

# Estacionamento Inteligente para Edifícios Comerciais

Iago de Andrade Sousa  
Universidade de Brasília - UnB  
Brasília, Brasil  
16/0124701  
iago.andradef@gmail.com

Híterson de Oliveira Silva  
Universidade de Brasília - UnB  
Brasília, Brasil  
13/0114227  
hiterfga@gmail.com

Lorena Albernaz Pinheiro  
Universidade de Brasília - UnB  
Brasília, Brasil  
14/0025715  
lorena.albernazz@gmail.com

**Resumo**—O projeto visa implementar em um Raspberry Pi um estacionamento inteligente para um edifício comercial privado visando a comodidade e segurança dos usuários que trabalham neste edifício.

**Index Terms**—sistemas embarcados; estacionamento; raspberry Pi; edifício.

## I. INTRODUÇÃO

Edifícios comerciais que possuem estacionamento privado para seus funcionários necessitam de um sistema de controle de entrada e saída de usuários. De acordo com a Pesquisa Distrital por Amostra de Domicílios realizada pela CODEPLAN, quarenta por cento dos brasilienses utilizam carro para se locomoverem até o local de trabalho. [1]

Além disso estima-se que quase sessenta por cento dos centros comerciais possuem estacionamento próprio e que a maioria dos usuários o utilizam de acordo com a quantidade de vagas disponíveis. [2]

Como o estacionamento é um componente notável do transporte urbano, é fundamental para Políticas de Mobilidade Urbana estabelecer níveis de oferta para o usuário. [3] Sendo assim, um sistema embarcado é uma possibilidade para elaborar uma solução para o controle de acesso de estacionamentos.

## II. JUSTIFICATIVA

O projeto visa a implementação de um sistema capaz de melhorar a qualidade da experiência do usuário de estacionamentos privados visando a segurança. Além de realizar o controle do estacionamento automaticamente com a entrada e saída do edifício apenas com o cadastro da placa do usuário.

## III. OBJETIVOS

O objetivo principal deste trabalho é a implementação de um sistema de automação para estacionamento inteligente capaz de utilizar uma câmera para reconhecimento da placa do carro do usuário utilizando um Raspberry Pi 3.

Os objetivos específicos são:

- Reconhecer a placa do carro do usuário para ativação da cancela;
- Cadastrar novos usuários;
- Utilizar o protocolo de comunicação TCP/IP;

## IV. REQUISITOS

Foram levantados os seguintes requisitos para implementar as funcionalidades do sistema:

### A. Controle de Entradas:

- A câmera deve permanecer em standby até ser acionada por um botão pelo motorista do veículo na entrada do estacionamento;
- Após reconhecimento do usuário, a cancela deve ser acionada por meio de uma string enviada por protocolo TCP/IP;
- Após a passagem do veículo, a cancela deve ser fechada.

### B. Cadastro de Usuários

- O usuário deverá solicitar ao administrador do local para ser feito um cadastro com as suas informações antes da utilização do sistema;
- O sistema de cadastro será feito pelo Telegram exclusivamente pelo administrador do local e o aplicativo deve requerer do usuário os seguintes dados para cadastro: nome completo, apartamento, placa, modelo e ano do veículo.

## V. BENEFÍCIOS

A utilização de algoritmos para reconhecimento de placas de carro é extremamente útil para controlar a entrada e saída de pessoas autorizadas em estacionamentos de forma automatizada e segura, além de monitorar o tempo de permanência no local. Nesse sentido, a utilização da Raspberry Pi é justificada pelo seu alto poder de processamento, no qual permite rápida execução de algoritmos de processamento de sinais.

Esse sistema é mais prático e mais seguro do que os convencionais sistemas que utilizam cartões, visto que esses cartões podem ser roubados, desmagnetizados ou até mesmo clonados.

