth</i> .....	15
2.3.1 Módulo de Conexão <i>Bluetooth</i> HC- 06 .....	15
2.4 <i>TinkerCAD</i> .....	16
2.5 <i>AppInventor</i> .....	16
2.6 <i>Processing</i> .....	17
<b>3 MATERIAIS E MÉTODOS</b> .....	18
3.1 Materiais .....	18
3.2 Metodologia .....	19
3.2.1 Definição do Escopo .....	20
3.2.2 Estudo das ferramentas e tecnologias envolvidas .....	20
3.2.3 Implementação.....	22
<b>4 DESENVOLVIMENTO</b> .....	23
4.1 Maquete 3D.....	23
4.2 Circuito Eletrônico .....	24



4.3 Construção do Aplicativo Móvel .....	27
4.4 Simulação no <i>Processing</i> .....	34
4.5 Problemas Enfrentados.....	36
<b>5 RESULTADOS E DISCUSSÕES</b> .....	<b>37</b>
<b>6 TRABALHOS FUTUROS</b> .....	<b>38</b>
<b>7 CONSIDERAÇÕES FINAIS</b> .....	<b>39</b>
<b>REFERÊNCIAS BIBLIOGRÁFICAS</b> .....	<b>41</b>
<b>APÊNDICES</b> .....	<b>47</b>
APÊNDICE A - Código do circuito eletrônico do estacionamento.....	47
APÊNDICE B - Código da conexão via Bluetooth.....	51
APÊNDICE C -Desenho esquemático da maquete 3D com as respectivas medidas..	53

# 1 INTRODUÇÃO

O fluxo de veículos automotores vem aumentando gradativamente à medida em que a população cresce. Pesquisas realizadas (IBGE 2013), mostram que no Brasil há cerca de um automóvel para quatro habitantes, ou seja, mais de 50 milhões de veículos. Embora já existam inúmeras vagas de estacionamentos - públicos ou privados- essas ofertas não estão suprindo a demanda que os motoristas precisam, tendo em vista a necessidade de locomoção e a quantidade de automóveis transitando pelas ruas. Por conseguinte, eles acabam encontrando diversas dificuldades relacionadas a disponibilidade de vagas, o que interfere diretamente no trânsito.

Ademais, em decorrência desse impasse, de acordo com o site do Jornal Nacional (G1, 2011) os condutores que residem nos grandes centros urbanos, na maioria das vezes, são obrigados a deixarem os automóveis nas ruas ou pagar pelo estacionamento. Analisando esses aspectos, nota-se que há espaço para inovação na elaboração de um protótipo que auxilie o controle de vagas nos estacionamentos, para que haja uma maior organização no fluxo dos veículos, bem como, a precaução dos riscos em que o proprietário corre ao deixar seu veículo na rua.

Visando amenizar esses problemas relacionados a disponibilidade de vagas, propomos um modelo de estacionamento que irá atender da melhor forma possível os motoristas que sofram com a presente situação de não encontrar vagas nos estabelecimentos. Embora essa versão do protótipo não garanta a implantação final no mercado, ela oferece subsídio para trabalhos futuros que visem permitir aos condutores maior praticidade para encontrar vagas, devido à presença de sensores que indicam a disponibilidade da vaga.

Atualmente, existem alguns modelos de produtos semelhantes ao do protótipo que visam a praticidade no cotidiano dos condutores e que já estão em funcionamento no mercado. Um exemplo disso é o sistema robotizado de estacionamento na China que, em vez de do cliente sair para procurar vagas, é necessário somente colocar o carro sobre o elevador, que se encarrega de fazer o serviço (AGÊNCIA EFE, 2016). Além desse exemplo, existem outros modelos presentes em *shoppings* e prédios que

possuem o mesmo mecanismo do *Smart Parking*, que contam com a presença de telões e *LEDs* para auxiliar na visibilidade das vagas disponíveis (AUTOMSOLUTION, s.d).

Primeiramente, este projeto apresenta a conceituação dos componentes que foram utilizados, bem como, suas características e abrangência. Em seguida, aborda-se o processo de utilização dos elementos na placa virtual, com foco nos referenciais teóricos. Assim, a partir dos estudos sobre os componentes da placa Arduino, aprendizagem da linguagem C++, funcionamento do *Processing* e pesquisas voltadas à trabalhos similares já existentes, pode-se inferir que, com as análises e estudos, a implantação do projeto dará embasamento a outras versões que poderão amenizar impasses relacionados à indisponibilidade das vagas.

## 1.1 Objetivos Gerais

Esse trabalho tem como objetivo geral o desenvolvimento de um simulador para um protótipo de estacionamento inteligente, controlado pela placa Arduino Uno e sensores. Será realizado o gerenciamento e monitoramento do estacionamento através de um aplicativo conectado à uma simulação realizada na ferramenta *Processing*, a qual indica se a vaga estará livre ou ocupada.

## 1.2 Objetivos Específicos

Para cumprir os objetivos gerais do projeto, propõem-se os seguintes objetivos específicos:

- Pesquisar ferramentas e tecnologias para satisfazer as necessidades do projeto;
- Aprender sobre o funcionamento das ferramentas para o desenvolvimento do circuito;
- Implementar o circuito eletrônico;
- Implementar o aplicativo móvel;
- Implementar a simulação do estacionamento.

## 1.2 Motivação e Justificativa

A motivação desse trabalho baseia-se na observação da situação do trânsito de Formiga-MG, bem como as dificuldades enfrentadas pelos motoristas da cidade. Devido ao fato de que as ruas são estreitas, os condutores encontram diversas dificuldades no que diz respeito a encontrarem vagas nas áreas próximas ao centro e também a geração de tumulto no trânsito, decorrente da superlotação dos estacionamentos, que dificulta a passagem dos automóveis.

Além disso, o projeto é de autoaprendizagem para o aluno, uma vez que o discente estudou e realizou muitas pesquisas para compreender o assunto e entender as ferramentas que foram utilizadas no decorrer do desenvolvimento do trabalho. Dessa forma, ele estará apto para realizar outros projetos futuros que possam vir a necessitar dos mecanismos que foram utilizados, o que irá contribuir diretamente em sua formação acadêmica e pessoal.

## 2 REFERENCIAL TEÓRICO

Nesta seção serão abordados, de forma resumida, a revisão da literatura a respeito das ferramentas que foram utilizadas no projeto, bem como, a apresentação de sistemas similares que já foram implantados no mercado, com intuito de facilitar o entendimento do leitor sobre a elaboração do trabalho.

### 2.1 Sistemas de estacionamentos

Um ponto importante é a introdução de sistemas de estacionamentos já existentes no mercado. Atualmente, no Brasil, existem sistemas que gerenciam as vagas dos estabelecimentos, principalmente nas grandes cidades onde o fluxo de automóveis é maior. Consequentemente, a superlotação das cidades, problemas relacionados ao transporte público e a grande necessidade de locomoção geram uma urgência no aumento de vagas e também no aperfeiçoamento das já existentes (SILVA e et al., s.d).

A cidade Águas de São Pedro no interior de São Paulo já implantou o sistema nas ruas, o qual é semelhante ao *Smart Parking*. Ele utiliza o mesmo mecanismo dos estacionamentos de *Shopping* com luzes indicadoras verdes e vermelhas espalhadas pelo chão para que os motoristas tenham ampla visão das vagas que estão disponíveis (LEITE, 2017). Além disso, o sistema é alimentado por placas solares, e são facilmente instalados por não precisar de fiação e, através de um aplicativo, o condutor verifica quais vagas estão disponíveis na rua e é levado à mais próxima, poupando seu tempo (MEIRELLES, 2014). A Figura 1 representa o estacionamento presente nas ruas da cidade Águas de São Pedro.



Figura 1 Exemplo de estacionamento nas ruas

Fonte: Joel Leite (2017).

O MorumbiShopping localizado na cidade de São Paulo foi um dos primeiros estabelecimentos do Brasil a adotar um sistema de estacionamento que utiliza sensores e luzes de *Led*, espelhando-se nos sistemas já existentes na Europa e Estados Unidos. O sistema possui o mesmo princípio do *Smart Parking* visto que, no posicionamento do automóvel sobre a vaga, o sensor detecta sua presença e automaticamente acende a luz vermelha indicando indisponibilidade (PRADA, 2010).

Quando o veículo se retira, a luz verde é ligada, haja vista que esse mecanismo é muito eficiente, pois, além de indicar a presença dos carros com as luzes, facilita a visualização a uma grande distância. Na entrada do estacionamento do Shopping há painéis que contabilizam a quantidade de vagas livres e ocupadas e indicam a direção por meio de setas. As Figuras 2 e 3 ilustram o funcionamento dos indicadores luminosos e contadores de vaga em uso no referido *shopping*.



Figura 2 Representação das luzes do sistema adotado pelo MorumbiShopping.

Fonte: PRADA (2010).



Figura 3 Ilustração dos Painéis do MorumbiShopping.

Fonte: PRADA (2010).

Outro exemplo, são os estacionamentos subterrâneos no Japão que armazenam tanto bicicletas quanto automóveis. Eles são totalmente automatizados e o cliente não tem acesso ao local onde o veículo é estacionado (KAWANAMI, 2014). Possuem um elevador que leva os carros até a vaga que estiver livre, nas quais os carros ficam uns sobre os outros de forma organizada sem a necessidade de ocupar grandes espaços na área urbana, além de promover uma maior segurança aos veículos (PORTAL METALICA, s.d).Vide as Figuras 4 e 5.



Figura 4 Estacionamento subterrâneo de bicicletas.

Fonte: Silvia Kawanami (2014).



Figura 5 Estacionamento subterrâneo de automóveis.

Fonte: Portal Metálica.

## 2.2 Arduino

O Arduino é uma placa micro controladora, criada em 2005 pelos pesquisadores *Massimo Banzì, David Cuartielles, Tom Igoe, Gianluca Martino e David Mellis*, os quais tinham como objetivo criar um dispositivo que fosse viável financeiramente e, ao mesmo tempo, eficiente. Além de ser mais acessível, a placa é considerada um *hardware* livre, em outras palavras, qualquer pessoa que se interessa pelo assunto pode



montar, modificar e melhorar o funcionamento do Arduino, procedendo do mesmo hardware básico (THOMSEN, 2014).

Salienta-se ainda que, segundo (SOUTO e et al., 2012), a plataforma de desenvolvimento Arduino é denominado uma plataforma de computação física, visto que existem vários componentes que se integram a ele sendo capaz de medir variáveis do ambiente físico, realizar cálculos numéricos e tomar decisões lógicas, como, por exemplo, sensores, *LEDs*, etc. Para iniciar a criação de um projeto Arduino é necessário estabelecer uma comunicação da placa com a IDE (*Integrated Development Environment*), a qual pode ser obtida através do site oficial do Arduino, cuja linguagem de programação é baseada em C e C++. Para maiores informações consultar o *site* disponível em <<https://www.arduino.cc/>>. Acesso em: 06 Jun 2018.

No mercado encontram-se diferentes modelos da placa Arduino, por exemplo, Arduino Uno-R3, Arduino Leonardo, Arduino *Mega* 2560 R3, dentre outros. Eles se diferem em relação ao número de pinos analógicos e digitais, fonte de alimentação e processamento interno de cada placa, sendo que umas podem possuir maior capacidade do que outras (SOUZA, 2013). A Tabela 1 exemplifica as características gerais da placa Arduino, as quais podem ser visualizadas na Figura 6.

Tabela 1 Características da placa Arduino.

<b>Características gerais da placa Arduino:</b>
Alimentação por meio do cabo USB ou fonte externa
Portas digitais e saídas analógicas
Comunicação serial

Fonte: Autor.

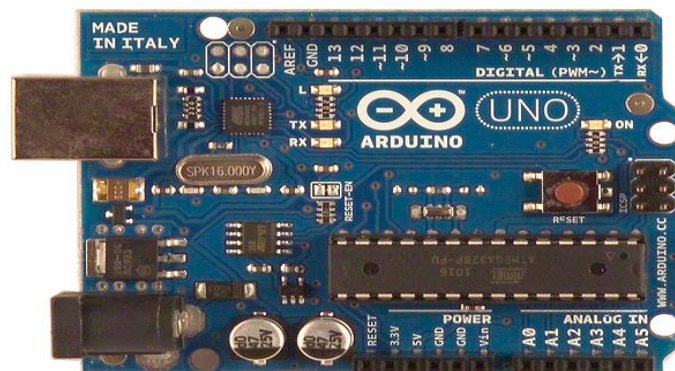


Figura 6 Placa Arduino Uno.

Fonte: Karla Soares (2013).

