

# Relatório da Etapa de Desenvolvimento

Fernando F. Dimas<sup>1</sup>, Mercedes Maria B. Diniz<sup>1</sup>

<sup>1</sup> Instituto de Tecnologia - ITEC - Universidade Federal do Pará (UFPA)  
Caixa Postal 479 - 66075-110 - Belém - PA - Brazil

`mercedes.diniz@itec.ufpa.br, fernandooficial0101@outlook.com`

**Abstract.** *This report has evaluative finalists for a curricular component, Electric Circuits, and corresponds to Phase 02 of the Final Laboratory Work. Here we will present the path of system development, including a modeling of the system in the simulator; simulations, observation of results and adjustments.*

**Resumo.** *Este relatório tem fins avaliativos para a componente curricular, Circuitos Elétricos, e corresponde a Fase 02 do Trabalho Final de Laboratório. Aqui apresentaremos o percurso do desenvolvimento do sistema, englobando a modelagem do mesmo no simulador, realização das simulações, observação dos resultados e ajustes.*

## 1. Introdução

Com o encerramento desta segunda fase de desenvolvimento, entregamos aqui o segundo relatório do nosso produto desenvolvido. Seguindo a requisição do investidor, projetamos uma cancela que não necessita de contato físico humano/máquina, ao implementarmos um sistema de gestão com Ticket Virtual.

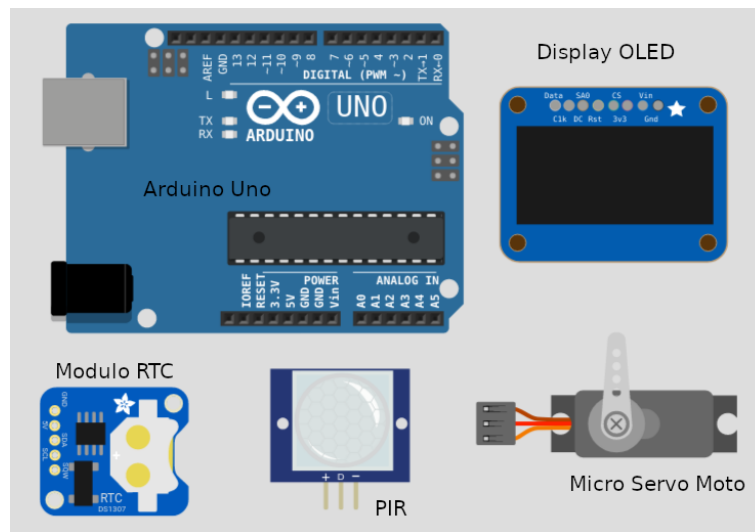
## 2. Características Gerais

O sistema foi criado integralmente em uma plataforma de simulação de placas eletrônicas, conhecida como Wokwi (<https://wokwi.com/>) e disponível gratuitamente. No tópico 3 temos mais detalhes sobre os componentes que foram necessários e que estão presentes nessa ferramenta.

Como mencionado no primeiro relatório entregue, o sistema trata-se de um controle de estacionamento, que irão trabalhar com a emissão e leitura de um QR Code (o

ticket), e detecção automática do veículo. Dado as limitações impostas pela falta de alguns recursos, muito do sistema inicial foi modelado de forma simplificada, assim como a logística do cenário de entrada e saída do estacionamento.

### 3. Componentes Utilizados

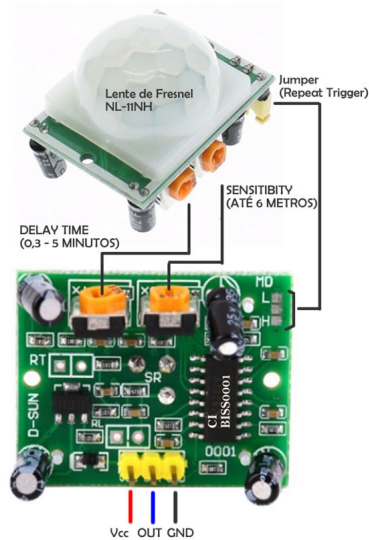


**Figura 1. Componentes Básicos**

#### 3.1. Arduino UNO

O Arduino Uno é uma placa microcontrolada baseada no ATmega328. Possui 14 pinos digitais de entrada/saída (dos quais 6 podem ser usados como saídas PWM), 6 entradas analógicas, um oscilador de cristal de 16 MHz, uma conexão USB, um conector de alimentação, um conector ICSP e um botão de reinicialização. Ele contém tudo que é necessário para apoiar o microcontrolador; simplesmente conecte-o a um computador com um cabo USB ou ligue-o com um cabo adaptador ou bateria AC/DC para começar. O Uno difere de todas as placas anteriores por não usar o FTDI Chip de driver USB para serial. Em vez disso, ele apresenta o Atmega8U2 programado como um conversor USB para serial.