## VI. MATERIAIS

Os materiais que serão utilizados para implementação do presente projeto são:

- 01- Raspberry Pi 3 model B;
- 01- Câmera;
- 01- Computador para simular a cancela eletrônica;

- 01- Botão
- 01- Resistor 1K  $\Omega$
- 01- Protoboard

## VII. DESENVOLVIMENTO

Para entender cada processo do sistema, foi elaborado um diagrama de blocos mostrado na Fig.1. O diagrama foi desenvolvido para auxiliar a modulação do projeto de forma que cada módulo possa ser testado individualmente para validação de seu funcionamento. Para elaborar a solução, foi considerado que o cadastro de usuários deve ser feito previamente e que o motorista do veículo deve pressionar um botão para iniciar o processo.

Conforme mostrado no diagrama de blocos, após o início do sistema, será ligada a câmera que fará a aquisição da imagem da placa veicular. O programa deve processar a imagem adquirida e realizar o reconhecimento dos caracteres da placa do veículo. O dado da placa será verificado no banco de dados, se o usuário já for cadastrado, o sistema aciona a abertura da cancela. Caso o usuário não seja cadastrado, o acesso ao estacionamento será negado.

### A. Aquisição da Imagem

Para adquirir a imagem, será posicionada uma câmera com o intuito de capturar a imagem da placa veicular. Após o usuário parar na entrada do estacionamento e pressionar o botão, o sistema se inicia a imagem é capturada.

O botão em questão é acionado pelos pinos GPIO da Raspberry, enquanto a captura da imagem será por meio de comandos da biblioteca fswebcam. Essa biblioteca é capaz de realizar a autoconfiguração relativa à qualidade e dimensão de qualquer imagem obtida por meio de uma entrada USB da Raspberry.

Para a identificação da placa veicular, foi escolhida a biblioteca de código aberto OpenALPR. Esta biblioteca é executada localmente e tem como saída os caracteres da placa no modelo ABC1234, três letras seguidas por quatro números.

A biblioteca retona as principais possibilidades de placa, mas para comparação no banco de dados é utilizada apenas a opção com mais confiabilidade.

### B. Comunicação entre o Telegram Bot e a raspberry Pi

O Telegram Bot com Raspberry Pi é um sistema que permite envio de comandos através de mensagens instantâneas do software Telegram. Com este software é possível realizar o acionamento de relés, LEDs e monitorar ambientes, em qualquer lugar do mundo a qualquer hora.

Nesta parte do projeto será testado a seguinte situação, o usuário mandará suas informações para o síndico do local ou administrador. Ele ficará encarregado de realizar o cadastro da pessoa e da placa do seu veículo no banco de dados. Com as informações o síndico realizará o cadastro do usuário no Telegram bot do edifício comercial, administrador do edifício enviará uma mensagem para o usuário que ele foi cadastrado. A realização deste procedimento é uma forma para segurança do sistema.

Após o cadastro quando o usuário apertar o botão para que a foto da placa do carro seja reconhecida pelo sistema, há o acionamento do destravamento da cancela.

Os comandos para o cadastro do usuário no telegram bot pelo administrador são:

- start;
- 123456789 (essa foi a senha utilizada como exemplo);
- morador (para registrar um novo morador);
- veiculo (para registrar um novo veículo sem precisar adicionar um morador);

O nome do Telegram bot para o estabelecimento foi denominado "Estacio\_Comercial". Para iniciar o cadastro deve-se acionar o comando " senha ". Após isso o administrador recebe as orientações do Telegram bot para o registro dos dados da pessoa. Após a confirmação do registro, os dados serão arquivados em um arquivo de formato .txt com o nome do usuário. Os dados serão inseridos automaticamente no banco de dados criado e posteriormente o usuário terá livre acesso ao edifício.

A implementação do Telegram bot foi realizada na linguagem de programação C++ seguindo os passos listados abaixo:

- Selecionou-se o BotFather do Telegram;
- Criou-se um novo Bot com o nome "Estacio\_Comercial";
- Adquiriu o código para acessar a HTTP API para controlar o Bot denominando Token;
- Escreveu-se o código denominado Telegram.cpp com os procedimentos em geral para cadastro;
- Compilou o código na Raspberry Pi no terminal com o acesso a internet;
- Funcionamento instâneo do programa 24h;

Considerando a implementação citada acima, o administrador terá acesso ao bot para somente após inserir o comando de senha correto. Com a confirmação dessa senha, o administrador poderá realizar o cadastro de um novo usuário conforme mostrado na Figura 2.

