

# Reconhecimento automático de placas para controle de acesso veicular a condomínios

PC4

Arthur Torres Magalhães – 15/0006063

Universidade de Brasília - UnB  
Brasília-DF - Brasil  
arthurtorres26@outlook.com

Jovelino Caetano Braz Junior – 14/0043641

Universidade de Brasília - UnB  
Brasília-DF - Brasil  
jovelinocbjunior@gmail.com

**Resumo** — Este trabalho propõe a criação de um protótipo, com o uso de conceitos de sistemas operacionais embarcados e do microcontrolador Raspberry Pi 3, para a implementação de um reconhecedor automático de placas veiculares para controle de acesso em condomínios com custo muito menor em relação a outros produtos com a mesma finalidade.

**Palavras chaves** – *Raspberry Pi 3; reconhecimento de placas; controle de acessos; condomínios.*

## I. JUSTIFICATIVA

É de conhecimento geral que as áreas urbanas se adensam cada vez mais e com isso houve um aumento no número e no tamanho dos condomínios. Dessa forma, o mercado de condomínios se tornou um mercado em ascensão no país nos últimos anos. De acordo com a Associação Brasileira de Síndicos e Síndicos Profissionais (ABRASSP), existem mais de 400 mil condomínios no país, sendo que estes movimentam um montante de mais de 165 bilhões de reais por ano [1].

Aliado ao crescimento no número de condomínios no país, surgiram também as dificuldades no gerenciamento dos moradores desses condomínios, tal como as dificuldades relacionadas a expansão na prestação de serviços e dos gastos envolvidos. Um problema que se destaca é o controle ao acesso de veículos nos condomínios. Tal controle é necessário e se torna uma ferramenta importante no auxílio a segurança dos condôminos.

Sem um controle eficaz de quem entra ou sai do condomínio, a ação de criminosos ou até mesmo prestadores de serviços (indesejados) é facilitada. É comum ver notícias sobre quadrilhas que invadem condomínios se passando por parentes de moradores ou prestadores de serviço. Portanto, implantar um sistema de registro e cadastro das entradas e saídas de veículos facilita a manutenção da segurança e consequentemente propicia aos moradores mais conforto e privacidade. É importante ressaltar que, apesar de não garantir que o condomínio não será vítima de roubos, um sistema que realize esse controle permite a identificação dos criminosos e facilita futuras investigações.

Com o avanço da tecnologia, surgiram diversas formas de realizar esse controle no acesso de veículos, como: controle por biometria, controle por TAGs (muito vista em pedágios, com o uso de aplicativos como ConectCar e Sem Parar, por exemplo), controle por RFID (controle feito por rádio frequência), por reconhecimento facial e por fim, o controle de

acesso através do reconhecimento de placas veiculares [2]. Além disso, é interessante que os condomínios estabeleçam um esquema de identificação das pessoas que adentram o condomínio, em conjunto com o registro de entrada e saída dos veículos, afim de obter uma maior eficácia no controle.

Sendo assim, o trabalho em questão busca a implementação de um sistema de controle de acesso veicular através do reconhecimento de placas, com a construção de um protótipo. Este tipo de controle foi escolhido como tema deste trabalho por abranger as especificações da disciplina e permitir o aprendizado da matéria, com o auxílio da Raspberry PI.

## II. OBJETIVO

Elaboração de um projeto com protótipo de reconhecimento automático de placas veiculares em condomínios que, de acordo com um banco de dados contendo informações acerca dos carros dos moradores, seja capaz de distinguir entre moradores e visitantes, além de criar um registro para controle de toda entrada e saída do condomínio.

## III. REQUISITOS

O sistema deve:

A. *Detectar e identificar corretamente a placa;*

O sistema deverá ser capaz de identificar a placa e obter a numeração da mesma de forma correta e rápida.

B. *Ter uma câmera com resolução suficientemente boa para o propósito do projeto;*

Para que o sistema funcione corretamente, a imagem a ser processada precisa ter uma boa qualidade, portanto a câmera utilizada deve ter uma resolução apropriada.

C. *Identificar se o veículo está registrado como sendo de um morador ou não;*

Após a identificação dos caracteres da placa, será necessário acessar um banco de dados verificando se o veículo pertence a um morador ou não.