### 3.2. Sensor de Movimento PIR



**Figura 2. PIR**

HC-SR501 é baseado na tecnologia infravermelha, módulo de controle automático, usando design de sonda LHI778 importado da Alemanha, alta sensibilidade, alta confiabilidade, modo de operação de voltagem ultrabaixa, amplamente utilizado em vários equipamentos elétricos de detecção automática, especialmente para produtos controlados automaticamente e alimentados por bateria. O sensor tem três conexões, sendo elas: Um GND (terra), um de voltagem (5V-20V) e um OUTPUT.

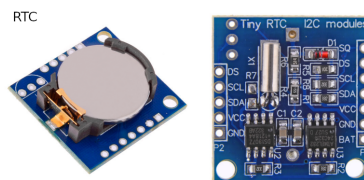
### 3.3. Servo Motor



**Figura 3. Micro Servo Motor**

Pequeno e leve, com alta potência de saída. O servo pode girar aproximadamente 180 graus (90 em cada direção) e funciona como os tipos padrão, mas menor. Você pode usar qualquer código servo, hardware ou biblioteca para controlar estes servos. Bom para iniciantes que querem fazer as coisas se moverem sem construir um controlador do motor com feedback e caixa de engrenagens, especialmente porque caberá em pequenas locais. Ele vem com 3 chifres (braços) e hardware. Ele conta com 3 conexões, sendo elas: Um GND (terra), um de voltagem (5V) e uma entrada de Sinal de controle do servo.

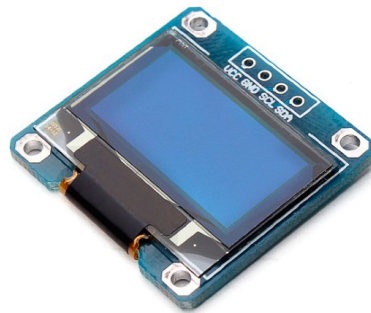
### 3.4. Real Time Clock (RTC)



**Figura 4. RTC**

O relógio serial em tempo real (RTC) DS1307 é um relógio / calendário decimal totalmente codificado em binário (BCD) de baixa potência mais 56 bytes de NV SRAM. Endereço e dados são transferido em série através de um I2C, barramento bidirecional. O relógio / calendário fornece segundos, minutos, horas, informações de dia, data, mês e ano. O fim de data do mês é automaticamente ajustada para meses com menos de 31 dias, incluindo correções para salto ano. O relógio funciona no período de 24 horas ou 12 horas. formato de hora com indicador AM / PM. O DS1307 tem um circuito integrado de detecção de energia que detecta falhas de energia e muda automaticamente para o fornecimento de backup. A operação de cronometragem continua enquanto a parte opera a partir da fonte de backup. Ele conta com 5 conexões, sendo elas: Um GND (terra), um de voltagem 5V, uma SDA que é a linha de entrada da data I2C, uma SCL que dá a linha de entrada da hora I2C e uma SQW que é a saída de onda quadrada (não utilizada nesse projeto).

### 3.5. Display OLED



**Figura 5. Display OLED**

SSD1306 é um driver CMOS OLED/PLED de chip único com controlador para emissão de luz orgânica/polímero sistema de exibição gráfica de matriz de pontos de diodo. Consiste em 128x64 pixels. Este IC é projetado para painel OLED tipo cátodo comum.

O SSD1306 possui controle de contraste, RAM de exibição e oscilador, o que reduz o número de componentes externos e consumo de energia. Possui controle de brilho de 256 etapas. Dados/comandos são enviados do MCU geral através da interface paralela compatível com a série 6800/8000 selecionável por hardware, Interface I2C ou Interface Periférica Serial. É adequado para muitas aplicações portáteis compactas, como sub-display de telefone celular, MP3 player e calculadora, etc. O SSD1306 possui 8 conexões, sendo eles: DATA linha de data I2C (SDA), CLK linha de hora I2C (SCL), 3V3 saída regular de 3.3v, GND (terra), VIN receptor suporte de energia (5V) e as conexões DC, RST e CS que não foram utilizadas no nosso projeto.