### C. Banco de dados

A criação de um banco de dados para o projeto é necessária, pois a cancela do estacionamento deverá abrir apenas se a placa reconhecida pela câmera estiver no registro dos veículos do estabelecimento, sendo que esses veículos são vinculados aos cadastros dos usuários. Para essa finalidade, foi utilizado o MariaDB server, que é um servidor utilizado para armazenamento e manipulação de banco de dados baseado no MySQL server.

Dessa forma, MariaDB é um sistema de gestão de base de dados relacional (mais conhecido por sua sigla em inglês RDBMS) gratuito e de código aberto que responde a pedidos (queries) feitos pela linguagem SQL. Isso significa que o servidor MariaDB armazena dados em forma de tabelas, as quais podem ser criadas, excluídas e modificadas por meio da linguagem SQL.

O MariaDB server é bem eficiente e é utilizado por grandes empresas para realizar tarefas extremamente complexas. Como exemplo de empresas que utilizam essa ferramenta, pode-se

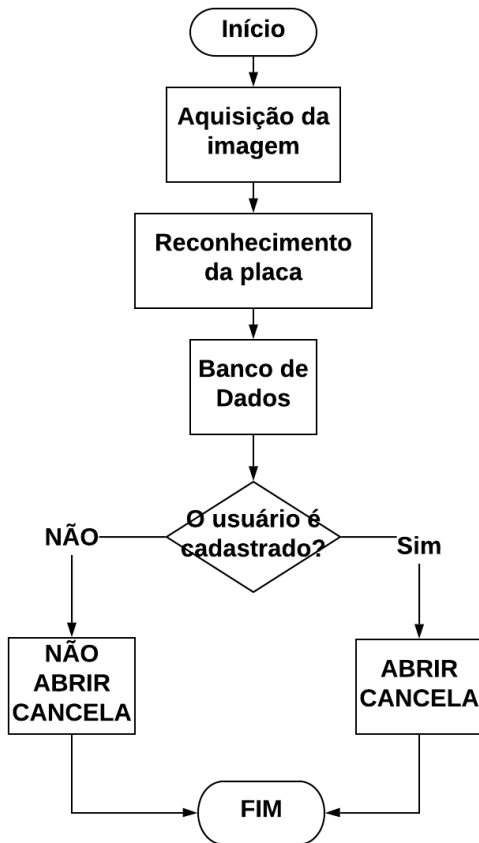


Figura 1. Diagrama de blocos do sistema.

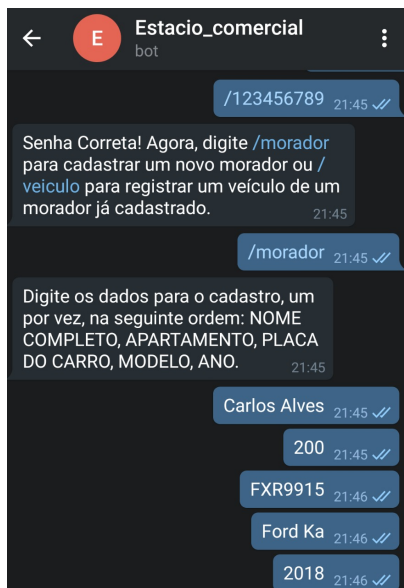


Figura 2. Interface Telegram Bot realizando cadastro pelo administrador.

citar: ServiceNow, DBS Bank, Google, Mozilla e Wikimedia Foundation[4].

Como usuários podem ter mais de um veículo registrado, foi necessário criar duas tabelas: uma com alguns dados dos moradores; e outra tabela com os dados dos veículos. Na segunda tabela são considerados os seguintes dados dos veículos: placa, modelo, ano e proprietário. A coluna “proprietário” está vinculada à primeira tabela e, dessa forma, é possível inserir mais de um veículo para cada pessoa, bastando identificar o mesmo proprietário para todos os veículos desse usuário na última coluna.