D. Fazer o registro da entrada contendo número da placa, data e hora que o veículo entrou/saiu do condomínio, assim como uma foto da placa e do motorista;

O registro serve para arquivar a entrada dos veículos a fim de ter acesso a informação, que possa ser usada posteriormente, seja por motivos de segurança ou outros. Além disso, deve possibilitar a identificação não apenas da placa, mas também do motorista do veículo, afim de manter o registro de quem entra no condomínio.

E. Possibilitar o cadastro de novos veículos no sistema;

O condomínio deve ser capaz de adicionar e remover placas (moradores) do sistema.

F. Estar devidamente conectado a algum tipo de estrutura que o dê suporte e proteção;

O sistema deve estar encapsulado de forma a estar protegido contra sol, chuva, vento e os demais fatores que o possam pôr em risco. Além disso, a estrutura deve ser compatível para caber todos os elementos necessários do sistema.

G. Possuir uma fonte de energia que dure tempo suficiente.

O sistema deve ficar ligado, se possível, por 24hrs, ou pelo menos pela quantidade de tempo em que é permitida a entrada de pessoas no condomínio.

#### IV. TABELA DE MATERIAIS

Tabela 1. Materiais usados no projeto

Quantidade	Equipamento	Marca
1	Raspberry PI 3 Model B	Raspberry
1	AN-VC500	LG
1	Fonte de tensão 5V/2A	-

#### V. HARDWARE E SOFTWARE

O hardware do projeto envolve basicamente dois componentes: uma Raspberry Pi 3 e uma camera Raspberry Pi.

A Raspberry Pi 3 Model B+ (figura 1) foi escolhida, dentre os vários modelos de Raspberry, por ser o modelo mais recente, possuir 4 entradas USB, conexão Wi-Fi (2,4 GHz e 5 GHz), entradas Ethernet, Bluetooth e HDMI. Enfim, é o mais completo dentre todos os modelos, e abrange completamente o escopo do projeto.

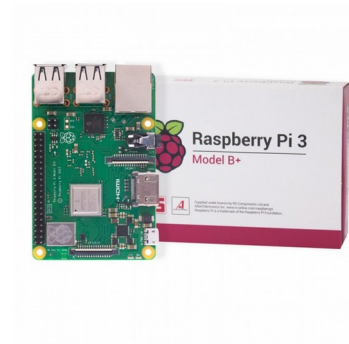


Figura 1. Raspberry Pi 3 Model B+

A câmera escolhida para o projeto inicialmente foi a câmera própria da Raspberry Pi, devido a qualidade de imagem que ela proporciona e pelo pouco espaço que ela ocupa, o que facilitaria na instalação da estrutura final.

Entretanto, após algumas falhas identificadas na câmera da Raspberry Pi relacionados ao mal funcionamento da mesma e tendo em vista os prazos de entrega do trabalho e os custos envolvidos, decidiu-se usar a câmera da marca LG, modelo AN-VC500.

A câmera é capaz de tirar fotos no formato MJPEG, com 1920x1080 pixels de resolução, o que para o presente projeto é satisfatório e atende as necessidades.



2.WebCam utilizada (AN-VC500)

Figura

Já o software do projeto é baseado em linguagem C e funciona de acordo com o fluxograma a seguir.

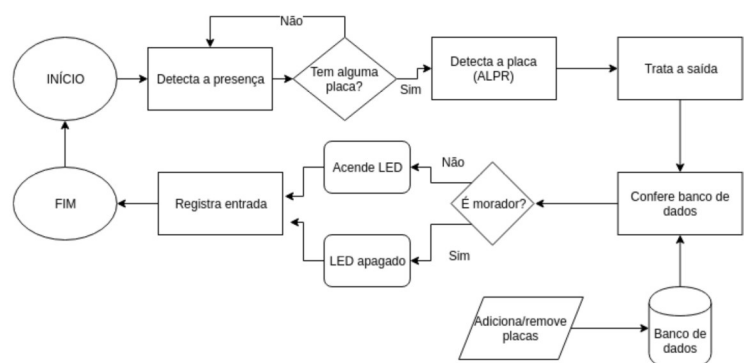


Figura 3. Fluxograma do projeto

A primeira parte está relacionada a detecção do carro no local. A ideia nessa parte é utilizar um sensor ultrassônico para medir a distância da câmera até o carro. Ao se atingir uma distância específica, o “gatilho” seria ativado, para que a foto fosse tirada. Caso na foto tirada nenhuma placa seja identificada, o programa encerra e volta para o ‘loop’ em que ele aguarda a chegada de um carro. Isso foi feito, tendo em vista que a presença de animais ou outros elementos externos ative esse gatilho.