### 2.2.1 LEDs

A expressão *LED* deriva-se do termo inglês *Light Emitting Diode*, que na tradução para língua portuguesa significa diodo emissor de luz, eles são emissores de luz utilizados principalmente em produtos que envolvem microeletrônica, além disso, também são encontrados em circuitos em Arduino. Eles são movidos à uma fonte de energia externa, ou seja, precisam de energia elétrica para funcionar. A Figura 7 ilustra os *LEDs*.



Figura 7 Representação dos LEDs.

Fonte: Tts (2014).

### 2.2.2 Resistor

De acordo com o site Mundo da Elétrica, (2018) o resistor é um componente elétrico cuja função é limitar o fluxo da corrente elétrica. Em outras palavras, ele transforma energia elétrica em energia térmica através do efeito *Joule*. Esse fenômeno se explica devido a sua capacidade de possuir uma resistência maior do que as dos cabos do circuito. Conseqüentemente, isto força a redução da corrente que passa por ele, provocando uma queda de tensão.

Eles são elementos de suma importância no circuito, pois, uma vez que os componentes são ligados diretamente na fonte de energia, surge o risco de queimar os demais elementos. Assim, com a utilização dos resistores, esse risco pode ser evitado. Entretanto, necessitam de uma resistência específica para limitar a corrente, o que torna indispensável a realização dos cálculos para o correto dimensionamento de seu uso.

Dessa forma, sugere-se que sejam efetuados cálculos para saber qual o valor da resistência correta para a utilização dos componentes. A Figura 8 ilustra como calcular a resistência por meio da lei de *Ohm*, nela está exemplificado qual os respectivos valores da tensão e da corrente, onde a tensão total equivale a 5 *Volts* e a do *LED* 2 *Volts*, além da corrente que é igual a 0,03 *Amperes*. Substituindo esses valores na fórmula encontraremos qual o valor da resistência correta para ser utilizada.

#### Resistor para ligar um LED em 5V

$R = V / I$	Substituindo
$V = 5V$	$R = (5V - 2V) / 0,03A$
$V_{Led} = 2V$	$R = 3V / 0,02A$
$I = 30 \text{ mA} = 0,03A$	$R = 150 \text{ Ohms}$

Figura 8 Lei de *Ohm*

Fonte: Mundo da Elétrica (s.d).

Portanto, a maioria dos resistores possuem marcações coloridas para distinguir o valor da resistência de cada um deles. Cada resistor possui quatro faixas, sendo que a faixa inicial atribui o valor da resistência do primeiro dígito, a segunda faixa informa o segundo dígito, já terceira faixa indica um fator de multiplicação, por fim, a quarta faixa exprime a tolerância do resistor (Mundo da Elétrica, s.d). A Figura 9 ilustra a variedade de resistores que existem e suas respectivas cores indicando qual o valor da resistência.



Figura 9 Exemplo da diferença entre os resistores.

Fonte: Baú da eletrônica (2017).

### 2.2.3 LDR

Segundo BRAGA, (2017) o *LDR* é um componente que se comporta como resistor, cuja resistência depende da intensidade luminosa que incide sobre o mesmo. Além desses aspectos, eles possuem a capacidade de diminuir a resistência quando uma energia luminosa o atinge e isto possibilita a utilização desse elemento em várias aplicações como, por exemplo, na aplicação de sensores que são ativados e desativados quando entram em contato com a luz.

Salienta-se ainda que, para (MOTA, 2017), o *LDR* é constituído de materiais semicondutores que apresentam alta resistência. A medida em que recebe uma grande quantidade de fótons provenientes da luz incidente ele absorve elétrons que aprimoram sua condutividade, a fim de reduzir sua resistência. Dessa forma, esse semicondutor pode assumir resistências na ordem de *mega Ohm* no escuro e resistência na ordem de poucas centenas quando exposto à luz. A Figura 10 apresenta uma representação ilustrativa de *LDR*.

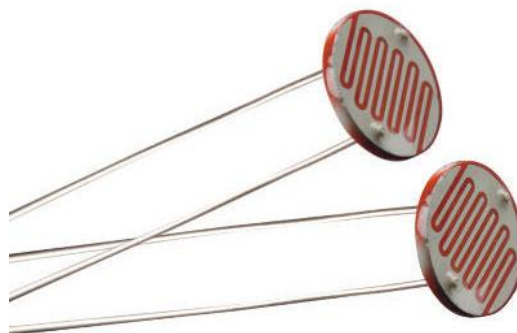


Figura 10 Representação do LDR.

Fonte: Indiamart (2011).

## 2.2.4 Servo Motor

Os servos motores podem ser compreendidos na categoria de motores especiais pelo fato de serem julgados como motores de precisão e estarem intrínsecos em várias atividades como sistema de posicionamento, robótica industrial e educativa, etc.

Os servos motores apresentam um leque de características que os distinguem dos restantes motores. Estes apresentam uma elevada dinâmica, isto é, funcionam a várias velocidades com binário constante, possuem também um rigoroso controlo de binário, isto é, pode-se facilmente atuar sobre o binário do servo motor por forma a aumentar e diminuir o mesmo sem prejudicar o servo motor. Além destas características os servos motores apresentam uma enorme precisão da sua velocidade bem como da posição do seu rotor, devendo-se isto ao fato dos seus dispositivos de controlo (MATOS, 2012).

Especificadamente, o servo motor utilizado no projeto é um Micro servo 9g *TowerPro*, que é o mais comum na utilização em projetos envolvendo Arduino e possui um carácter mais habitual se comparado aos demais. Sua rotação máxima é de 180 graus sendo que ele alcança a velocidade de 0,12 segundos para 60 graus, dessa forma, ele supre todas as necessidades do projeto. A Figura 11 é uma imagem ilustrativa de um Micro Servo 9g *TowerPro*.



Figura 11 Micro Servo 9g *TowerPro*.

Fonte: FilipeFlop.(s.d)

### 2.2.5 Display LCD

Para (BECKER e DOTTO, 2013) o *LCD (Liquid Crystal Display)* é um componente eletrônico e possui a função de exibir caracteres em sua interface com o intuito de interagir com o usuário por meio de mensagens, números ou até mesmo símbolos especiais. Existem no mercado inúmeras variedades de *LCD* com capacidade de exibir gráficos, imagens e textos com diferentes formatos e dimensões. No projeto, foi utilizado o *Display LDC 16x2*, ou seja, ele possui dezesseis colunas e duas linhas. O modelo ilustrado na Figura 12 é mais simples e usual e possibilita a visualização de frases e símbolos especiais.

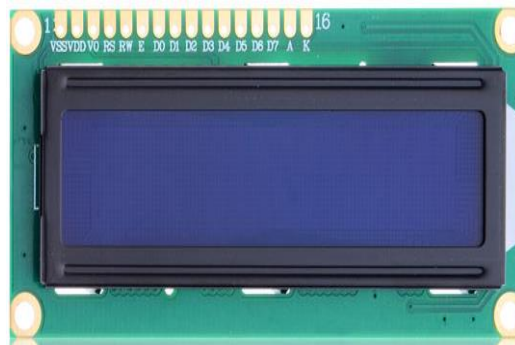


Figura 12 *Display LCD 16x2*.

Fonte: Filipeflop (s.d).

### 2.2.6 Protoboard

Na terminologia da língua portuguesa, *Protoboard* significa matriz de contato. Em outras palavras, esse termo nos refere ao ambiente onde serão realizadas as ligações entre os componentes eletrônicos e a placa Arduino por meio de fios ou *jumpers*. Segundo (MOTA, 2018), para que haja a transferência de corrente elétrica entre os fios e os elementos, deve haver em sua parte interior conexões que interligam eletricamente os componentes. As *protoboards* podem se diferenciar por modelos verticais e horizontais, o que irá influenciar no modo em que as ligações serão

realizadas. Além disso, elas possuem terminais positivos e negativos que estão representados na Figura 13, os quais ligam as portas 5V e GND do Arduino.

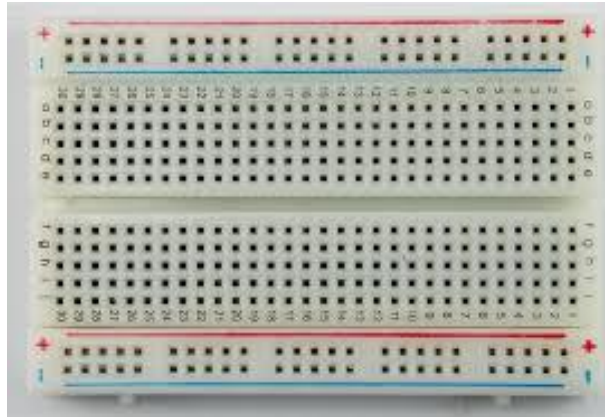


Figura 13 Protoboard.

Fonte: Ebanataw (2018).

## 2.2.7 Potenciômetro

De acordo com (BARBOSA e et al., 2012) o potenciômetro é um elemento muito utilizado na eletrônica devido ao seu caráter de um resistor ajustável. A maioria dos modelos caracterizam-se por possuir três pinos sendo que sua conexão central é manipulável, tais características podem ser visualizadas na Figura 14. Além disso ele pode atuar como um divisor de tensão se os três pinos forem utilizados. No projeto utilizou-se potenciômetro para regular a luminosidade do *Display LCD*.



Figura 14 Ilustração do potenciômetro.

Fonte: Baú da Eletrônica.

## 2.3 *Bluetooth*

Para (ALECRIM, 2018) o *Bluetooth* é uma tecnologia que não necessita de fiação para trocar informações com outros dispositivos devido ao seu mecanismo de rádio frequência, ou seja, os dados são transmitidos por meio de ondas, as quais se propagam até chegar no sinal desejado, vale ressaltar que a distância de sua propagação pode variar de acordo com o modelo do *Bluetooth*, podendo alcançar até 100 metros. Além disso, o *Bluetooth* não necessita de internet que é um ponto positivo para quem visa praticidade. Devido a essa simplicidade foi escolhido para estar presente no projeto, pois oferece uma maior acessibilidade aos motoristas que não possuem acesso à internet no momento em que desejam acessar o aplicativo.

### 2.3.1 Módulo de Conexão *Bluetooth* HC- 06

Apesar de a placa Arduino possibilitar diversas aplicações, ela não tem conexão *Bluetooth* de forma nativa, sendo assim, é necessário a aquisição de módulos externos que são capazes de estabelecer tal conexão. Para (ALMEIDA, 2017) o módulo HC – 06 possibilita que o Arduino se comunique com dispositivos que possuem conexão *Bluetooth*, permitindo a troca de informações entre ambos. O módulo de conexão *Bluetooth* proposto por ele apresenta as seguintes características e pode ser visualizado na Figura 15:

- Pinos para transmissão e recepção de dados;
- Pinos de alimentação do sistema;
- Tensão de funcionamento 3.3V, 5V;
- Taxa de transmissão 2Mbps.





Figura 15 Módulo Bluetooth HC-06.

Fonte: Vida de Silício (2017).

## 2.4 *TinkerCAD*

Na perspectiva de (PRADO, 2018) o *TinkerCad* da é uma plataforma online e gratuita que permite realizar a montagem e a simulação de circuitos eletrônicos, bem como, a criação de desenhos 3D. A ferramenta é bem intuitiva e simples de usar, apresentando galerias com diferentes exemplos, além de tutoriais detalhados do seu funcionamento. A Figura 16 ilustra o logo da referida ferramenta.



Figura 16 Logo da ferramenta *TinkerCad*.

Fonte: Autodesk *TinkerCad* (s.d).

## 2.5 *AppInventor*

Segundo (LULCHEVA, 2017) o *AppInventor* é uma ferramenta cuja linguagem de programação é feita por meio de blocos, visando a acessibilidade entre leigos e indivíduos que já são experientes em programar. Ele permite o desenvolvimento de

aplicativos para dispositivos móveis Android de maneira simplificada e intuitiva se comparado as plataformas tradicionais de programação, como por exemplo, o Android Studio.

Ademais, para (GOMES e MELO, 2013) ele é uma plataforma que está disponível na *web* e qualquer pessoa pode ter acesso, desde que efetue o *login*. Sua linguagem de programação é inspirada em *Logo* e *Scratch*. O programa permite realizar aplicações que englobem serviços baseados na *web*, bem como, a interação com redes sociais e sensores de orientação e geolocalização, leitura de códigos de barra, entre outros. Devido a esses motivos, o discente optou junto com o orientador, a utilização da ferramenta para o desenvolvimento do aplicativo móvel. A Figura 17 ilustra o logo da plataforma de programação do *App Inventor*.



Figura 17 Logo da ferramenta *AppInventor*.

Fonte: MIT *AppInventor* (s.d).

## 2.6 *Processing*

Como descrito pelo seu site oficial (PROCESSING, 2018), o *Processing* surgiu em 2001, no *Massachsetts Institute of Technology* (MIT) como uma linguagem de programação de código livre, a fim de favorecer a área das artes visuais. Inclusive ele é bastante utilizado por artistas, *designers* e arquitetos para criar animações e prototipagem de projetos. Seu código de programação é baseado em Java e possibilita a integração com Arduino, podendo comunicar com a porta serial por possuir bibliotecas destinadas ao uso.