## 4. Modelagem do Sistema

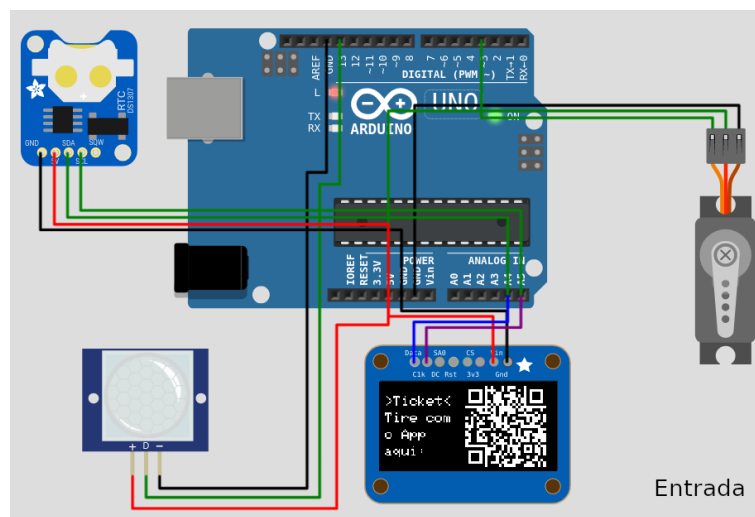
O modelo do nosso projeto é totalmente funcional com as requisições do nosso investidor, dessa forma, explico como vem a funcionar o nosso sistema. Existe um totem de entrada e um de saída, os dois tem um sensor PIR, um RTC e um Servo Motor. Os dois exibem um Ticket (QRcode), entretanto um é para o Totem de entrada e no de saída o Ticket é exibido no monitor serial, porém, esse existe para emular o dispositivo do usuário.

#### 4.1. Circuitos Base

De acordo com a descrição do sistema do estacionamento, o veículo deve ser detectado de forma automática, e com esse objetivo foi implementado o sensor PIR, descrito anteriormente, tanto na entrada como na saída do mesmo. Outro componente presente nesses dois cenários, é o Micro Servo Motor, que representa o cancela na nossa abstração.

Como ilustrado pela imagens abaixo, o circuito da entrada também necessita de um dispositivo de saída de dados para exibir o ticket em formato de QR Code para o usuário. Dessa forma, o display OLED foi selecionado para essa função.

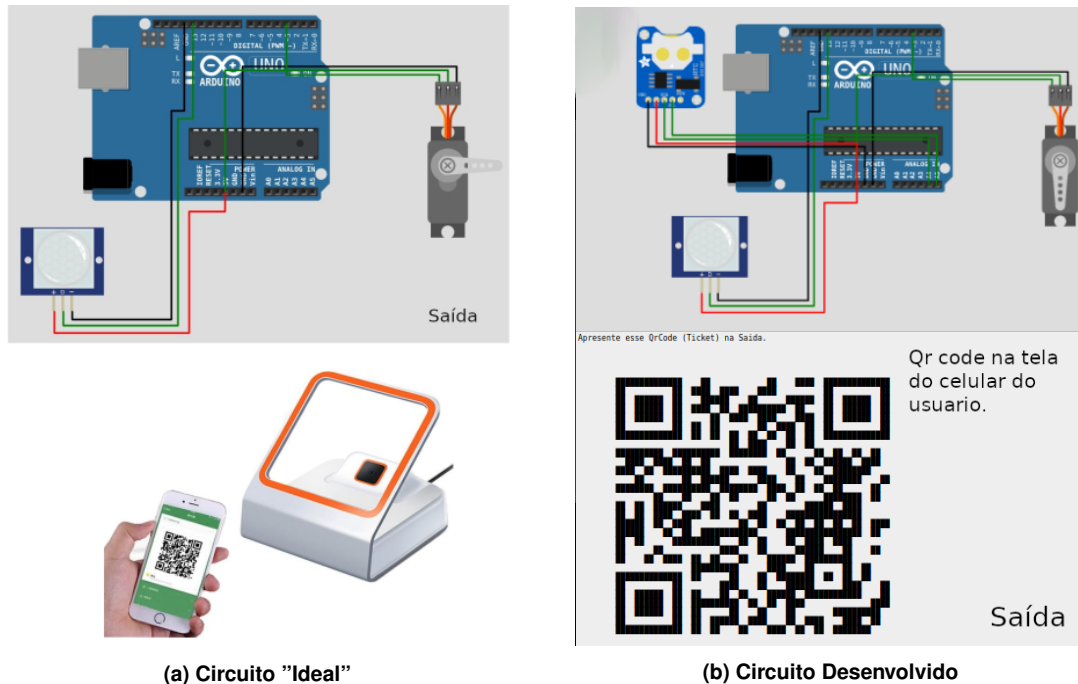
Em uma situação real, a leitura o código teria que ser feita por um aplicativo específico do estabelecimento, para que as informações ficassem registados em um banco de dados, e também através no mesmo o pagamento poderia ser feito. Para o atual protótipo, o ticket gerado nas simulações foi observado através de um leitor de QR Code genérico, logo os dados não são mantido para a etapa da saída.