Após a foto ser tirada, o sistema faz a detecção dos caracteres da placa com a ferramenta ALPR a partir da foto tirada. A ferramenta ALPR, foi instalada previamente, assim como todas as suas dependências necessárias (OpenCV, Tesseract e Leptonica). Essa ferramenta permite que a detecção dos caracteres da placa sejam definidos para o padrão da brasileiro, assim como outras línguas. No caso do presente projeto, o complemento usado foi “-c br”, por exemplo. A saída é dada num arquivo de texto (“saida.txt”). Um exemplo de foto tirada pode ser visto a seguir.



Figura 4. Exemplo de foto tirada pelo sistema

Após o ALPR ser feito, é preciso tratar o arquivo de texto referente a saída. A saída do ALPR contém diversos elementos além do número da placa (que é o necessário ao projeto), como a descrição dos resultados e um nível de confiança. Um exemplo de saída do ALPR pode ser visto a seguir.

O tratamento da saída é feito basicamente através da manipulação de strings e de arquivos de texto. Além de extrair o número da placa, essa parte do código é responsável por corrigir possíveis erros na detecção dos caracteres. Um exemplo de correção é o fato do sistema confundir 1 com I, 0 com O, D ou Q, 8 com B. Logo, sabendo que o padrão brasileiro consiste de 3 letras seguidos de 4 algarismos, a correção nesse sentido é possível.

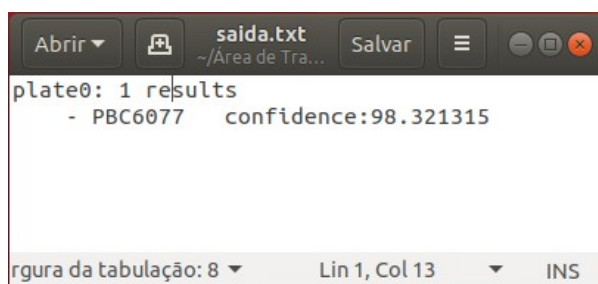


Figura 5. Saída do ALPR sem o tratamento

A saída após o tratamento é dada tanto numa string no código, quanto num arquivo de texto, como pode ser visto a seguir.

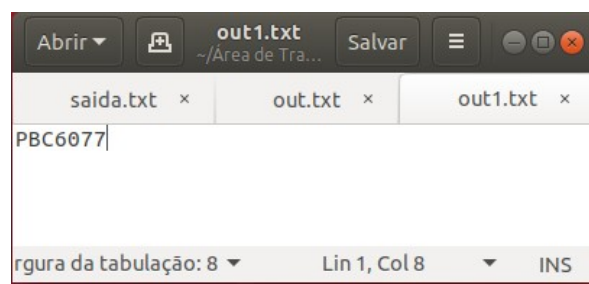


Figura 6. Saída do ALPR após o tratamento

O próximo passo a ser feito é conferir a placa no banco de dados, para ver se o carro é de um morador ou não. O banco de dados, é dado por um arquivo .txt contendo as placas dos moradores. Se a placa obtida da foto for de um morador, o led permanece apagado, demonstrando que não há nenhum problema. Caso a placa não seja de um morador, o led acenderá, indicando ao porteiro(?) que o carro na entrada é de um visitante e que ele eventualmente precisa se identificar para adentrar o condomínio.

Por fim, a entrada do veículo é registrada em uma pasta do dia atual, para melhor divisão, com uma foto cujo nome é a própria placa do carro. Uma outra forma seria salvar tais dados numa planilha para facilitar e agilizar a verificação, o que pode ser feito pelo Google Apps Script.