## 3 MATERIAIS E MÉTODOS

Nesta seção serão abordados quais foram os materiais usufruídos no decorrer do projeto e os métodos para desenvolver e finalizar o trabalho.

### 3.1 Materiais

Para o desenvolvimento do projeto utilizou-se, na maioria das tarefas, o computador pessoal cujas especificações são: Modelo: *Aspire A515-51*; Processador: Intel Core i5-7200U 2.5GHz; RAM: 8GB; Sistema Operacional: Windows 10 Home - 64 bits, as quais permitiram que as atividades fossem cumpridas sem interferência de outras máquinas. Além disso, utilizou-se as ferramentas *TinkerCad*, *AppInventor* e *Processing*, que foram citadas no referencial teórico. Para realizar o teste do aplicativo móvel utilizou-se o celular Samsung J7 Prime. Acrescente-se que, para a construção do circuito, foram utilizados vários componentes eletrônicos abaixo descritos na Tabela 2.

Tabela 2 Descrição dos componentes do circuito.

<b>Componentes</b>	<b>Quantidade</b>
Placa Arduino Uno	01
Módulo <i>Bluetooth</i>	01
<i>LEDs</i>	04
<i>LDR</i>	03
Servo motor	02
<i>Display LCD</i>	01
<i>Buzzer</i>	01
Potenciômetro	01
<i>Protoboard</i>	01

Fonte: Autor.

A tabela 2 exemplifica os componentes que foram utilizados na elaboração do circuito eletrônico, por meio de uma simulação realizada na plataforma *TinkerCad*, a qual possibilitou a vivência com os elementos, além da experiência do funcionamento do estacionamento.

## 3.2 Metodologia

A metodologia utilizada no projeto baseia-se no modelo em cascata, considerando que cada tarefa só iniciava após o término da outra. O presente estudo foi desenvolvido com o intuito de amenizar os contratempos dos motoristas no que diz respeito a procura por vagas nos estacionamentos. Os dados utilizados foram obtidos por meio de pesquisas na internet e observações realizadas na cidade de Formiga, onde as vagas de estacionamento atrapalham o fluxo de veículos nas ruas. Para chegar a uma conclusão, algumas etapas foram desenvolvidas como: definir e delimitar o escopo do projeto, pesquisar e aprender sobre as ferramentas e tecnologias, e implementar o código das simulações com diferentes linguagens de programação. Diante do exposto, cada uma das subseções que seguem irá especificar detalhadamente cada uma destas etapas que estão ilustradas de forma resumida na Figura 18.

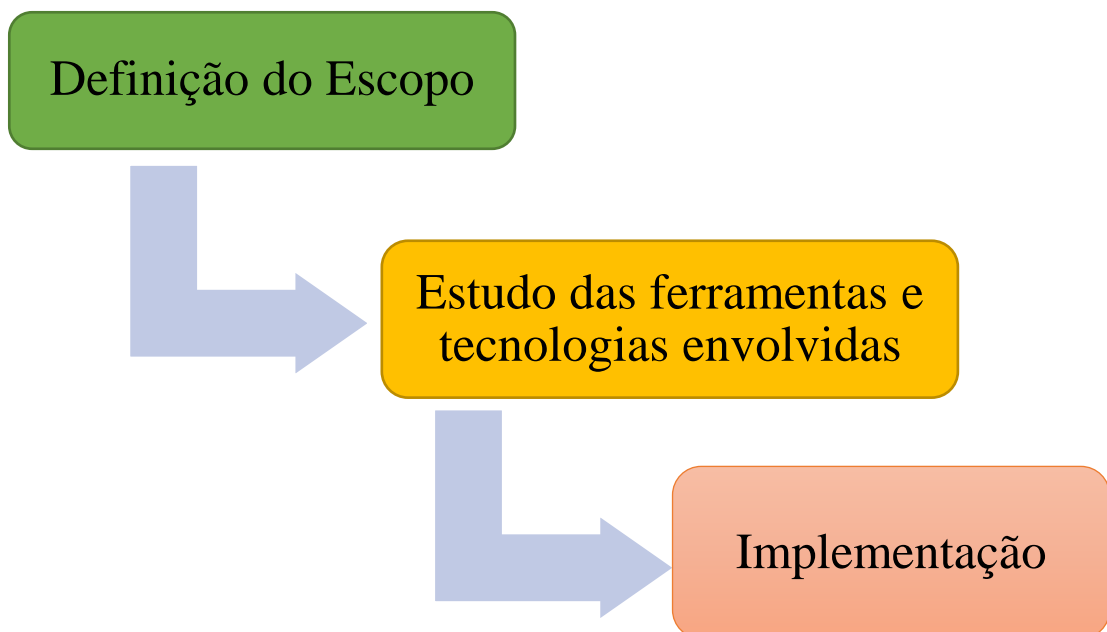


Figura 18 Metodologia

Fonte: Autor.

### 3.2.1 Definição do Escopo

Inicialmente, delimitamos o escopo do projeto a fim de não ultrapassar o tempo estipulado e deixar claro quais são suas limitações. Foram definidos que no circuito os servos motores controlam a entrada e saídas dos automóveis, que os *LEDs* indicam em quais das vagas o motorista pode estacionar e, além disso, que o *Display LCD* contabiliza o número de vagas disponíveis e ocupadas e exibe as informações em sua tela. Para o motorista interagir com o sistema optamos por elaborar um aplicativo simples, para que o mesmo pudesse visualizar qual vaga estaria ocupada e não perder tempo na procura. A princípio tinha-se a intenção de construir uma maquete física, porém, devido ao curto tempo e aos gastos que iriam envolver na sua construção, optamos em realizar uma simulação do estacionamento no *Processing*.

### 3.2.2 Estudo das ferramentas e tecnologias envolvidas

A pesquisa e aprendizagem das ferramentas e tecnologias envolvidas aconteceram de forma virtual. Foram realizados estudos para compreender a linguagem de programação do Arduino, *Processing* e *AppInventor*. A princípio, realizou-se dois cursos *online* sobre como utilizar o Arduino com o uso do simulador e como utilizá-lo na prática.

Os cursos estão disponíveis na plataforma *Udemy*, instruídos pelos professores *Vitor Mazuco* e *Roni Shigueta*. Ambos aconteceram por meio de vídeos que eram divididos em seções que contemplavam diferentes tópicos, exemplificando vários componentes eletrônicos e distintas aplicações. Salienta-se ainda que os cursos foram de grande aplicabilidade, pois facilitaram a montagem do circuito eletrônico e sua programação além de fornecerem certificados que confirmam sua conclusão. As Figuras 19 e 20 ilustram os certificados que o discente recebeu após o término dos cursos que o orientador disponibilizou.



Figura 19 Certificado do curso Aprenda Arduino com uso de simulador.

Fonte: *Udemy*.



Figura 20 Certificado do curso Arduino- Curso completo e prático.

Fonte: *Udemy*.

Em seguida, após o estudo sobre o funcionamento do Arduino, deu-se início aos estudos sobre o a ferramenta *AppInventor*. A aprendizagem aconteceu através de vídeos no *YouTube*, além de pesquisas em *Websites*, os quais ensinavam sobre como manusear a plataforma. Acrescenta-se que, para a aprendizagem da linguagem *Processing*, o orientador disponibilizou três tutoriais explicativos para introduzir o conceito e mostrar as funções que seriam utilizadas no projeto.

### 3.2.3 Implementação

Após o estudo e aprendizagem das ferramentas e tecnologias, iniciou-se a implementação das simulações. Nesta etapa o desenvolvimento da maquete 3D, a execução do circuito eletrônico, a realização do aplicativo móvel e a simulação do estacionamento foram implementados e testados para assegurar que ambos estariam funcionando em boas condições. Portanto, para maiores detalhes sobre a implementação e resultados obtidos consultar a **Seção 4**.

## 4 DESENVOLVIMENTO

### 4.1 Maquete 3D

A maquete 3D foi construída no *Tinkercad*, através de formas geométricas básicas que foram arrumadas de modo a originar a estrutura desejada. Ela foi organizada sobre medida, pensando no conforto do motorista para realizar manobras em segurança. O estacionamento é de 90° e as vagas seguem um padrão imposto pela ABNT. Cada segmento possuiu 2,5 de largura e 5,0 de comprimento com espaço de 4,5 ao centro destinado as manobras. Além disso, também é estipulado que nos estacionamentos o número de vagas é superior a dez, seja obrigatório a presença de vagas especiais, as quais, possuem 1,2 metros a mais das demais (OLIVEIRA, s.d).

As Figuras 21 e 22 representam a estrutura 3D para simular o estacionamento, e adquirir uma visão genérica de como seria o estabelecimento, além de possibilitar a organização da posição de cada componente. As bolinhas amarelas representam os *LEDs*, os tracejados posicionados no chão ilustram as vagas, além disso a casinha posicionada na entrada equivale ao local em que o *display* ficará posicionado, e por fim, as duas barras horizontais no começo correspondem as cancelas de entrada e saída, as quais representam o lugar onde o servo motor se posicionará.

Mas, no decorrer do desenvolvimento optamos em realizar uma simulação, tendo em vista o curto tempo, os gastos e complexidade que sua construção física iria envolver. No entanto, foi modelada a estrutura 3D considerando a hipótese da implementação de trabalhos futuros. No Apêndice C está contido um desenho esquemático com as medidas da maquete.



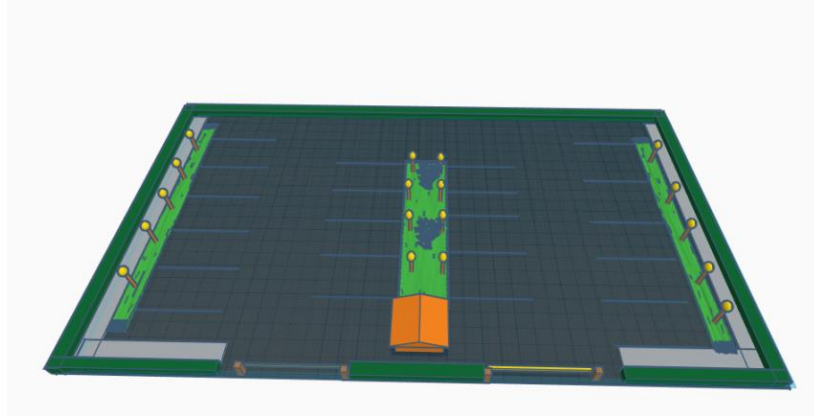


Figura 21 Maquete do estacionamento (visão superior).

Fonte: Autor.

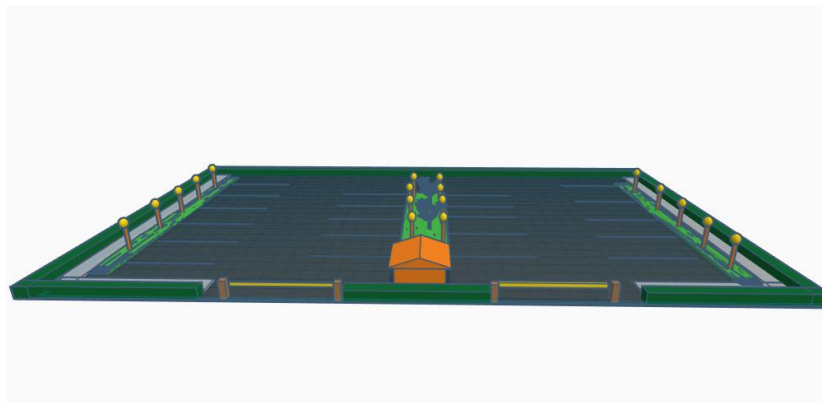


Figura 22 Maquete do estacionamento (visão lateral).

Fonte: Autor.

## 4.2 Circuito Eletrônico

O circuito consistia na simulação do estacionamento que contemplasse vinte vagas. No entanto, a ferramenta *TinkerCad* não possui todos os itens que seriam necessários para a montagem, então realizamos a montagem genérica de duas vagas, as quais se aplicariam para as demais. Em seguida, foram implementados os códigos de cada elemento separadamente para certificar que eles estavam conectados corretamente, com isso, logo após testá-los começou a implantação do código do circuito. Quando a programação foi finalizada, iniciou-se o processo de testes para certificar que o circuito estava funcionando como o esperado e também, para implementação de melhorias.

Além disso, foram realizadas montagens de alguns circuitos para o aprendizado dos componentes. As Figuras 23 e 24 apresentam algumas das montagens realizadas.