**Figura 6. Circuito da Entrada do Estacionamento**

Apos a detecção da presença do cliente, e entre a etapa de amostragem do ticket no display do totem, o mesmo deve ser gerado. Para tal, algumas informações básicas precisam está contidas no Qr Code, entre elas a principal é a data e o horário de entrada no estabelecimento. Nativamente, o Arduino, o núcleo do desenvolvimento desse

projeto, não possui nenhum recurso que permita nos dar a data e hora atuais, sendo assim necessário algum recurso externo. O modulo RTC foi o recurso designado para essa função, informações sobre o hardware estão presentes no tópico 3.4, e mais detalhes sobre a implementação do mesmo no projeto será relatado no próximo tópico.



**Figura 7. Circuitos da Saída do Estacionamento**

O cenário da saída do estacionamento é o mais complexo de simular, pois entre o período em que o usuário "leu" o ticket na entrada e, após algum tempo, foi para o sistema da saída, esse primeiro QR Code com os dados do ticket se perdeu. Além disso não possuímos, nessa versão, um leitor no totem para que o motorista apresente o ticket, que estaria no aplicativo anteriormente citado, através desse procedimento seria avaliado se o ticket apresentado é válido, ou em outras palavras, foi pago. E apenas depois disso a cancela seria aberta para que o veículo saísse.

Para contornar tal problemática, optamos por utilizar um valor fixo, tanto para o tempo de permanência quanto para o valor a ser pago. Apenas para ser utilizado de forma lúdica, já que a simulação nos impossibilita a utilização de uma leitura no simulador. Idealizamos a situação para que seja funcional.

## 4.2. Software

Analisando a perspectiva de que o totem de entrada e o de saída são dois circuitos distintos, foram criados dois códigos, desenvolvidos de acordo com a IDE do Arduino, para as diferentes situações.

Devido a utilização de alguns componentes citados, foi necessário dispor de bibliotecas específicas para aplicar determinados recursos. Entre elas, a principal para o desenvolvimento desse projeto foi a **"Qrcode"** do Richard Moore, uma biblioteca simples de geração de código QR. Para lidar com o display OLED e o módulo RTC, foi empregado as bibliotecas do autor Adafruit, **"Adafruit-SSD1306"** e **"RTClib"**, respectivamente. Todas estas reconhecidas pela comunidade e presentes na documentação oficial no site da Arduino.

O código de entrada funciona seguindo o planejamento. Primeiramente definimos os pinos utilizados na placa Arduino e então definimos a altura e a largura da tela OLED, definimos todas as constantes necessárias, como dia, mês e hora. Assim, definimos a mensagem exibida na tela OLED a qual o usuário vai ler com o dispositivo.

Passando para o Setup, fizemos toda a pinagem de entrada da cancela e o teste de função da tela OLED e RTC. em caso de erro exibia-se na tela serial a falha.

Para o Loop, iniciamos com a detecção do veículo, passando uma leitura alta e movendo a cancela (Servo Motor). Então partimos para a obtenção da data e hora através do RTC e transformamos int para char. Logo concatenamos os dados obtido e formamos o que vai ser exibido no ticket, por fim, exibimos o QRcode na tela. Caso não tenha nenhuma detecção de veículo, define-se o sinal de entrada do PIR para baixo e abaixa a cancela.

Ainda na entrada, se inicia uma função para a exibição das informações na tela OLED, geramos um ponto de partida para o texto e um ponto para a exibição do QRcode. finalizamos com uma função de concatenação.

