



UNIVERSIDADE FEDERAL DO CEARÁ
CENTRO DE TECNOLOGIA
DETI



Monitoramento de Acesso de Veículos

Fortaleza, 2017

SUMÁRIO

1. INTRODUÇÃO

2. IMPLEMENTAÇÃO

3. RESULTADOS E DISCUSSÕES

4. CONCLUSÃO

5. REFERÊNCIA

APÊNDICE

1 INTRODUÇÃO

Em resposta aos problemas de segurança, a universidade começou monitorar o acesso de alguns veículos ao campus. O presente trabalho tem como objetivo verificar horário de entrada e saída de veículos e saber quais estão circulando no campus, de maneira automatizada e rápida. O sistema foi pensado para utilização no campus do Pici - UFC, mas pode, com poucas adaptações, ser utilizado em qualquer estacionamento que tenha controle de acesso, como os de empresas.

O sistema possui dois modos de funcionamento:

Para um veículo de indivíduo com vínculo ao campus: O indivíduo deve ter uma Tag RFID. Ao chegar ao campus, o leitor RFID irá identificar a Tag, a cancela abrirá, a raspberry irá salvar no servidor horário de entrada do veículo.

Para um veículo de indivíduo sem vínculo ao campus: O indivíduo, ao parar na frente da cancela, será notado através dos sensores ultrassônicos. Uma vez que é identificado que existe um veículo na entrada, a câmera é ativada, captura uma imagem e a cancela é liberada. A imagem capturada é enviada para o servidor.

Os dados salvos no servidor podem ser acessados por uma API.

2 IMPLEMENTAÇÃO

2.1 MATERIAIS UTILIZADOS

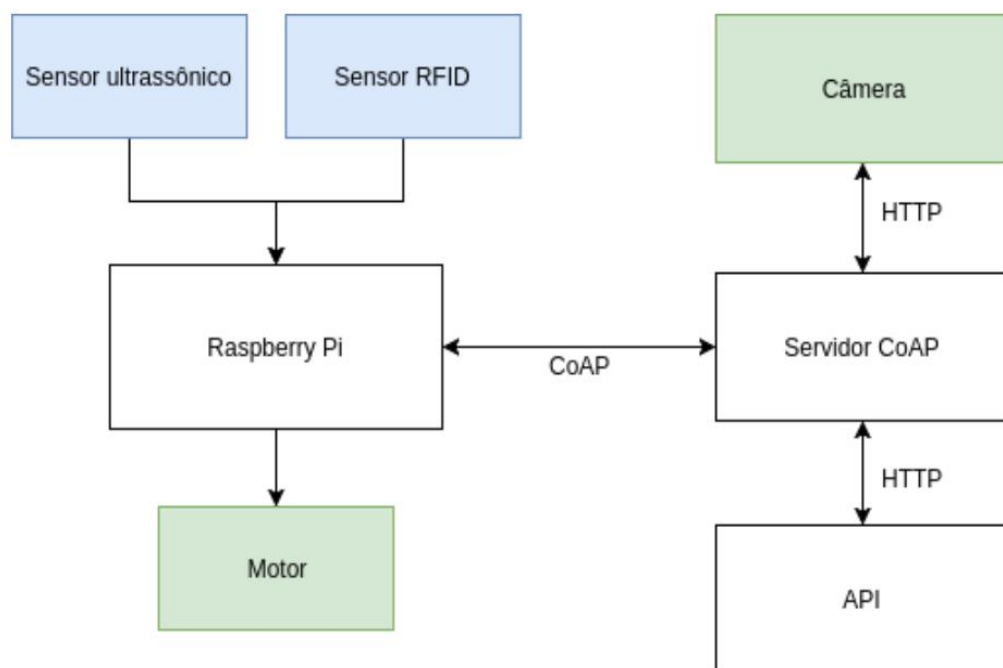
- Tag RFID x N*
- Leitor de RFID (RC522) x 1
- Sensor ultrassônico (HC-RS04) x 2
- Motor servo (9g Tower Pro SG90) x 1
- Raspberry x 1
- Host x 1

*Depende do número de usuários

2.2 PROCEDIMENTO

Foi utilizada uma Raspberry Pi 3 model B para controlar os sensores ultrassônicos, o sensor RFID e o (atuador) motor servo. A Raspberry deve se manter na mesma rede que o servidor CoAP, e esse deve ter acesso a (atuador) câmera IP e à API, como mostra a Figura 1.