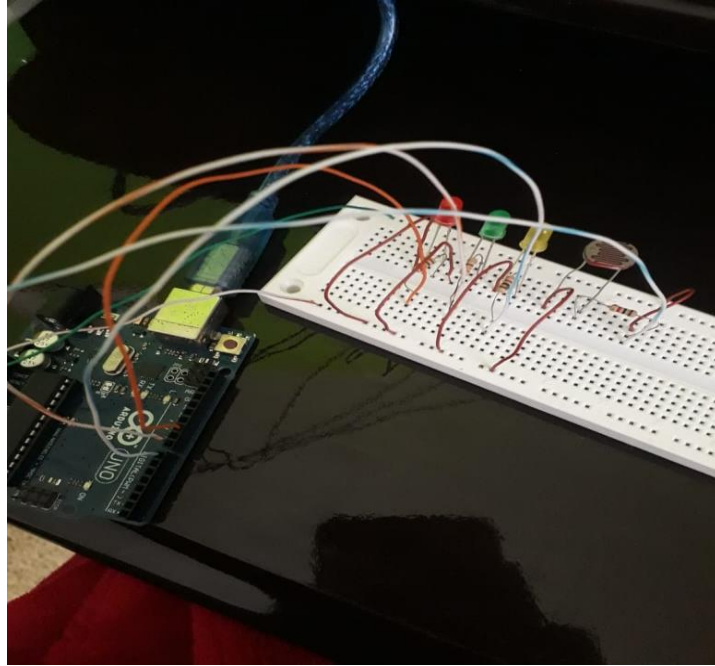


Figura 23 Ilustração dos *LEDs* que representam uma vaga.

Fonte: Autor.

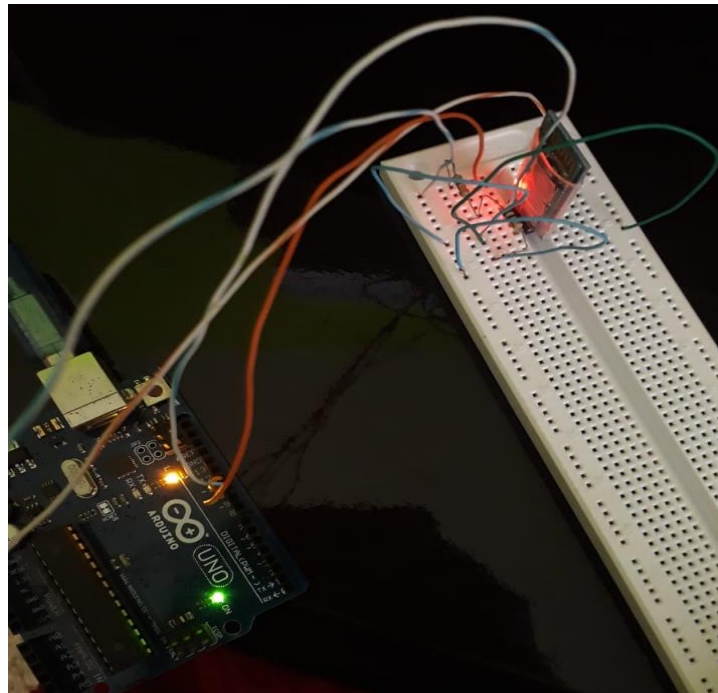


Figura 24 Teste com o módulo Bluetooth.

Fonte: Autor.

O circuito é composto por uma placa Arduino, a qual é um elemento de suma importância na montagem pois sem ela os outros componentes não funcionariam, além de *leds*, *ldr*, fios para conexão, resistores, *buzzer*, *display lcd*, protoboard, potenciômetro, e servos motores. Os *leds* verdes representam disponibilidade e os vermelhos ilustram indisponibilidade, assim, cada par de *led* corresponde a uma vaga, a qual poderá estar livre ou ocupada sendo contabilizada e lida pelo *display lcd*.

Acrescenta-se ainda, que a programação do circuito foi desenvolvida na linguagem C/C++, que é bem semelhante à linguagem de programação Java. Devido a isso o discente não enfrentou muitas dificuldades, visto que no curso Técnico em Informática ele obteve dois anos de aulas de Programação Orientada a Objetos, o que auxiliou bastante na hora da codificação.

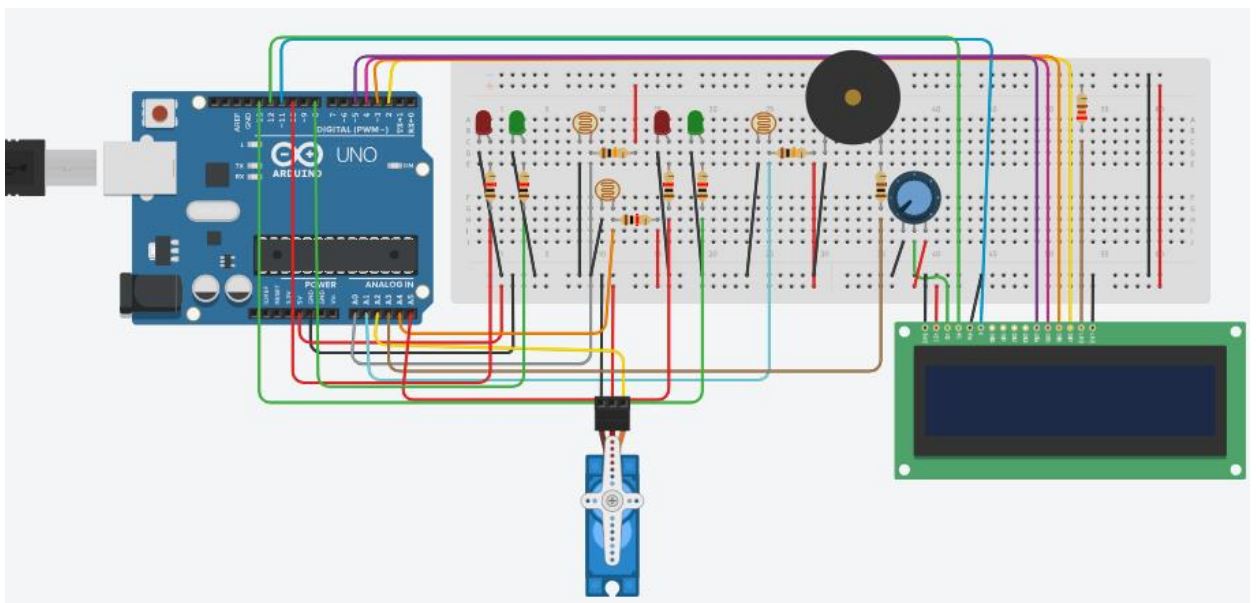


Figura 25 Circuito eletrônico do estacionamento.

Fonte: Autor.

É apresentado na Figura 26 o funcionamento do circuito. De forma análoga ao estacionamento físico, o circuito inicia-se quando o *ldr* da entrada constata que a luminosidade do mesmo indica a presença do automóvel, então, o servo motor gira formando um ângulo de 90 graus simbolizando a entrada do veículo. A medida em que o carro estaciona sobre a vaga, que é representada pelos pares de *leds* e o *ldr*,

automaticamente a cor do *led* modifica-se para vermelho e a vaga é contabilizada pelo *display LCD* que ilustra quantas estão disponíveis.

Ademais, para que haja acessibilidade para os portadores de deficiências, foi desenvolvido um código de acesso que permite somente a eles estacionar na vaga específica. Quando o condutor não possuir tal código e estacionar na vaga especial, um alarme será acionado, o qual é representado pelo *buzzer* e pelo *led* vermelho. Em caso de dúvidas e necessidade de mais informações, o código do circuito está contido no Apêndice A.

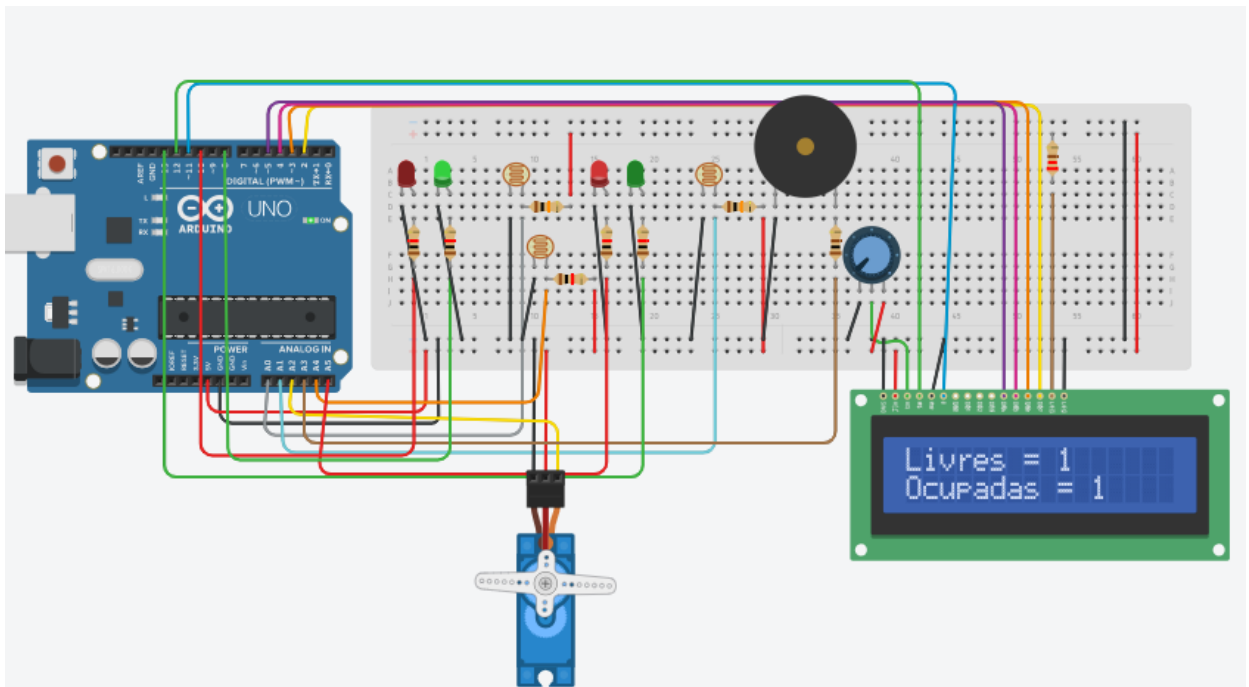


Figura 26 Circuito eletrônico em funcionamento.

Fonte: Autor.

### 4.3 Construção do Aplicativo Móvel

O aplicativo móvel tem como principal objetivo fazer uma interação simples e usual com o Arduino e o *Processing*, para que o motorista possa identificar qual vaga ele deseja ocupar. O *app* foi desenvolvido pensando na praticidade e usabilidade do mesmo devido a isso, sua interface é bem simples contendo apenas uma tela que se

resume exclusivamente em uma breve explicação do projeto e um mapa do estacionamento.

Para que haja uma interação com os outros *softwares* o usuário deve primeiramente fazer a conexão com o *Bluetooth*, então se o celular não estiver conectado o aplicativo envia uma mensagem perguntando se o usuário deseja conectar, se ele responder sim o *Bluetooth* do celular é ativado.

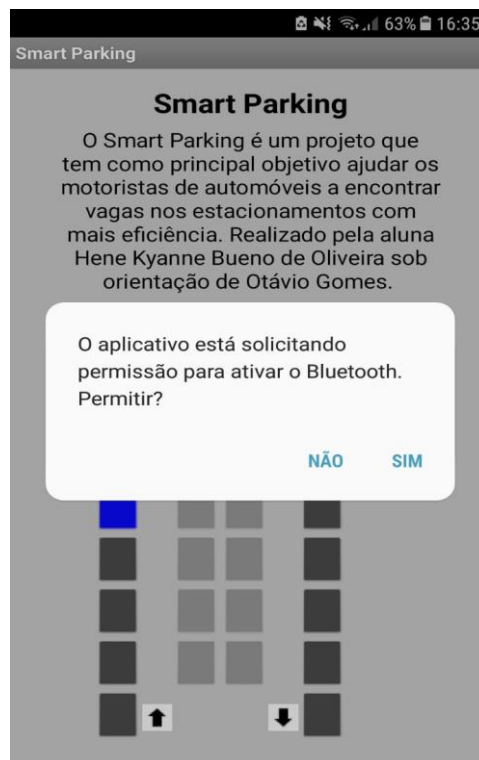


Figura 27 Permissão para ativar o Bluetooth.

Fonte: Autor.



Figura 28 Bluetooth do smartphone ativado.

Fonte: Autor.

Como podemos observar nas Figuras 27 e 28 o aplicativo envia a solicitação para a ativação do *Bluetooth* e se o usuário clicar em SIM o *Bluetooth* do celular é ativado, se ele clicar em NÃO, não haverá mudanças, pois, o aplicativo só funciona depois que o módulo *Bluetooth* é conectado.

