

Logotipo

Descrição gerada automaticamente

Sistema de cashback e pagamento para estacionamentos privados,

aliado ao uso da automação.

Junho de 2023

**Florianópolis - SC**



**17/12/2015**

Florianópolis/SC

**SUMÁRIO**

[1. INTRODUÇÃO 3](#_heading=h.tyjcwt)

[2. DESENVOLVIMENTO 4](#_heading=h.30j0zll)

[2.1. CÓDIGO FONTE 5](#_heading=h.1fob9te)

[2.1.1 BACK-END 5](#_heading=h.1fob9te)

[2.1.2 FRONT-END 5](#_heading=h.1fob9te)

[2.2. SENSOR 6](#_heading=h.1fob9te)

[2.3. ATUADOR 8](#_heading=h.1fob9te)

[3. RESULTADOS 9](#_heading=h.3znysh7)

[4. REFERÊNCIAS 12](#_heading=h.2et92p0)

**1. INTRODUÇÃO**

Construímos um aplicativo que conecta os estacionamentos aos usuários, dessa forma você é capaz de encontrar o local mais próximo para estacionar com segurança.

Nossa interface é simples e propõe com poucos cliques encontrar o local, realizar o pagamento diretamente pelo aplicativo e ainda aproveitar o tempo não utilizado como um cashback de horas em toda a rede credenciada.

A parte de hardware do projeto simulado uma cancela de estacionamento e é constituída por uma Placa NodeMCU ESP8266 (onde o código construído será responsável por ler e identificar a entrada e saída do usuário ao estacionamento), um sensor RFID (responsável por acionar presencialmente a leitura do cartão do usuário via tag ou NFC com o smartphone) e dois leds (vermelho e verde, que após a leitura do sensor, alternar a cor dando assim a confirmação de entrada e saída ao usuário).

**2. DESENVOLVIMENTO**

O sistema da Ignite Park & Go propõe um layout simples e intuitivo onde ocorrerá toda a parte de interação do usuário com o sistema. Nela é possível realizar o cadastro, verificar a rede de estacionamentos credenciados e acessar a carteira onde o usuário pode adicionar saldo, conferir o extrato de transações e a sessão de cashback adquirido.

Nossa missão é integrar os estacionamentos privados e torná-los mais acessíveis para os usuários. Uma vez adicionado o valor a carteira do aplicativo, o usuário tem o poder de utilizar o saldo em toda a rede credenciada. Que por sua vez, terão o poder de utilizar o saldo investido nestes lugares de forma mais democrática através do sistema de “cashback”. Esta é a grande inovação proposta por nós.

E como funciona na prática o sistema de cashback? Da seguinte forma, vamos supor que o estabelecimento tem como regra o valor de 15 reais por três horas de permanência, porém o usuário utilizou apenas uma hora, nossa startup propõe um retorno em cashback de horas na proporção de 25% do horário não utilizado pelo usuário retornado ao banco de horas no aplicativo.

Outro uso proposto é o pagamento do estacionamento sem a necessidade pagar fisicamente em guichês ou tótens, com o Ignite Park & Go é possível pagar através do aplicativo com o saldo existente.

Num primeiro momento a identificação do usuário no estacionamento se dará através de um cartão atrelado a tecnologia RFID por meio de um sensor instalado. Este fará a leitura de entrada e saída do usuário registrando assim o tempo de uso. Futuramente a ideia é substituir o sensor do sistema RFID por um sensor LPR que fará a leitura de entrada e saída do usuário através das placas dos carros com uma câmera.

Neste semestre utilizaremos como back end do projeto o sistema da “Tago IO” para fazer a comunicação dos sensores com o sistema. Desde a leitura do cartão do usuário, até o acionamento de abertura da cancela e a contagem de carros utilizando o nosso aplicativo.

**2.1. CÓDIGO FONTE**

O projeto foi desenvolvido em 3 pilares, back-end, front-end e o protocolo de comunicação para hardware MQTT.

**2.1.1 BACK-END**

Nossa back-end hoje faz toda a comunicação com o protocolo MQTT para capturar os dados que o hardware fornece e para enviar alguns dados necessários para o hardware atuar.

O sistema está sendo desenvolvimento em Java, utilizando o framework Spring Boot, o sistema por debaixo dos panos que faz todo o controle de registro de um novo cliente, de um novo empresário que quer credenciar o seu estacionamento dentro do nosso aplicativo e monitoramento do seu estacionamento em tempo real por uma dashboard dentro do sistema, podendo utilizar via desktop para uma melhor visualização das métricas que o sistema fornece.

**2.1.2 FRONT-END**

Na parte do cliente (o aplicativo) fornecemos uma interface simples que na primeira página após logado tem um mapa capturando sua geolocalização e fornecendo pontos no mapa dos estacionamentos que são credenciados pelo nosso sistema, assim para fornecer uma melhor agilidade e uma comodidade maior no seu dia a dia.

O sistema é autossuficiente, pode tanto se fazer a parte de pagamento pelo próprio aplicativo, como o cliente pode optar por pagar no local estacionado. Mas vale ressaltar que pagando pelo aplicativo tem suas vantagens

O pagamento feito pelo aplicativo é registrado e podendo assim receber um bônus por ter estacionado e não utilizado todo o tempo fornecido pelo estabelecimento, sendo assim o cliente tem um retorno em % de minutos ou horas de volta (cashback) para dentro do seu aplicativo, assim podendo utilizá-lo em outro estacionamento na rede credenciada em nosso aplicativo.