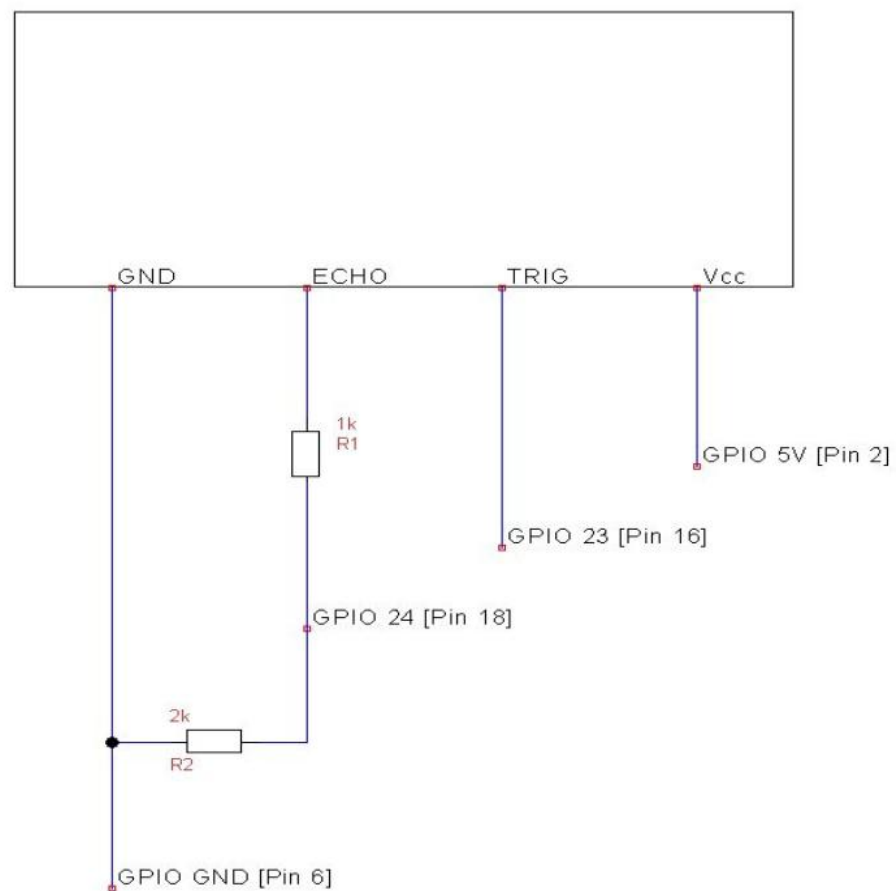
Figura 1 - Arquitetura.



Para o controle dos sensores ultrassônicos e do motor servo, foi utilizada a biblioteca piio (<https://github.com/leitel/piio>), que nós desenvolvemos, e para o RFID, uma vez que a comunicação via SPI foi habilitada na Raspberry Pi, foi utilizada a biblioteca MFRC522-python e para criar instância do servidor CoAP e realizar a comunicação foi utilizada a biblioteca CoAPthon.

Apenas para os sensores ultrassônicos foi necessário criar um circuito divisor de tensão, devido a diferença entre a tensão de saída (5 v) do mesmo pelo pino echo com o aceito como entrada na Raspberry Pi 3 model B (3.3 V). O circuito está ilustrado na Figura 2. Os demais sensores e atuadores necessitaram apenas de ligação direta com alimentação e pinos de controle. O esquemático completo está exposto no Apêndice.

Figura 2 - Circuito para sensores ultrassônicos.