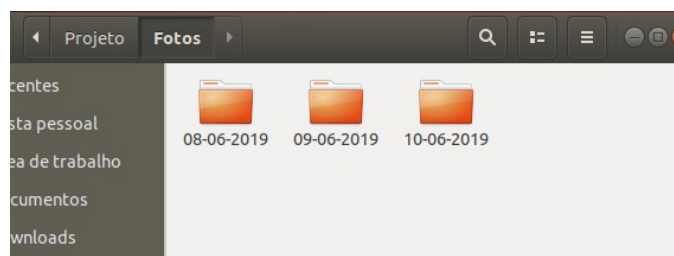


Figura 7. Exemplo das pastas para registro

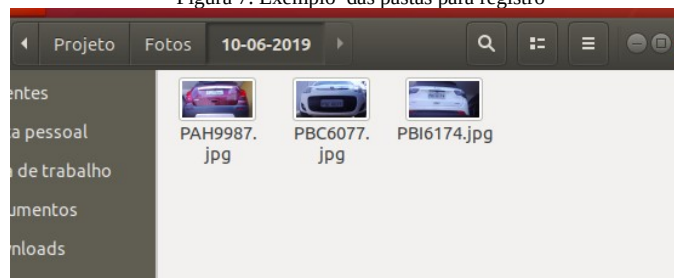


Figura 8. Exemplo das imagens nas pastas

O código pode ser visto no anexo.

## VI. BENEFÍCIOS

Ao se fazer o controle de acesso, através do reconhecimento de placas de veículos, o condomínio inibe de certa forma a ação de agentes criminosos ou até mesmo prestadores de

serviço indesejados, o que traz consigo mais segurança e comodidade.

Já quanto aos moradores, a entrada pode ser facilitada e agilizada, uma vez que a placa já estaria registrada no sistema.

Outro ponto é que, ao registrar as placas dos veículos que entraram e saíram do condomínio, junto com a respectiva data e hora (e até mesmo uma foto de quem está ao volante, por exemplo), a identificação das pessoas no condomínio é facilitada, além de facilitar o gerenciamento do próprio condomínio.

## VII. REVISÃO BIBLIOGRÁFICA

Atualmente, existem muitos modelos de controle de acesso veicular para condomínios, que fazem o controle e registro de todos automóveis que entram e saem. Um problema dos modelos existentes é o alto custo, pois estão vinculados à mão de obra para instalação e implantação dessas prestadoras de serviços. Dois exemplos desses modelos são:

### A. Sistemas De Leitura De Placas De Veiculos Em Estacionamento – Ark/SIRAM [3]

#### a. Funcionalidades:

- i. Banco de dados com registros de entradas e saída;
- ii. Registra a imagem no instante da entrada do veículo;
- iii. Detecção de placas diurnas e noturnas;
- iv. Relatório diário de entrada e saída de veículos.
- v. Custo: R\$ 6.800,00.

### B. PlateView Embedded – Pegasus Technology [4]

#### a. Funcionalidades:

- i. Detecção de aproximação do veículo por sensores elétricos e detecção da câmera;
- ii. Visualização ao vivo da entrada do local onde o sistema é implementado;
- iii. O usuário pode consultar informações históricas de placas num determinado período;
- iv. Permite o cadastro de placas de veículos que serão autorizadas a entrar numa determinada área.
- v. *Custo*: Somente disponibilizado após contato para orçamento.

## VIII. RESULTADOS E DISCUSSÕES

Para o presente ponto de controle, foram-se gastas muitas horas de trabalho. Muitos avanços foram obtidos: conseguiu-se implementar o código em praticamente sua totalidade, juntando todos os códigos em um. Algumas alterações foram

realizadas tanto na forma que os códigos funcionariam quanto no próprio fluxograma do projeto.

As dificuldades encontradas no ponto de controle anterior foram em sua maioria resolvidas. A câmera da Raspberry foi substituída pela WebCam e não houveram problemas maiores quanto a isso.

Entretanto, outros problemas foram identificados no ponto de controle. A ausência de um gatilho para tirar a foto foi um deles. Tal problema será sanado ao se utilizar um sensor ultrassônico para detectar a distância do carro para que a foto seja tirada.

Um outro ponto a ser abordado é a utilização de planilhas para registrar a entrada e saída de carros. Será discutido entre a dupla se isso será inserido e como isso será feito (Excel, Google Apps Scripts...).

O último ponto é quanto a estrutura do projeto. Espera-se que esse problema seja resolvido para o próximo ponto de controle, assim como os outros citados.