O MariaDB server foi implementado na Raspberry Pi 3 B. Dessa forma, a Raspberry armazena todo o banco de dados na sua memória e permite que os dados sejam acessados pelo programa de controle da cancela quando assim necessário.

#### D. Código para Controle da Cancela

O código que controla o acesso ao estacionamento foi implementado na linguagem de programação C e está em execução na Raspberry Pi 3 B. Por meio dela é possível controlar a câmera para tirar foto da placa do carro por meio de um botão e, depois disso, a Raspberry faz um processamento de imagens para conseguir identificar a placa do carro na foto. Por fim, é feita uma comparação entre a placa obtida pela câmera e as placas dentro do banco de dados. Caso a placa identificada pela foto seja igual a alguma das placas dentro do banco de dados, a Raspberry manda um sinal “ABRIR” e, caso o programa não encontre nenhuma placa correspondente no banco de dados, a Raspberry manda um sinal “NAO ABRIR”. Esses sinais são utilizados para comunicar à cancela eletrônica a ação a ser feita: abrir ou permanecer fechada. A comunicação entre a Raspberry e a cancela eletrônica é feita por meio do protocolo TCP/IP em rede local, sendo que a cancela funciona como o servidor que aguarda passivamente por pedidos de algum cliente e a Raspberry é o cliente que faz os pedidos à cancela.

Primeiramente, o código configura os pinos da Raspberry por meio da biblioteca Wiring Pi e coloca o pino GPIO 4 como um pino de entrada. Esse pino identifica o momento em que o botão é pressionado: ele é conectado com um resistor de pull up, ou seja, a entrada fica em nível lógico alto quando o botão não é pressionado e, quando pressionado, ela vai para nível lógico baixo. Então, o programa abre o socket para o cliente e se conecta ao servidor, estabelecendo, assim, a comunicação por TCP/IP.

Quando o pino GPIO 4 identifica nível lógico baixo, a Raspberry manda um sinal para a câmera tirar uma foto e salvar em um arquivo jpg. Essa câmera é conectada a uma das portas USB da Raspberry e seu acionamento é feito por meio da biblioteca fswebcam[5]. Depois que a foto é salva em um arquivo chamado “teste.jpg”, ela é aberta por meio da biblioteca openalpr[6] e essa mesma biblioteca faz o reconhecimento da placa do carro que está no arquivo jpg. Quando o reconhecimento é feito, a biblioteca retorna algumas opções de placas possíveis em ordem decrescente de chance de acerto. Essas opções são salvas em um arquivo “placa.txt”

e todas as placas dentro desse arquivo são salvas em um vetor de strings.

Então, o programa manda um pedido para o banco de dados procurar pela primeira placa obtida no arquivo “placa.txt” (a qual apresenta a maior chance de estar correta) e o resultado dessa busca é salvo em um arquivo denominado “answer.txt” e em uma string. Por fim, o programa compara a primeira placa obtida no arquivo “placa.txt” com essa string que contém o resultado da busca no banco de dados. Caso as duas strings sejam iguais, o programa mostra a placa correspondente ao resultado do pedido ao banco de dados e envia a mensagem “ABRIR” para o servidor (cancela). Por outro lado, caso as strings sejam diferentes, o programa envia a mensagem “NAO ABRIR”. Depois disso, o programa desconecta o socket criado, zera o vetor de placas obtidas anteriormente, estabelece uma nova conexão com o servidor e fica esperando o botão ser pressionado novamente, pois o código fica em um loop infinito permitindo, assim, que outras pessoas sejam capazes de acessar o estacionamento.

### E. Resultados

A seguir, é mostrado o resultado da execução do código para controle da cancela. A foto da Figura 3 foi tirada de uma imagem de placa de carro do Google Imagens com a câmera utilizada para o projeto. Como mostrado na Figura 3, o programa retorna 6 possibilidades de placas e a primeira, a qual apresenta maior chance de acerto, é a que está correta. Por esse motivo, aparece a mensagem “ABRIR”. Essa mensagem é mostrada no terminal e também é enviada para o servidor.