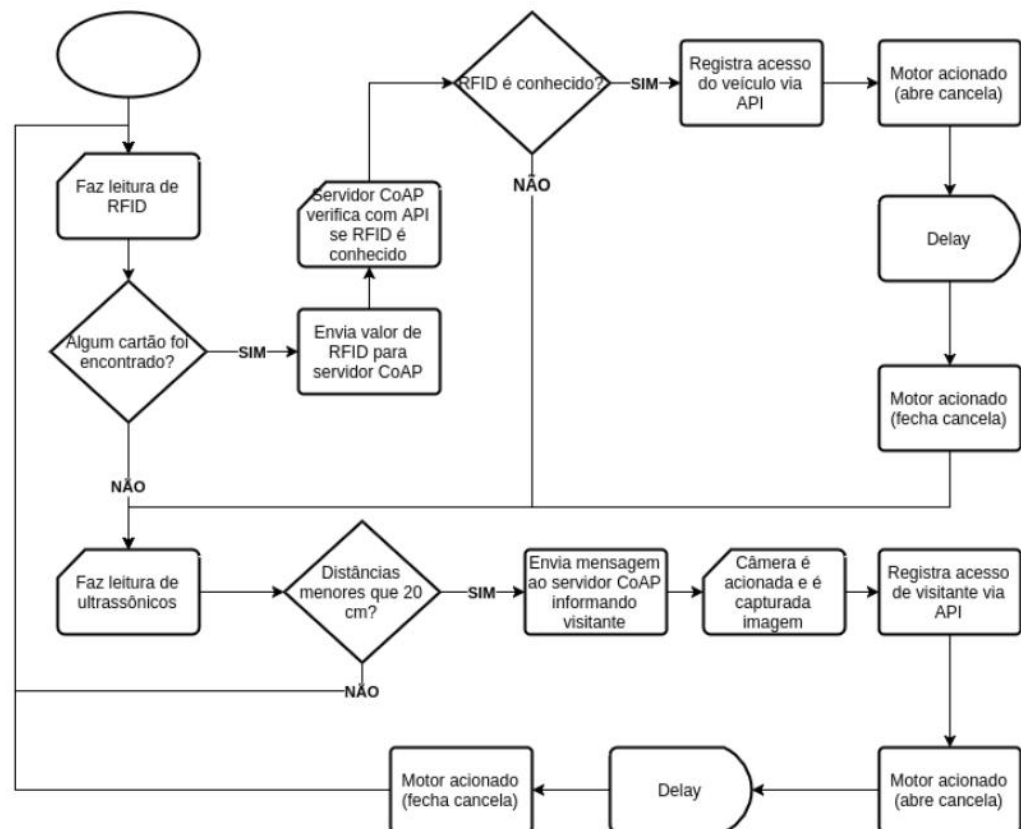


A ideia inicial era, para o protótipo, utilizar um kinect como câmera, mas ele mostrou alguns desafios e acabou sendo substituído. A câmera utilizada é a de um smartphone Android, que permite que o aparelho funcione como uma câmera IP

com o uso do aplicativo IP Webcam, disponível na Play Store. A API Rest foi produzida com o framework Adonisjs. Para os testes e protótipo, as instâncias do servidor CoAP e da API foram criadas em um único host, mas esse não é um requisito.

Com o sistema montado, para o funcionamento devem ser iniciadas a API, o servidor CoAP e o script controlador do nó (Raspberry). O sistema deve funcionar assim como mostra o diagrama da Figura 3.

Figura 3 - Diagrama de funcionamento.



-
1. O script da Raspberry verifica se existe alguma tag RFID
 - a) **Se existir** tag RFID é enviada uma mensagem via CoAP ao servidor com o valor do RFID lido.
 - b) O servidor CoAP faz uma requisição HTTP à API, que contém informações sobre os RFIDs cadastrados
 - c) Se o RFID for conhecido é cadastrado o acesso do veículo e enviada uma confirmação ao servidor CoAP.
 - d) O servidor responde à Raspberry. Se for uma confirmação de RFID conhecido, ela aciona o motor e abre a cancela. Após 3 segundos (tempo utilizado no protótipo) a cancela é fechada.
 2. O script da Raspberry verifica se os dois ultrassônicos possuem algum objeto à distância menor que 20 cm (distância utilizada no protótipo).
 - a) **Se existir** objetos à distância menor que 20 cm, é enviada uma mensagem via CoAP ao servidor requisitando que seja capturada uma imagem.
 - b) O servidor CoAP faz uma requisição HTTP à câmera IP, que captura a imagem e envia para o mesmo.
 - c) O servidor, com imagem capturada, faz uma requisição HTTP à API, informando data e nome da imagem para registrar o acesso de um visitante.
 - d) Quando a Raspberry recebe mensagem do servidor CoAP, ela aciona o motor e abre a cancela. Após 3 segundos a cancela é fechada.