Após a conexão do *Bluetooth* do celular é necessário conectar o *Smartphone* ao Arduino. Para que isso aconteça, basta o usuário clicar em conectar, assim que ele acionar o botão aparecerá uma lista com os nomes dos dispositivos ativos, então ele deve escolher a opção que corresponde ao módulo. Quando a conexão for estabelecida o botão conectar troca o nome para desconectar, e embaixo aparece qual dispositivo está conectado. Para maiores detalhes consultar o Apêndice B referente ao código que realiza a conexão *Bluetooth*.



Figura 29 Lista dos dispositivos com *Bluetooth* ativado.

Fonte: Autor.

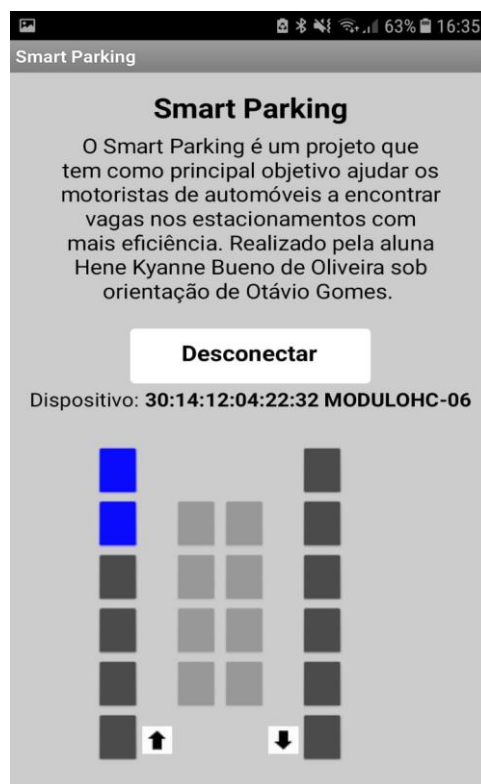


Figura 30 Aplicativo conectado ao módulo.

Fonte: Autor.

Como o *TinkerCad* não possui o módulo *Bluetooth* HC-06 os experimentos foram realizados na placa física, a fim de testar a comunicação entre o aplicativo e o Arduino. A Figura 31 ilustra o circuito em que a montagem da placa física foi inspirada.

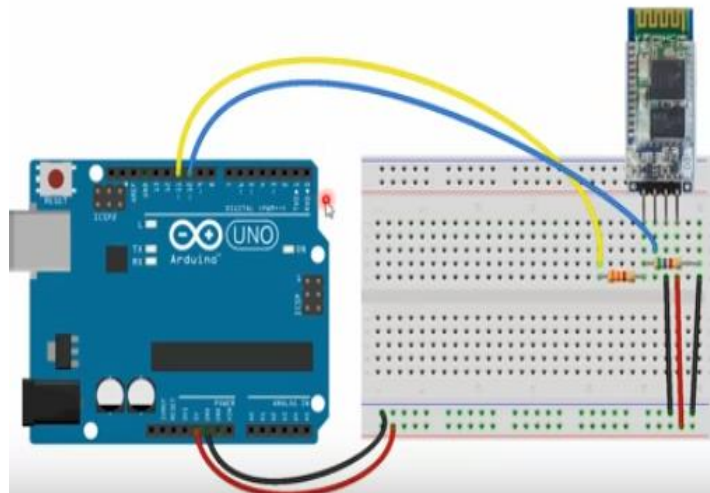


Figura 31 Circuito do módulo *Bluetooth*.

Fonte: Brincando com ideias (s.d).

Cada botão representa uma vaga. Os das cores cinzas não possuem restrições, porém os azulados simbolizam as vagas para portadores de deficiência. As setas na parte inferior da tela ilustram a entrada e saída do estabelecimento. Assim que o usuário escolher qual vaga deseja, ele deve clicar nela e automaticamente sua cor mudará para vermelho indicando que aquela vaga foi ocupada, ressaltando que cada motorista deve escolher qual vaga está livre para que não haja colisão com as vagas escolhidas por outrem.

No entanto, a interação só ocorre após a conexão com o *Bluetooth*, ou seja, para o motorista escolher qual vaga ele deseja primeiramente deve conectar o aplicativo ao módulo. Se o aplicativo não estiver conectado uma mensagem de erro é enviada ao usuário. As Figuras 32 e 33 ilustram respectivamente a mensagem de erro enviada ao indivíduo e as vagas que foram escolhidas pelos motoristas.



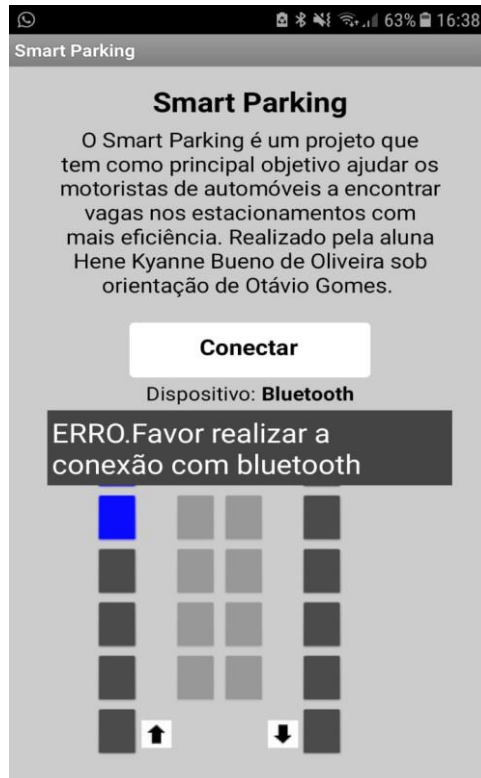


Figura 32 Mensagem de erro.

Fonte: Autor.

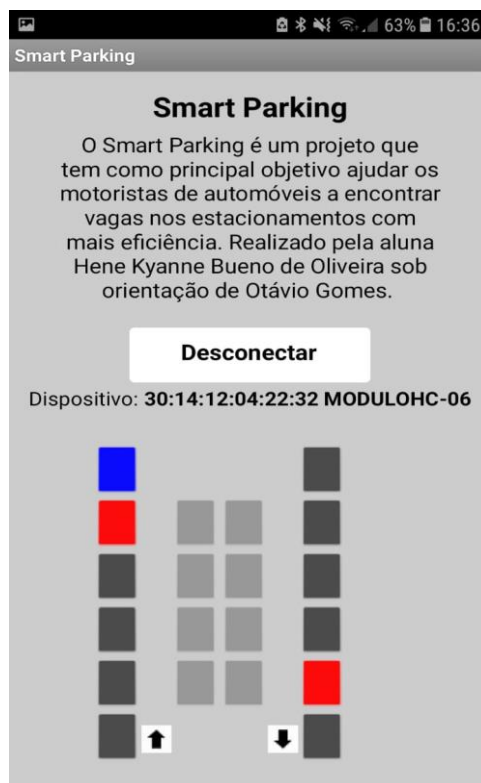


Figura 33 Ilustração das vagas ocupadas.

Fonte: Autor.

Portanto, o código que permite a interação entre o usuário e o aplicativo é igual para todas as vagas, tendo em vista que todas elas possuem o mesmo mecanismo. Dessa forma, apenas as vagas das laterais foram ativadas. Após pressionar a vaga o código verifica se o *Bluetooth* está ativado e conectado, se não estiver ele envia uma mensagem de erro informando aos usuários que a conexão precisa ser estabelecida.

Em seguida, é averiguado se a variável global equivale a “*OFF*”, então se essa afirmação estiver verdadeira, a cor de fundo do botão modifica-se para vermelho e consequentemente a variável muda para “*ON*” e envia uma mensagem de texto para o Arduino, por meio da conexão *Bluetooth*. Logo, se a variável global for diferente de “*OFF*” a cor do botão permanece a mesma e uma mensagem é enviada ao Arduino para que o *LED* seja apagado. A Figura 34 representa o código das vagas.

```

initialize global start to " off "

when vagaespecial1 .Click
do
  if BluetoothClient1 . Enabled and BluetoothClient1 . IsConnected
  then
    if get global start = " off "
    then
      set vagaespecial1 . BackgroundColor to red
      set global start to " on "
      call BluetoothClient1 . SendText
      text " a "
    else
      set vagaespecial1 . BackgroundColor to black
      set global start to " off "
      call BluetoothClient1 . SendText
      text " b "
  else
    call Notifier1 . ShowAlert
    notice " ERRO.Favor realizar a conexão com bluetooth "
  
```

Figura 34 Codificação das vagas.

Fonte: Autor.

## 4.4 Simulação no *Processing*

A simulação no *Processing* foi desenvolvida com auxílio do orientador além de pesquisas e tutoriais disponibilizados por ele. A sua interface foi inspirada na a maquete 3D, e os retângulos foram inseridos uns ao lado dos outros para representar cada vaga. Seu principal objetivo é fazer a interação com o aplicativo móvel que se conecta com a simulação por meio do módulo *Bluetooth* adicionado à placa Arduino. O usuário pode escolher sua vaga pela simulação como também, pelo aplicativo.

A simulação inicia-se com as vagas desabilitadas, exemplificando que o estabelecimento está fechado, em outras palavras, as vagas começam com a cor branca para que em seguida elas possam ser habilitadas trocando sua cor para verde, ressaltando que as vagas para portadores de deficiências iniciam-se com a cor azul. A Figura 35 ilustra todas as vagas do estacionamento desabilitadas.

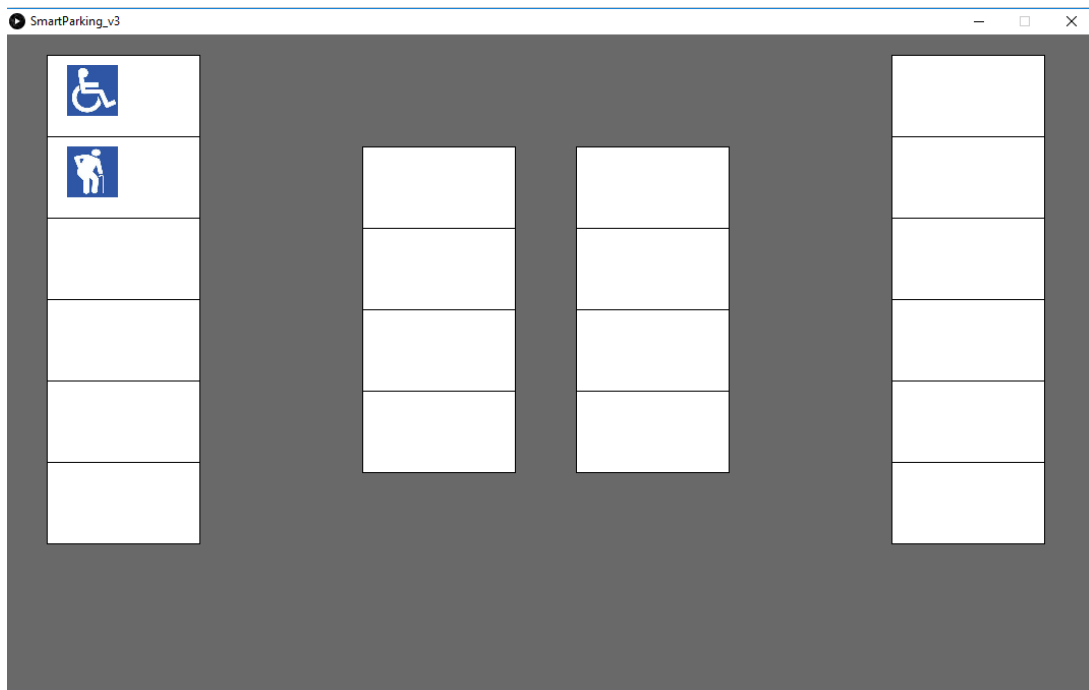


Figura 35 Ilustração das vagas desabilitadas.

Fonte: Autor.