Para a saída, fizemos a reutilização de alguns código da entrada. utilizamos a

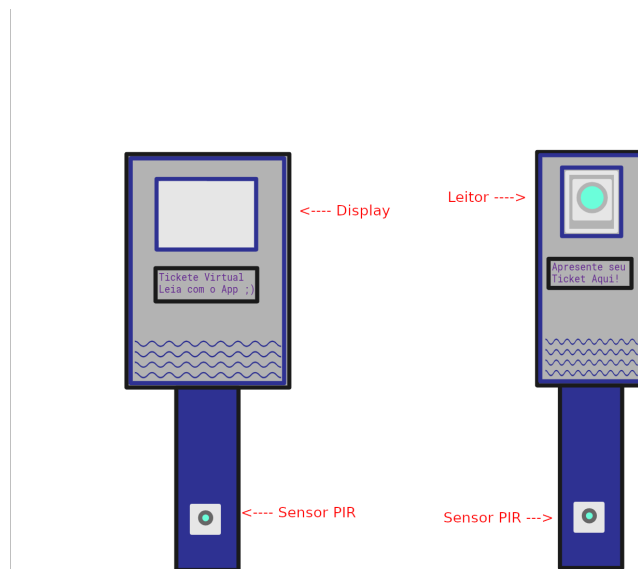


mesma pinagem menos a utilização do tela OLED. Adicionamos uma linha de texto que vai ser exibida no monitor serial (a qual emula o dispositivo do usuário) e gerenciamos o que vai ser exibido ao ler o QRcode do usuário, o qual contem o valor pago, o tempo de permanência e a hora de saída.

No Setup, assim como no código de entrada, fizemos a pinagem de entrada da cancela e teste de RTC e em caso de erro exibe-se na tela serial a falha.

Em Loop, faz-se a detecção do carro, e se existe um, a cancela permanece abaixada até ser feita a leitura. Passamos para a obtenção da data e hora atual (que é a hora de saída), bem como a transformação de int para char. Concatenamos a mensagem a ser exibida ao ler o QRcode e encerramos exibindo uma mensagem, o QRcode no monitor serial e levantando a cancela.

#### 4.3. Perspectiva do Produto Final



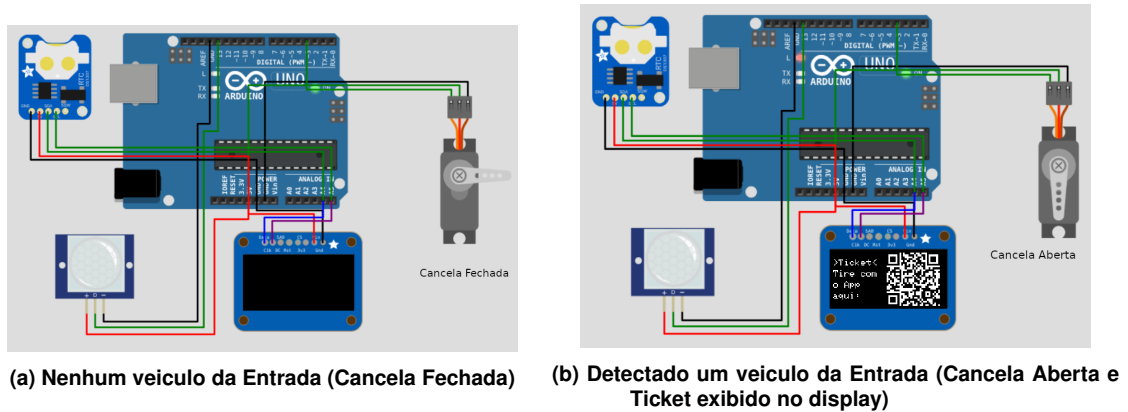
**Figura 8. Idealização do produto final**