Por fim, a Figura 4 mostra o servidor recebendo dados da Raspberry. O servidor simulando a cancela na verdade é um computador e os resultados são exibidos no terminal, assim como os indicativos de conexão com a Raspberry.

## VIII. CONCLUSÃO

É possível perceber, a partir dos resultados obtidos, que o projeto final atendeu os requisitos propostos: o reconhecimento das placas de veículos funcionou, assim como a integração dos códigos em C e C++ com o banco de dados e também a comunicação por protocolo TCP/IP entre a Raspberry Pi e a cancela, a qual foi simulada por um computador se comportando como um servidor.

Porém, ainda há alguns problemas que precisam ser resolvidos caso o projeto venha a ser utilizado de forma comercial. Seria necessária, por exemplo, uma câmera melhor, pois a qualidade das imagens influencia muito na precisão do openalpr. Talvez também seria necessário um programa de reconhecimento de placas mais preciso. Outro problema encontrado é que tanto o programa de cadastro pelo telegram quanto o programa de controle da cancela eletrônica se encerram caso haja falha de conexão e, portanto, precisaria de alguma forma impedir que os programas se encerrem ou voltar a iniciá-los automaticamente caso eles fechem.

Para trabalhos futuros para o aprimoramento do trabalho proposto temos:



Figura 3. Foto de uma placa de carro tirada pela câmera utilizada no projeto e o resultado da execução do código para reconhecer a foto da placa e fazer o pedido à cancela.

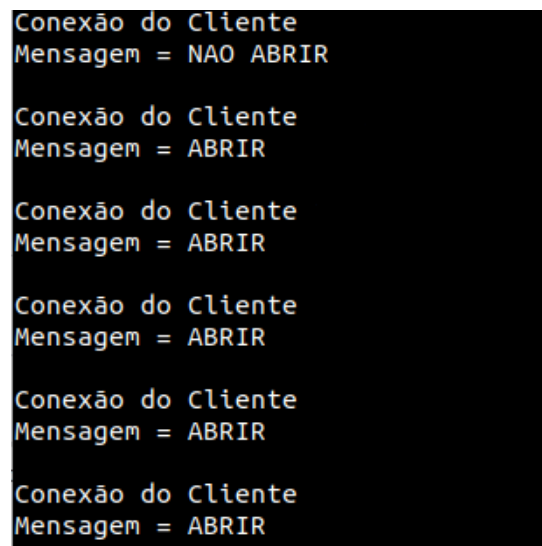


Figura 4. Servidor recebendo dados da Raspberry Pi.

- Melhorar o programa de processamento de imagens da placa, ou mesmo comprar um comercial;
- Testar o programa com uma cancela real;
- Realizar um programa de falha de conexão;
- Realizar um programa de segurança para não haver vazamento de informações dos usuários;

## REFERÊNCIAS

- [1] CODEPLAN. Pesquisa Distrital por Amostra de Domicílios. Brasília, DF. Data: 2018. Disponível em: [www.codeplan.df.gov.br/pdad/](http://www.codeplan.df.gov.br/pdad/). Acesso: 30/08/2019.
- [2] FECOMÉRCIO. Pesquisa de Levantamento e Percepções sobre o Estacionamento de Veículos. Campo Grande, MS. Data: 07/2012.
- [3] LEI 12.587/12. Lei da Mobilidade Urbana. Brasil. Data: 03/01/2012. Disponível em: <http://www.planalto.gov.br/ccivil03/ato2011-2014/2012/lei/l12587.htm>. Acesso: 30/08/2019.
- [4] MariaDB. Disponível em <https://en.wikipedia.org/wiki/MariaDB>. Acesso em 30/09/2019.
- [5] Using a standard USB webcam. Disponível em <https://www.raspberrypi.org/documentation/usage/webcams/>. Acesso em 01/11/2019.
- [6] openalpr. Disponível em <https://github.com/openalpr/openalpr>. Acesso em 01/11/2019.