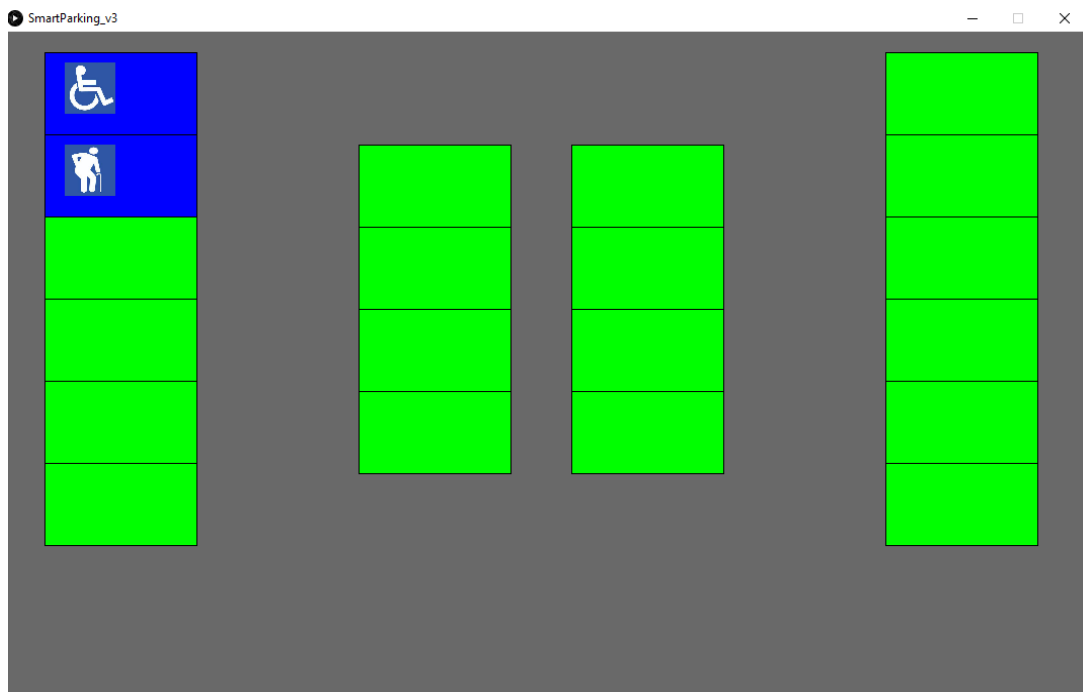


Figura 36 Ilustração das vagas habilitadas.

Fonte: Autor.

Posteriormente, quando as vagas estiverem habilitadas, o usuário poderá escolher qual vaga deseja tanto pela simulação quanto no aplicativo. Na simulação quando a vaga for pressionada sua cor mudará para vermelho, de forma análoga ao estacionamento físico, quando o condutor escolhe o segmento para estacionar o veículo. Já no aplicativo móvel o motorista também escolhe a vaga desejada, a qual envia informações para que a simulação modifique sua cor para vermelho indicando indisponibilidade.

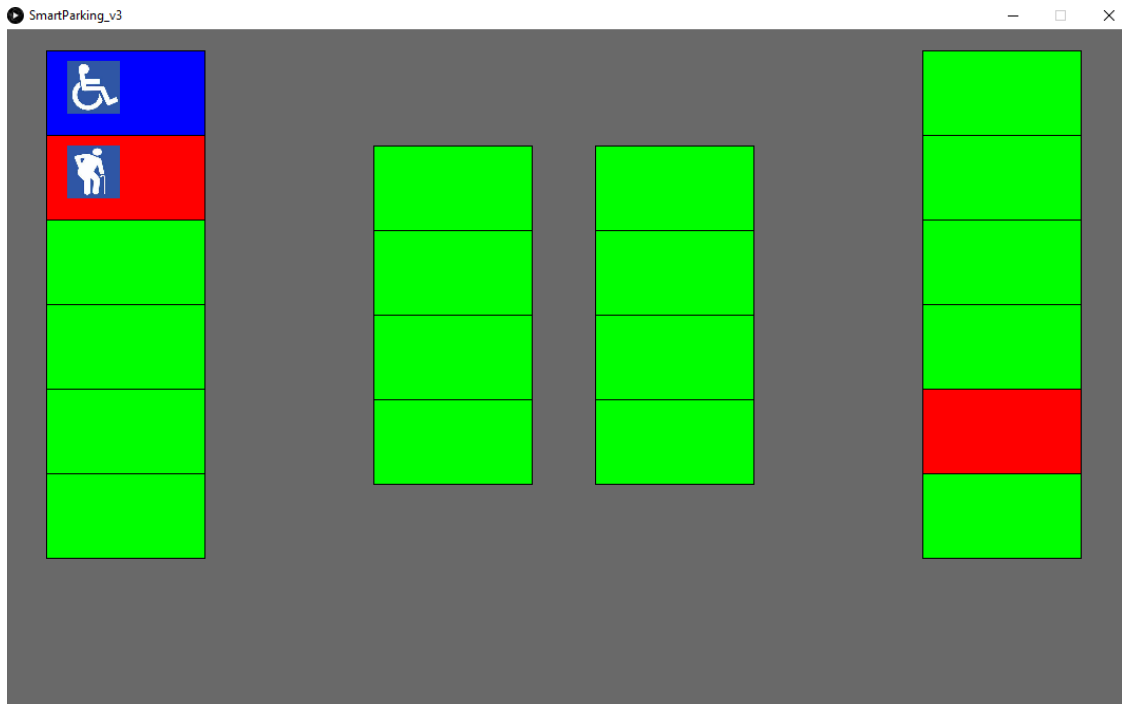


Figura 37 Vagas ocupadas.

Fonte: Autor.

## 4.5 Problemas Enfrentados

Inicialmente o discente iria fazer outro projeto, que consistia na reconstrução e manutenção de máquinas em algumas escolas públicas da cidade de Formiga MG, mas infelizmente, devido a problemas relacionados a aprovação da documentação o projeto foi inviabilizado. Conseqüentemente houve mudanças em relação a escolha do tema. Devido a essas mudanças, o projeto atual *Smart Parking* foi impactado no que diz respeito ao tempo de execução, que se tornou reduzido, gerando uma prova de conceito simples, mas suficiente para validar a idéia de estacionamento inteligente.

Nos primeiros estágios de concepção do trabalho, pretendia-se fazer uma maquete física do estacionamento, controlada por vários componentes eletrônicos. No decorrer do desenvolvimento do projeto percebeu-se que o tempo estava limitado, considerando a complexidade que envolvia sua construção. Devido a isso, optou-se por fazer uma simulação no *Processing* que envolveria menos tempo e não prejudicaria a execução das outras tarefas previstas.

## **5 RESULTADOS E DISCUSSÕES**

Os resultados encontrados no presente trabalho foram excelentes. O desenvolvimento do simulador de protótipo foi de grande valia, visto que ele oferece subsídio a implementação de trabalhos futuros, com a finalidade de que sejam desenvolvidos, testados e inseridos no mercado. Apesar do discente ter encontrado algumas dificuldades na execução do projeto, o resultado final foi muito bom, considerando que todas as etapas foram cumpridas de acordo com o cronograma.

## 6 TRABALHOS FUTUROS

Muitas são as possibilidades para trabalhos futuros neste simulador de protótipo. Para aprimorar os resultados finais, seria conveniente optar por um sensor infravermelho ativo para deixar o protótipo mais realista, pois o *LDR* não seria muito usual na prática, visto que ele depende da luminosidade do local, que pode variar de acordo com cada ambiente. Além disso, a construção da maquete física seria de suma importância para demonstrar o funcionamento do circuito na prática.

Ademais, é aconselhável aperfeiçoar o aplicativo móvel, implementando melhorias nas reservas das vagas e na visualização em tempo real das mesmas. Pensando na continuação desse projeto, o ideal seria desenvolver um aplicativo no Android Studio, o qual é mais complexo e permite outras aplicações que no *App Inventor* não seriam possíveis.

Outrossim, seria bom que cada usuário tivesse sua conta no aplicativo, permitindo-o salvar suas rotinas, horários e o tempo em que irá ficar estacionado na vaga, uma vez que evitaria colisões com os horários de outros motoristas. Como o protótipo pode abranger tanto estacionamentos públicos quanto privados, seria interessante o aplicativo ter a função de *scan code*, em vez de usar resistores *LDR* para liberar a entrada no estacionamento. Além disso, para os privados o ideal seria que a função *scan code* fosse utilizada também como forma de pagamento.

Além dessas sugestões, seria interessante a inserção de uma terceira luz na cor amarela no circuito eletrônico, para aplicar nos estacionamentos públicos rotativos. Com o intuito de alertar o cliente que seu tempo está se esgotando, além de auxiliar no monitoramento das vagas para portadores de deficiência, uma vez que o cliente não possuir o código especial a luz amarela em conjunto com o alarme acenderá.

Salienta-se ainda que para as vagas exclusivas para portadores de deficiência seria eficaz que uma luz amarela apontasse quando o usuário não possuir o código juntamente com o alarme.

## 7 CONSIDERAÇÕES FINAIS

Para o desenvolvimento do *Smart Parking* o discente associou diversas áreas e conhecimentos adquiridos durante o curso Técnico em Informática. O aluno envolveu no projeto as disciplinas de Robótica, Programação Orientada a Objetos, Desenvolvimento Móvel, Infraestrutura e Organização de Computadores. Conseqüentemente, essas matérias contribuíram muito para a conclusão do trabalho, pois utilizou-se vários conceitos que envolviam cada uma.

O projeto foi espelhado em protótipos já existentes no mercado, no entanto, o discente pode aprofundar seus conhecimentos por meio de uma ampla pesquisa científica que foi utilizada para o embasamento do desenvolvimento do trabalho. A pesquisa assim como a execução do projeto proveu valiosos conhecimentos em relação a montagem de circuitos, codificação para Arduino, bem como, a linguagem de programação *Processing*.

Infelizmente, o tempo foi um grande fator limitante para a evolução do protótipo, visto que, o período destinado a implementação era apenas de 10 meses, com alguns imprevistos na escolha do projeto inicial impedindo sua execução. Assim deu-se início ao processo de mudança do projeto que resultou em um atraso de aproximadamente dois meses. Mesmo assim o aluno pode contornar o acontecido usufruindo o período das férias para prosseguir com o andamento do projeto.

No que diz respeito as plataformas utilizadas, todas foram fundamentais para o desenvolvimento. A plataforma *TinkerCad* é uma incrível ferramenta para o desenvolvimento, visto que, é gratuita e possibilita diversas aplicações como circuitos e desenhos 3D. Mas vale ressaltar, que ela não atendeu todas as expectativas, pois não oferecia vários componentes que seriam utilizados no projeto.

A linguagem *Processing* utilizada pelo aluno para realizar a simulação do estacionamento, bem como a comunicação com módulo Bluetooth, foi de grande valia visto que a mesma é bastante usual e atendia as necessidades do discente. Acrescenta-se que, através dessa simulação, foram adquiridos novos conhecimentos a respeito da linguagem de programação *Processing*.



Por fim, pode-se concluir que o desenvolvimento da simulação do protótipo embasa diversas possibilidades de projetos futuros, permitindo a implementação do trabalho aprimorado nos estacionamentos, o qual proporcionará grandes benefícios no âmbito social, visto que ele reduz contratempos e facilita o cotidiano de muitos condutores, devido a sua praticidade e usabilidade. Os objetivos foram todos cumpridos de acordo com o cronograma e o projeto poderá servir de base para novos alunos que desejem trabalhar com sistemas de estacionamento inteligente envolvendo Arduino.

## REFERÊNCIAS BIBLIOGRÁFICAS

AGARWAL, Kaustubh. **Smart Parking**. 2016. Disponível em: <<https://www.hackster.io/KaustubhAgarwal/smart-parking-bdfa99>>. Acesso em: 12 maio 2018.

AGÊNCIA EFE. **Estacionamento robotizado, a última moda nas congestionadas cidades chinesas**: O cliente deixa o veículo em um elevador e o sistema carrega o carro para uma vaga. 2016. Disponível em: <<https://epocanegocios.globo.com/Tecnologia/noticia/2016/05/estacionamento-robotizado-ultima-moda-nas-congestionadas-cidades-chinesas.html>>. Acesso em: 02 jul. 2018.

ALECRIM, Emerson. **Tecnologia Bluetooth: o que é e como funciona?** Disponível em: <<https://www.infowester.com/bluetooth.php>>. Acesso em: 08 jul. 2018.

ALESSANDRO MEIRELLES (Piracicaba). **'Estacionamento inteligente' mostra vagas desocupadas por aplicativo**: Sistema de sensores ópticos em Águas de São Pedro indica 300 vagas. Informação é vista por celular ou em totem instalado na entrada do Centro. 2014. Disponível em: <<http://g1.globo.com/sp/piracicaba-regiao/noticia/2014/11/estacionamento-inteligente-mostra-vagas-desocupadas-por-aplicativo-guadesaopedro.html>>. Acesso em: 17 jun. 2018.

ALMEIDA, Danilo. **Módulo Bluetooth HC-05 e HC-06 – Acionando um módulo relé via Bluetooth**. 2017. Disponível em: <<https://portal.vidadesilicio.com.br/modulo-bluetooth-hc-05-e-hc-06/>>. Acesso em: 09 jul. 2018.

AMIRSAB. **Arduino e Processamento, LDR**. 2010. Disponível em: <<https://chalmersphyscomp10.wordpress.com/2010/08/05/ldr/>>. Acesso em: 11 set. 2018.

ARDUINO + Bluetooth + App Inventor 2. [s.i]: Arduinolivre, 2014. (26 min.), son., color. Disponível em: <<https://www.youtube.com/watch?v=DbQoxSqPTaI>>. Acesso em: 11 jun. 2018.