**2.2. SENSOR**

O sensor utilizado por nós é o RFID-RC522 baseado no chip MFRC522 da empresa NXP é altamente utilizado em comunicação sem contato a uma frequência de 13,56MHz. Este chip, de baixo consumo e pequeno tamanho, permite, sem contato, ler e escrever em cartões que seguem o padrão Mifare.

**Ligação:**

|  |  |
| --- | --- |
| **ESP32** | **MFRC522** |
| G21 | SDA |
| G18 | SCK |
| G23 | MOSI |
| G19 | MISO |
| GND | GND |
| G22 | RST |
| 3V | 3.3V |

**Funcionamento:**

* 3.3V – Pino de alimentação do módulo (2,5 a 4V).
* RST – Pino reset que serve para colocar o RC522 no modo de baixo consumo (“power-down”).
* Nível alto = Funcionamento normal.
* Nível baixo = Modo “power-down”.
* GND – Terra da alimentação.
* MISO – Pino de entrada do mestre da comunicação SPI.
* A interface I2C também pode utilizar este pino como SCL.
* MOSI – Pino de saída do mestre da comunicação SPI.
* SCK – Pino de clock da comunicação SPI.
* SDA – Pino SS ou NSS da comunicação SPI.
* A interface I2C também pode utilizar este pino como SDA.

**Especificações:**

* Corrente de trabalho: 13-26mA / DC 3.3V;
* Corrente ociosa: 10-13mA / 3.3V;
* Corrente Slep: < 80uA - Pico de corrente: < 30Ma;
* Freqüência de operação: 13,56MHz;
* Tipos de cartões suportados: Mifare1 S50, S70 Mifare1, Mifare UltraLight, Mifare pro, Mifare Desfire;
* Temperatura de operação: -20 a 80 graus Celsius;
* Temperatura ambiente: -40 a 85 graus Celsius;
* Umidade relativa: 5% - 95%;
* Parâmetro de Interface SPI;
* Taxa de transferência: 10 Mbit/s;
* Dimensões: 8,5 x 5,5 x 1,0cm;
* Peso: 21g.

**2.3. ATUADOR**

O atuador utilizado são os leds vermelho e verde, é um componente diferenciado e de extrema funcionalidade.

**Ligação:**

|  |  |  |
| --- | --- | --- |
| **Cor do LED** | **Tensão em Volts (V)** | **Corrente em Miliamperes (mA)** |
| Vermelho | 1,8V – 2,0V | 20 mA |
| Verde | 2,0V – 2,5V | 20 mA |

**Funcionamento:**

* Pino negativo do led conecta no negativo (GND da placa);
* Pino positivo do led é conectado em um resistor e o mesmo em uma das entradas analógicas da placa;

**Resistores:**

Linha do tempo

Descrição gerada automaticamente

**3. RESULTADOS**

**IMAGEM 1: Dashboard Tago IO.**

**Tabela

Descrição gerada automaticamente**

Assim é a dashboard da plataforma Tago IO, podemos observar o funcionamento do sistema em ação. Quando a tag de número 73 é lida e reconhecida, o status da cancela se altera e abre para o acesso do usuário ao estacionamento.

**IMAGEM 1.2: Dashboard Tago IO.**

**Tabela

Descrição gerada automaticamente**

O exemplo contrário, a tag de número 72 é lida pelo sistema e não é reconhecida. O status da cancela não altera e permanece fechada.

**IMAGEM 2: Hardware Montado.**

Cabo de equipamento eletrônico

Descrição gerada automaticamente com confiança média

Nessa imagem estamos apresentando o nosso hardware montado, utilizando os seguintes componentes:

* Controlador ESP-WROOM-32;
* Sensor RFID – RC522;
* 2 Capacitores (330 Ohms / cada);
* 1 Led Vermelho;
* 1 Led Verde.

**IMAGEM 3: Sensor RFID-RC522.**

**Uma imagem contendo eletrônico, circuito, azul

Descrição gerada automaticamente**

Sensor RFID é responsável por fazer a leitura da tag/cartão do cliente e passar ao sistema que fará ou não a identificação do mesmo.

**IMAGEM 4: Controladora ESP32.**

**Circuito eletrônico com fios

Descrição gerada automaticamente com confiança média**

O controlador ESP32, instruído com o código fonte para realizar todo o controle de entrada e saída do estacionamento.

**4. REFERÊNCIAS**

* CORP, Arana. Using an RFID module with an ESP8266. Aranacorp. 2021. Disponível em: https://www.aranacorp.com/en/using-an-rfid-module-with-an-esp8266/. Acesso em: 06 abr. 2023.
* E CIA, Arduino. Como usar o NodeMCU ESP8266 com display Oled I2C. Arduino e Cia. 2018. Disponível em: https://www.arduinoecia.com.br/nodemcu-esp8266-com-display-oled-i2c/. Acesso em: 06 abr. 2023.
* ELETROGATE, Blog. Componentes Básicos do Arduino – O que é Resistor, LED, Potenciômetro, Push Button. Eletrogate. 2020. Disponível em: https://blog.eletrogate.com/componentes-basicos-do-arduino-o-que-e-resistor-led-potenciometro-push-button Acesso em: 12 jun. 2023.
* LOCATELLI, Carolina. Componentes Básicos do Arduino – O que é Resistor, LED, Potenciômetro, Push Button. Blog Curto Circuito. 2022. Disponível em: https://curtocircuito.com.br/blog/eletronica-basica/o-que-e-um-led Acesso em: 12 jun. 2023.
* GEEKSFORGEEKS. ESP32 - Introduction, Pinout, Specifications & Programming. Disponível em: https://www.geeksforgeeks.org/esp32-introduction-pinout-specifications-programming/. Acesso em: 15 jun. 2023