### 5. Resultados das Simulações

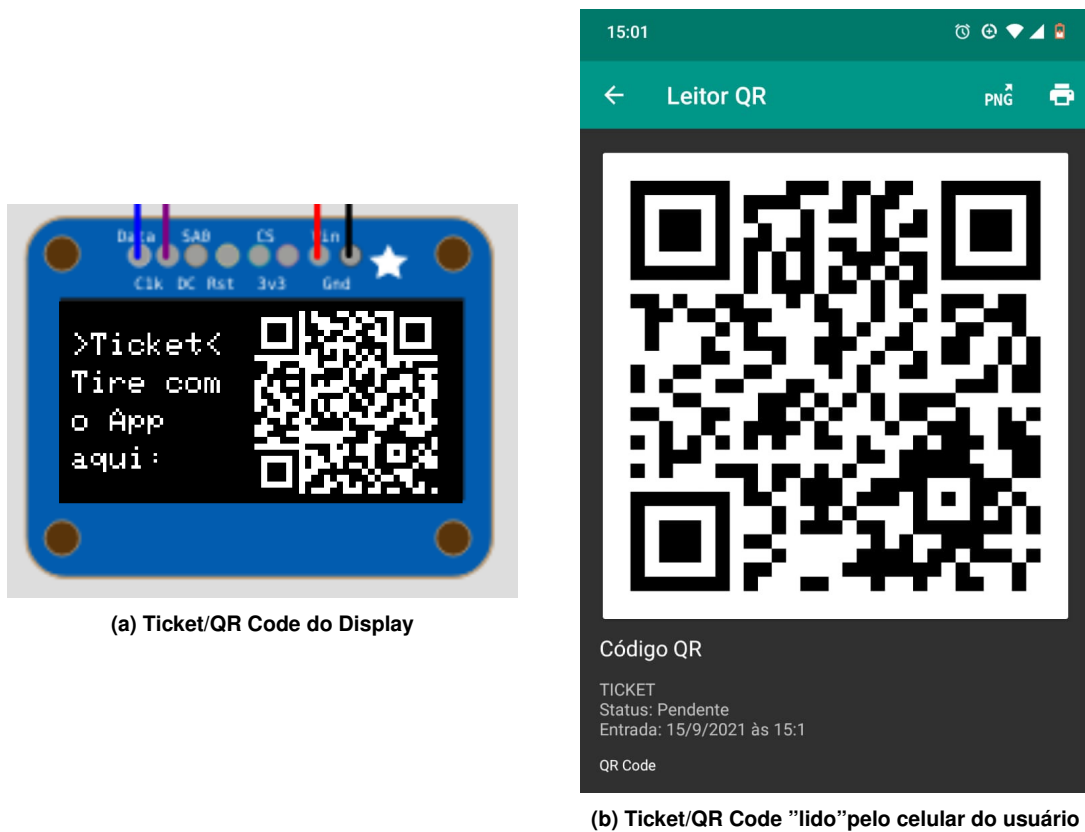
Em nossas simulações iniciais percebemos que havia grande dificuldade em exibir o QRcode na tela OLED, entretanto, ao utilizarmos a função de forma correta conseguimos utilizar a tela e também o monitor serial para exibição do QRcode.

A cancela também foi complicada de alinhar com a leitura do QRcode mas utilizando o sensor PIR conseguimos manter a cancela no devido lugar. Mas no fim das simulações conseguimos alcançar os objetivos almejados, tanto com o QRcode quanto com a Cancela.

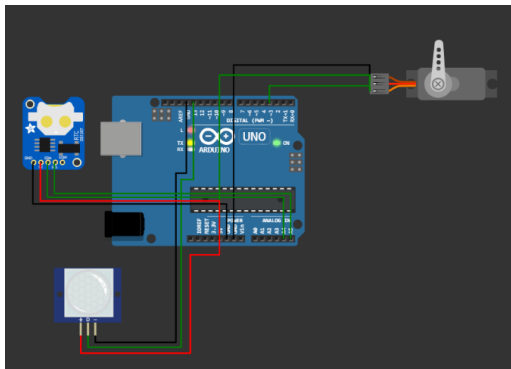
Observe nas imagens abaixo os resultados obtidos nas simulações:



**Figura 9. Simulação da Entrada de um Veículo**



**Figura 10. Emulação do Ticket na Entrada**



(a) Cancela aberta após leitura do Ticket na saída



(b) Ticket/QR Code exibido pelo dispositivo do usuário

**Figura 11. Simulação da Saída de um Veículo**

## 6. Conclusão

Este relatório tem o objetivo de informar ao nosso investidor (o professor) como segue o desenvolvimento do nosso trabalho, com base no que foi descrito, temos fechado os ajustes e o desenvolvimento do primeiro protótipo do produto mínimo viável. Seguimos em dia com o projeto, avançando na pesquisa e na produção técnica sem muitos problemas.

## 7. Referencias

Referência para os dispositivos utilizados:

**Arduino UNO:** Arduino-A000066-Datasheet

**Sensor de Movimento PIR:** PIR-HC-SR501-Datasheet

**Servo Motor:** SG90-TowerPro-Datasheet

**Real Time Clock RTC:** DS1307-Maxin Integrated-I2C-Datasheet

**Display OLED:** SSD1306-SolomonSystech-Datasheet