ARDUINO E CIA. **Sensor de estacionamento / ré com Arduino e sensor ultrassônico HC-SR04**. 2014. Disponível em: <<https://www.arduinoecia.com.br/2014/04/sensor-de-estacionamento-re-com-arduino.html>>. Acesso em: 08 maio. 2018.

ARDUINO, Squids. **Projetos Arduino: Projeto 30 - Sensor de luminosidade LDR com sinalizador de Leds**. Disponível em: <<http://squids.com.br/arduino/index.php/projetos-arduino/projetos-basicos/115-projeto-29-sensor-de-luz-ldr-com-sinalizador-de-leds>>. Acesso em: 10 maio 2018.

ARDUINO. **Alterando o código para usar dois servos motores aleatoriamente.** 2015. Disponível em: <<https://forum.arduino.cc/index.php?topic=320362.0>>. Acesso em: 15 mar. 2018.

ARDUINO. **Contador de um estacionamento.** 2015. Disponível em: <<https://forum.arduino.cc/index.php?topic=354251.0>>. Acesso em: 05 abr. 2018.  
AUTOMSOLUTION (São Paulo). **Controle de estacionamento.** Disponível em: <<http://www.automsolution.com.br/services/controle-de-estacionamento-de-veiculos/>>. Acesso em: 03 jul. 2018.

BANDEIRA, Thayanne Barros; FONSECA, Wellington; FERNANDES, Filipe Cavalcanti. **PROTÓTIPO DE ESTACIONAMENTO AUTOMATIZADO UTILIZANDO MODELO COMPUTACIONAL MATRICIAL E MICROCONTROLADOR ARDUINO.** In: CONGRESSO NACIONAL DE MATEMÁTICA APLICADA À INDÚSTRIA, 18., 2014, Caldas Novas. **Anais...** Caldas Novas: [s.i], 2014. p. 1 - 8. Disponível em: <<http://pdf.blucher.com.br.s3-sa-east-1.amazonaws.com/mathematicalproceedings/cnmai2014/0151.pdf>>. Acesso em: 15 abr. 2018.

BARBOSA, Leonardo Barreto; MARCELO, Denival; JÁCOMO, Daniel. **FUNDAMENTOS DE CIRCUITOS ELÉTRICOS DIVISOR DE TENSÃO E POTENCIÔMETRO.** Barra Mansa: [s.i], 2012. 4 p. Disponível em: <<https://www.ebah.com.br/content/ABAAAfMRIAG/relatorio-03-potenciometro>>. Acesso em: 12 jun. 2018.

BECKER, Daiane; DOTTO, Karina Rosana. **DESENVOLVIMENTO DE UM CONTROLE DE ACESSO MICROCONTROLADO COM CÓDIGO DE BARRAS.** 2013. 57 f. TCC (Graduação) - Curso de Tecnologia em Manutenção Industrial, Universidade Tecnológica Federal do Paraná, Medianeira, 2013. Disponível em: <[http://repositorio.roca.utfpr.edu.br/jspui/bitstream/1/1378/1/MD\\_COMIN\\_2012\\_2\\_02.pdf](http://repositorio.roca.utfpr.edu.br/jspui/bitstream/1/1378/1/MD_COMIN_2012_2_02.pdf)>. Acesso em: 15 jul. 2018.

BRAGA, Giovani Eugenio Batista. **Controle para indicar quantidades de vagas livres e ocupadas.** 2016. Disponível em: <<http://labdegaragem.com/forum/topics/controle-para-indicar-quantidades-de-vagas-livres-e-ocupadas>>. Acesso em: 01 maio 2018.

BRAGA, Newton C. **LDR (ALM332).** Disponível em: <<http://www.newtoncbraga.com.br/index.php/almanaque-tecnologico/201-1/7547-ldr-alm332>>. Acesso em: 19 jul. 2018.

**Brasil entra na era do estacionamento robotizado.** 2015. Disponível em: <<http://jconline.ne10.uol.com.br/canal/veiculos/noticia/2015/11/18/brasil-entra-na-era-do-estacionamento-robotizado-208471.php>>. Acesso em: 05 jul. 2018.

**CEO. DIMENSÕES DE VAGAS DE GARAGEM E ROTAS INTERNAS.**

Disponível em: <<https://www.condominioemordem.com.br/vagas-de-garagem/>>. Acesso em: 20 mar. 2018.

**CIA, Arduino e. Controle de vagas de estacionamento com o sensor HC-SR04.**

2013. Disponível em: <<https://www.arduinoecia.com.br/2013/12/controle-de-vagas-de-estacionamento-hc-sr04.html>>. Acesso em: 28 abr. 2018.

**CONEXÃO** com o módulo Bluetooth - App Inventor 2 - Android #02. [s.i]: Vitor Santos, 2016. Son., color. Disponível em:

<<https://www.youtube.com/watch?v=gfOaMKVdj8I>>. Acesso em: 08 jun. 2018.

**CONTE, Marcelle. Falta de vagas de estacionamento preocupa comerciantes do centro.** 2016. Disponível em: <<http://diariobotucatu.com.br/cidade/falta-de-vagas-de-estacionamento-preocupa-comerciantes-do-centro/>>. Acesso em: 19 maio 2018.

**CONTROLE de Leds - bluetooth - App Inventor 2 - Android #04.** [s.i]: Vitor Santos, 2016. (13 min.), son., color. Disponível em:

<<https://www.youtube.com/watch?v=2g0UiGCansA&t=49s>>. Acesso em: 12 jun. 2018.

**ELÉTRICA, Mundo da. O que é um resistor?** Disponível em:

<<https://www.mundodaeletrica.com.br/o-que-e-um-resistor/>>. Acesso em: 20 jul. 2018.

**ELETRÔNICA DIDÁTICA. PROTOBOARD.** Disponível em:

<<http://www.eletronicadidatica.com.br/protoboard.html>>. Acesso em: 17 jul. 2018.

**FILIPEFLOP. Micro Servo 9g SG90 TowerPro.** Disponível em:

<<https://www.filipeflop.com/produto/micro-servo-9g-sg90-towerpro/>>. Acesso em: 13 jul. 2018.

**GOMES, Tancicleide; MELO, Jeane. App Inventor for Android: Uma Nova Possibilidade para o Ensino de Lógica de Programação.** In: CONGRESSO BRASILEIRO DE INFORMÁTICA NA EDUCAÇÃO, 2., 2013, Recife. **Workshops.** Pernambuco: S.i, 2013. p. 1 - 10. Disponível em: <<http://www.br-ie.org/pub/index.php/wcbie/article/view/2725/2379>>. Acesso em: 03 jul. 2018.

**JORNAL NACIONAL. Motoristas não encontram vagas de estacionamento nas grandes cidades do Brasil:** Segundo especialista, número de veículos cresce mais rápido do que o de vagas, o que eleva preços de estacionamentos. 2011. Disponível em: <<http://g1.globo.com/jornal-nacional/noticia/2011/02/motoristas-nao-encontram-vagas-de-estacionamento-nas-grandes-cidades-do-brasil.html>>. Acesso em: 13 jun. 2018.

**KAWANAMI, Silvia. Estacionamento de Bicicletas Subterrâneo: Eco Cycle – Estacionamento de Bicicletas Subterrâneo.** 2014. Disponível em:

<<https://www.japaoemfoco.com/estacionamento-de-bicicletas-subterraneo/>>. Acesso em: 15 jul. 2018.

LEITE, Joel. **Ruas já têm estacionamento inteligente:** Sistema de indicação de vagas nas ruas reduz emissões e favorece mobilidade. 2017. Disponível em: <<http://automotivebusiness.com.br/artigo/1431/ruas-ja-tem-estacionamento-inteligente>>. Acesso em: 10 jun. 2018.

LULCHEVA, Elena. **BUILDING ANDROID APPLICATIONS WITH MIT APP INVENTOR.** Disponível em: <<https://www.coding-girls.com/blog/building-android-applications-mit-app-inventor>>. Acesso em: 04 jul. 2018.

MATOS, Nuno Miguel Rodrigues de. **Análise do Funcionamento de um Servo motor de Corrente Alternada com Ímãs Permanentes.** 2012. 114 f. Monografia (Especialização) - Curso de Engenharia Elétrica, Universidade do Porto, Blumenau, 2012. Disponível em: <<https://repositorio-aberto.up.pt/bitstream/10216/72684/1/000155343.pdf>>. Acesso em: 20 jul. 2018.

MEIRA, Marcos Cavalcante; LEANDRO, Victor Luiz dos Santos; FREIRE, Carmonizia da Silva. Projeto Vaga Fácil: Uma solução automatizada na viabilização de vagas em estacionamentos públicos e privados. In: CONGRESSO NORTE NORDESTE DE PESQUISA E INOVAÇÃO, 5., 2012, Palmas. **CONNPI.** Tocantins: [s.i], 2012. p. 1 - 5. Disponível em: <<http://propi.ifto.edu.br/ocs/index.php/connpi/vii/paper/viewFile/5094/2589>>. Acesso em: 11 maio 2018.

MÓDULOS para Arduino - Vídeo 08 - Bluetooth HC-06. [s.i]: Brincando Com Ideias, 2016. (36 min.), son., color. Disponível em: <<https://www.youtube.com/watch?v=Zl3IvfNaafA>>. Acesso em: 07 jun. 2018.

MONITOR Serial - bluetooth - App Inventor 2 - Android #03. [s.i]: Vitor Santos, 2016. (10 min.), son., color. Disponível em: <<https://www.youtube.com/watch?v=x1NqRMmsTyM&t=69s>>. Acesso em: 11 jul. 2018.

MOTA, Allan. **Sensor de Luz – Aprendendo a usar o LDR com Arduino.** 2017. Disponível em: <<https://portal.vidadesilicio.com.br/sensor-de-luz-com-ldr/>>. Acesso em: 26 abr. 2018.

MOTA, Allan. **Sensor de Luz – Aprendendo a usar o LDR com Arduino.** 2017. Disponível em: <<https://portal.vidadesilicio.com.br/sensor-de-luz-com-ldr/>>. Acesso em: 17 jul. 2018.

NIKHILBUDHIRAJA. **SMART PARKING USING ARDUINO UNO.** 2017. Disponível em: <<http://www.instructables.com/id/Smart-Parking-Using-Arduino-Uno/>>. Acesso em: 21 maio 2018.

NIKHILBUDHIRAJA. **SMART STREETLIGHT SYSTEM**. 2017. Disponível em: <<http://www.instructables.com/id/Smart-Streetlight-System/>>. Acesso em: 23 jun. 2018.

OLIVEIRA, Euler. **Arduino – Cancela Controlada por RFID**. Disponível em: <<http://blogmasterwalkershop.com.br/arduino/arduino-cancela-controlada-por-rfid/>>. Acesso em: 18 abr. 2018.

PORTAL METALICA. **Empilhando carros: o que são e como funcionam os estacionamentos verticais**. Disponível em: <<http://wwwo.metalica.com.br/o-que-sao-e-como-funcionam-os-estacionamentos-verticais>>. Acesso em: 16 jul. 2018.

PRADA, Rodrigo. **Tecnologia ajuda os motoristas a encontrar uma vaga no shopping**. 2010. Disponível em: <<https://www.tecmundo.com.br/internet/5066-tecnologia-ajuda-os-motoristas-a-encontrar-uma-vaga-no-shopping.htm>>. Acesso em: 04 set. 2018.

PRADO, Thiago Pereira do. **Tinkercad: ferramenta online e gratuita de simulação de circuitos elétricos**. 2018. Disponível em: <<https://www.embarcados.com.br/tinkercad/>>. Acesso em: 08 jul. 2018.

PROCESSING. **Processing**. Disponível em: <<https://processing.org/>>. Acesso em: 12 ago. 2018.

ROBOCORE. **Sensor LDR Arduino**. Disponível em: <<https://www.robocore.net/tutoriais/sensor-ldr-arduino.html>>. Acesso em: 11 maio 2018.

SIGNIFICADOS. **Significado de LED: O que é led**. Disponível em: <<https://www.significados.com.br/led/>>. Acesso em: 19 jul. 2018.

SILVA, Antônio Diermison Rosa da et al. **PROTÓTIPO DE ESTACIONAMENTO INTELIGENTE COM COMUNICAÇÃO SEM FIO**. Teresina: [s.i], 2013. 5 p. Disponível em: <<http://sistemaolimpico.org/midias/uploads/e94ee130799aac20911b1ea832e07930.pdf>>. Acesso em: 03 abr. 2018.