3 RESULTADOS E DISCUSSÕES

3.1 DESAFIOS

Um dos primeiros desafios foi a utilização do aparelho da Microsoft, o Kinect. Por ser um dispositivo com documentação fechada, não foi possível implementá-lo no sistema, mesmo utilizando códigos de terceiros, a comunicação com os demais dispositivos do sistema não foi realizada. Logo, foi necessário buscar uma alternativa que se encaixasse na idéia do projeto. O aplicativo IP Webcam, permite que um smartphone funcione como uma câmera IP e foi escolhido para o projeto.

O projeto inicial contava com a utilização de dois nós sensores, um para entrada e um para saída. Também era planejado o uso de sensores para verificar o momento em que o veículo passou pela cancela. Entretanto, por falta de recursos, só foram implantados sensores para a detecção da entrada de veículos. Para resolver o problema de saber quando o carro passou pela catraca, foi configurado um tempo limite para fechar a cancela automaticamente.

3.2 RESULTADOS

Quando é colocado uma tag RFID não cadastrada no sistema, o leitor RFID envia os dados para o servidor que verifica se o cartão está cadastrado. Como está sendo utilizado um cartão não cadastrado, o servidor envia o sinal para que a cancela não libere de imediato o acesso ao veículo.

Em seguida, neste cenário em que o cartão de identificação do usuário não está cadastrado no sistema, o veículo será detectado pelos sensores ultrassônicos, que enviam um sinal para o servidor confirmando a presença do veículo. Por fim, o servidor faz download de uma foto, tirada pela câmera IP, do veículo e depois armazena-a no seu banco de dados junto com os dados de entrada. Feito isso, o acesso é liberado.

Utilizando uma tag registrada, o leitor RFID realiza a leitura e envia para o servidor para realizar a verificação. Como a tag está registrada no sistema, o servidor envia a liberação imediata para a cancela e registra data de acesso do veículo, sem precisar utilizar os sensores ultrassônicos e a câmera IP.

4 CONCLUSÃO

Mesmo com a limitação dos recursos o sistema permite o monitoramento de acesso de veículos. O sistema de controle utilizando o leitor RFID para usuários cadastrados e a junção da câmera IP com os sensores ultrassônicos para usuários externos se mostrou bastante eficiente, de uma forma que o banco de dados terá as informações de entrada de todos os veículos, por meio do código RFID ou da foto.

Para ser utilizado em escala real, seria interessante substituir o leitor e todas RFID de baixa frequência para alta, que faz leituras mais rápidas e a maior distância. Para diminuir o volume de dados no servidor, realizar processamento de imagens, extraíndo apenas a placa do carro (convertida para texto ASCII), seria uma questão fundamental. Desta forma, o projeto apresenta uma solução eficiente e barata para o controle de acesso na UFC, assim como em outros lugares em que exista um fluxo constante de veículos e necessidade de monitoramento.

5 REFERÊNCIA

1. Sensor ultrassônico, <http://www.micropik.com/PDF/HCSR04.pdf>, acessado em Dezembro de 2017.
2. Servo motor, <http://www.micropik.com/PDF/SG90Servo.pdf>, acessado em Dezembro de 2017.
3. Configurar Raspberry para RC522, <https://pimylifeup.com/raspberry-pi-rfid-rc522/>, acessado em Dezembro de 2017.
4. MFRC522-python, <https://github.com/mxgxw/MFRC522-python>, acessado em Dezembro de 2017.
5. CoAPthon, <https://github.com/Tanganelli/CoAPthon>, acessado em Dezembro de 2017.
6. Adonisjs, <https://github.com/adonisjs/adonis-framework>, acessado em Dezembro de 2017.
7. RaspberryPi, <https://www.raspberrypi.org/>, acessado em Dezembro de 2017.
8. Tutorial com Sysfs (kernel linux) para controle GPIO de Raspberry, <http://hertaville.com/introduction-to-accessing-the-raspberry-pis-gpio-in-c.html>, acessado em Dezembro de 2017.

APÊNDICE

Esquemático do nó sensor: Raspberry Pi 3, com sensores ultrassônicos, leitor RFID e servo motor.