SOUTO, Filipe Vidal; COSTA JUNIOR, Ademar Gonçalves da. **Desenvolvimento de um protótipo de automação predial/residencial utilizando a plataforma de prototipagem eletrônica Arduino**. 2012. 9 f. Monografia (Especialização) - Curso de Engenharia, Centro de Energias Alternativas e Renováveis, Universidade Federal de Paraíba, João Pessoa, 2012. Disponível em: <<http://www.abenge.org.br/cobenge/arquivos/7/artigos/103723.pdf>>. Acesso em: 24 jun. 2018.

SOUZA, Fábio. **Arduino UNO**. 2013. Disponível em:  
<<https://www.embarcados.com.br/arduino-uno/>>. Acesso em: 07 jul. 2018.

THOMSEN, Adilson. **O que é Arduino?** 2014. Disponível em:  
<<https://www.filipeflop.com/blog/o-que-e-arduino/>>. Acesso em: 25 jun. 2018.

# APÊNDICES

## APÊNDICE A - Código do circuito eletrônico do estacionamento

```

#include <LiquidCrystal.h> //bibliotecas do display e do servo
#include <Servo.h>

Servo motor; // declarar o servo
LiquidCrystal lcd(12, 11, 5, 4, 3, 2);

int ledDef_red = 10;
int ledDef_green = 8;
int ledNor_green = 13;
int ledNor_red = A5;
int buzzerPin = A3;
int frequencia = 500;
const int ldrPin1 = A0;
int ldrValor1 =0; //valor lido no ldr1
const int ldrPin2 = A1;
int ldrValor2 = 0;// valor lido no ldr 2
const int ldrServo = A4;
int valorLuminosidade = 500;
int vagas[2] = {0,0}; //inicializa vetor
int i = 0; //contador que ira percorres as vagas
int contador1 = 0; // contador ldr1
int contador2 = 0; //contadorldr2
int vagaslivres = 2;
int vagasocupadas = 0;

void setup()
{
  motor.attach(A2);// fala qual pino esta conectado
  motor.write(0);// posição inicial do servo

  pinMode(ldrPin1, INPUT);
  pinMode(ldrPin2, INPUT);
  pinMode(ledDef_red, OUTPUT);
  pinMode(ledDef_green, OUTPUT);
  pinMode(ledNor_red , OUTPUT);
  pinMode(ledNor_green , OUTPUT);
  pinMode(buzzerPin, OUTPUT);

  Serial.begin(9600);
  lcd.begin(16,2);//inicializar lcd
  lcd.clear(); //Limpa o LCD
}

```



```
int posicao = 0; //armazeza a posicao do servo

String leStringSerial()// Função que lê uma string da Serial e retorna-a
{
    String conteudo = "";
    char caractere;
    // Enquanto receber algo pela serial
    while(Serial.available() > 0)
    {
        // Lê byte da serial
        caractere = Serial.read();
        // Ignora caractere de quebra de linha
        if (caractere != '\n')
        {
            // Concatena valores
            conteudo.concat(caractere);
        }
        // Aguarda buffer serial ler próximo caractere
        delay(10);
    }
    Serial.print("Recebi: ");
    Serial.println(conteudo);

    return conteudo;
}

void loop()
{
    int valorLdr = analogRead(ldrServo); //ler o valor do ldr e armazena na variavel
    Serial.print("valor do ldr servo:");
    Serial.println(valorLdr);
    //se o valor lido for menor ou igual a 500 o servo continua na posicao 0 e espera 1s
    if(valorLdr < valorLuminosidade)
    {
        motor.write(0);
        delay(900);
    }
    if(valorLdr > valorLuminosidade)
    {
        motor.write(90);
        delay(900);
    }
}
```

```
for(i=0; i <= 2; i++) //percore as vagas do estacionamento
{
  // calcula a quantidade de vagas livres e ocupadas
  vagasocupadas = contador1 + contador2;
  vagaslivres = 2 - vagasocupadas;
  lcd.setCursor(0,0);
  lcd.print("Livres = ");
  lcd.print(vagaslivres);
  lcd.setCursor(0,1);
  lcd.print("Ocupadas = ");
  lcd.print(vagasocupadas);
}

//ldr deficiente 1

ldrValor1 = analogRead(ldrPin1);
Serial.print("valor ldr1:");
Serial.println(ldrValor1);

if(ldrValor1 >= valorLuminosidade)
{
  digitalWrite(ledDef_green, HIGH);
  digitalWrite(ledDef_red, LOW);
  contador1 = 0; // 0 LIVRE
}
else
{
  digitalWrite(ledDef_red, HIGH);
  digitalWrite(ledDef_green, LOW);
  contador1 = 1; //1 OCUPADA
}
```

```
// Se receber algo pela serial
if (Serial.available() > 0)
{
  // Lê toda string recebida
  String codigo = leStringSerial();

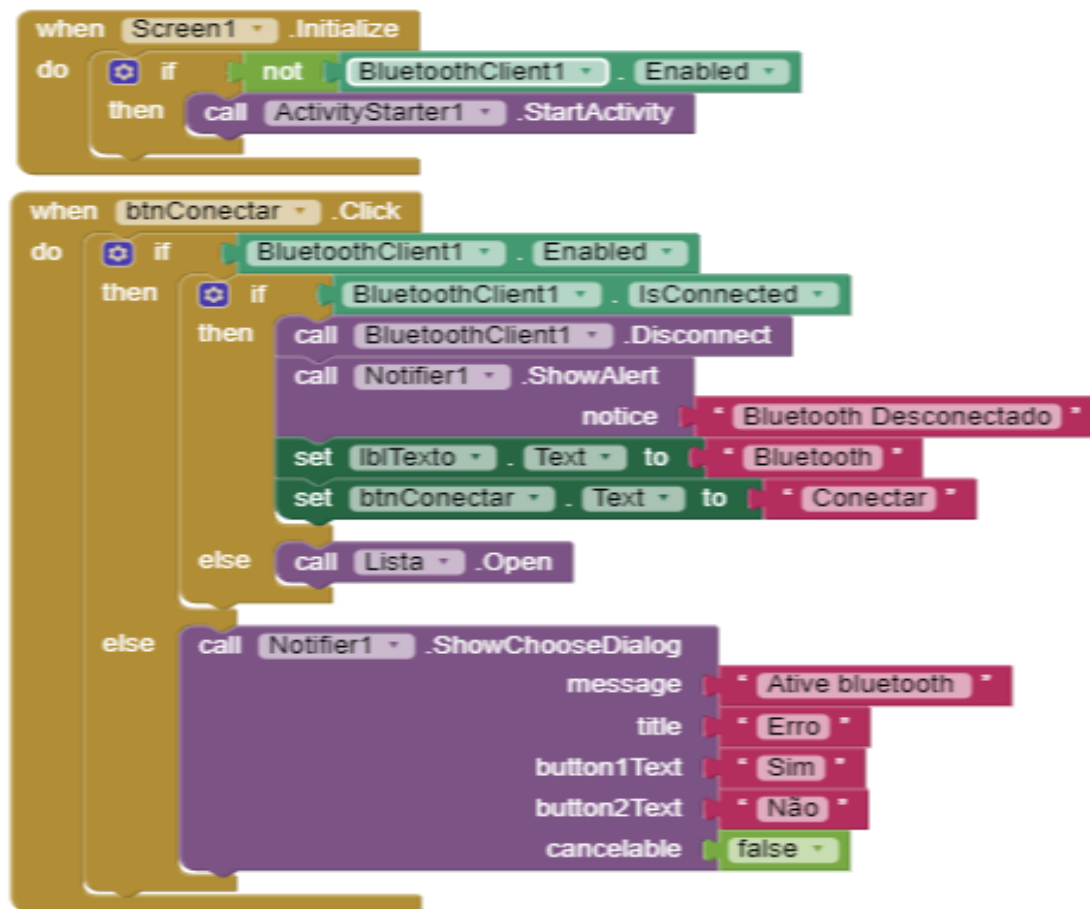
  if (codigo != "Acesso123")
  {
    while(codigo != "Acesso123")
    {
      digitalWrite(ledDef_green, LOW);
      digitalWrite(ledDef_red, HIGH);
      tone (buzzerPin, frequencia);
      delay(50);
      digitalWrite(ledDef_red, LOW);
      tone (buzzerPin, frequencia);
      delay(50);
      digitalWrite(ledDef_red, HIGH);
      tone (buzzerPin, frequencia);
      delay(50);
      tone (buzzerPin, frequencia);
      noTone(buzzerPin); //para a emissao do som por 5s
      delay(100);
    }
  }
}

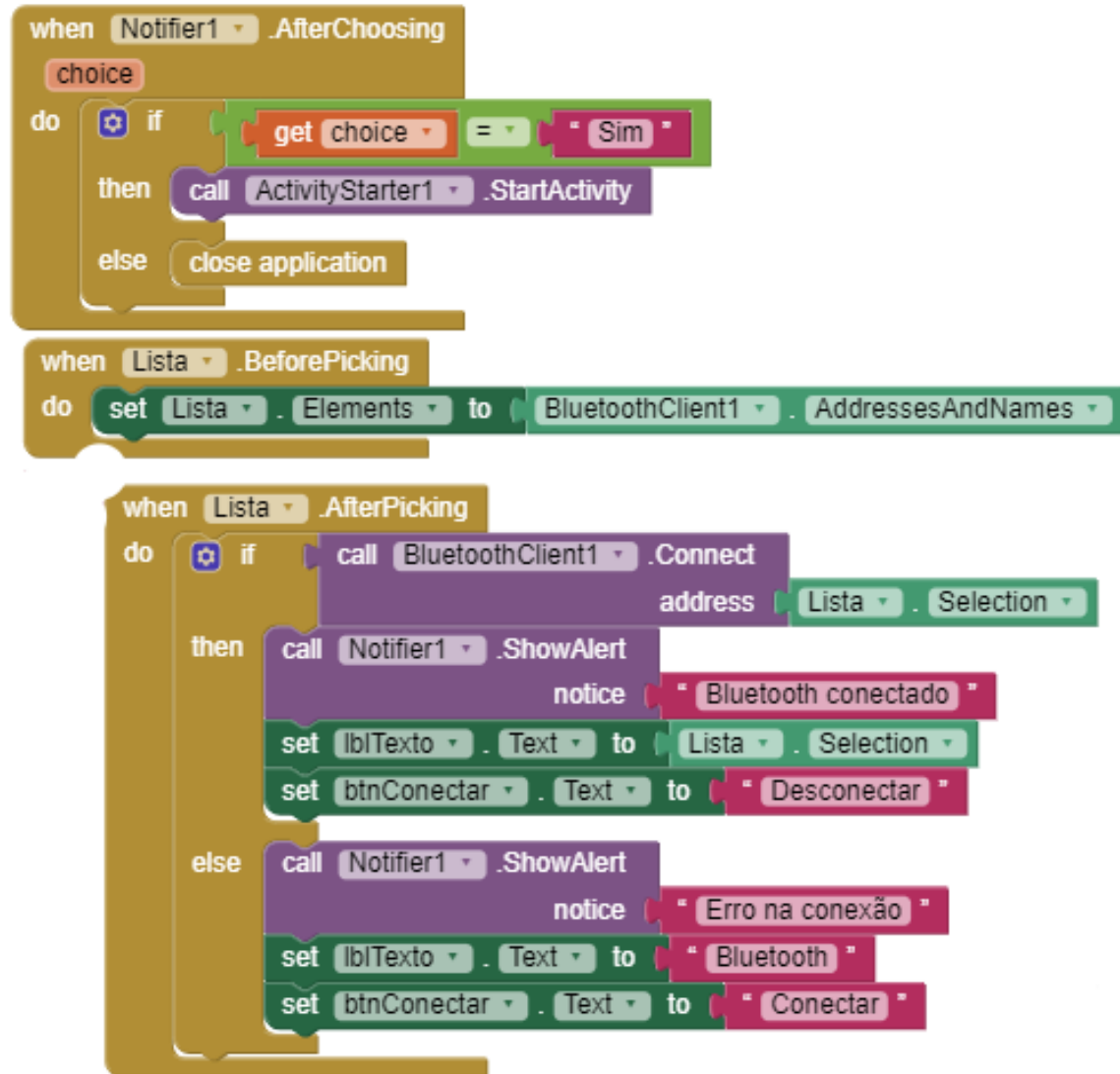
// ldr normal

ldrValor2 = analogRead(ldrPin2);
Serial.print("Valor ldr2:");
Serial.println(ldrValor2);

if(ldrValor2 >= valorLuminosidade)
{
  digitalWrite(ledNor_red,LOW);
  digitalWrite(ledNor_green, HIGH);
  contador2 = 0; // LIVRE
}
else
{
  digitalWrite(ledNor_red, HIGH);
  digitalWrite(ledNor_green, LOW);
  contador2 = 1; //OCUPADA
}
}
```

## APÊNDICE B - Código da conexão via Bluetooth





APÊNDICE C – Desenho esquemático da maquete 3D com as respectivas medidas.