Portanto, percebeu-se que houve muitos avanços para este ponto de controle se comparado aos pontos de controle anteriores. Além disso, o aprendizado adquirido com esse projeto foi de grande valia.

## REFERÊNCIAS

- [1] ABRASSP. Disponível em: < <http://www.abrassp.com.br/>>. Acessado em 28/03/2019.
- [2] MTG Tech. Controle de acesso veicular: como fazer da forma certa. Disponível em: < <http://mtgtech.com.br/controle-de-acesso-veicular/>>. Acessado em 28/03/2019.
- [3] Mercado Livre. Disponível em: < [- \[4\] Pegasus Tec. Disponível em: <https://www.pegasustec.com.br/plateview-embedded>. Acessado em 28/03/2019.](https://produto.mercadolivre.com.br/MLB-834785122-sistemas-de-leitura-de-placas-de-veiculos-em-estacionamento-_JM?matt_tool=31832664&matt_word=&gclid=EAIaIQobChMIzJynmoOm4QIVDguRCh3d8wMFEAYYASABEGJWy_D_BwE>)

## ANEXO

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <unistd.h>
#include <pthread.h>
#include <time.h>
#include <wiringPi.h>

#define LedPin 0
#define MAX 256

int main()
{
/*****
Essa parte tira a foto
*****/
    system("fswebcam -d /dev/video0 -r 1280x720 -S 10 -F 5 --no-banner foto.jpeg");
/*****
Essa parte faz o ALPR da foto
*****/
    system("alpr -c br tracker0.jpg -n 1 > saida.txt");
/*****
Essa parte trata a saída do ALPR, para que fique sendo apenas o numero da placa, ignorando os demais elementos.
*****/

    system("grep -vwE 'No' saida.txt > saidatratada.txt");
    system("grep -vwE 'plate0' saida.txt > saidatratada.txt");
    FILE *f = fopen("saidatratada.txt", "r");
    fseek(f, 6, SEEK_SET); // set file pointer to 2nd position (0-indexed)
    char part[10] = {0}; // array for 9 characters plus terminating 0
    fread(part, 1, 12, f); // read 9 members of size 1 (characters) from f into part

    FILE *s = fopen("out.txt", "w");
    fprintf(s, "%s", part); // or any other way to output part, like using in printf()
    fclose(s);

/*****
Essa parte corrige o numero da placa caso haja alguma confusão no ALPR (Ex.: 0 com O, 8 com B...)
*****/

    FILE *fp = fopen("out.txt", "r+");
    char array[100];
    fscanf(fp, "%s", array);
    int counter;

    while(array[counter] != '\0')
    {
        if(counter<3) //LETRAS
        {
            if(array[counter] == '0')
            {
                array[counter] = 'O';
            }
            if(array[counter] == '1')
            {
```

```

        array[counter] = 'I';
    }
        if(array[counter] == '2')
    {
        array[counter] = 'Z';
    }
        if(array[counter] == '4')
    {
        array[counter] = 'A';
    }
        if(array[counter] == '5')
    {
        array[counter] = 'S';
    }
        if(array[counter] == '8')
    {
        array[counter] = 'B';
    }
    }

else //NÚMEROS
{
    if(array[counter] == 'O')
    {
        array[counter] = '0';
    }
    if(array[counter] == 'Q')
    {
        array[counter] = '0';
    }
    if(array[counter] == 'D')
    {
        array[counter] = '0';
    }
    if(array[counter] == 'I')
    {
        array[counter] = '1';
    }
    if(array[counter] == 'Z')
    {
        array[counter] = '2';
    }
    if(array[counter] == 'A')
    {
        array[counter] = '4';
    }
    if(array[counter] == 'S')
    {
        array[counter] = '5';
    }
    if(array[counter] == 'B')
    {
        array[counter] = '8';
    }
}
counter++;
}

```

```
fprintf(fp,"%s", array);
fclose(fp);
```

```
FILE *f2 = fopen("out.txt", "r");
fseek(f2, 7, SEEK_SET);
char part2[10] = {0};
fread(part2, 1, 7, f2);
```

```
FILE *s2 = fopen("out1.txt", "w");
fprintf(s2,"%s", part2);
fclose(s2);
```

```
printf("\n%s\n", part2);
```

```
/******
```

Essa parte confere se o carro entrando no condomínio pertence a um morador (led apagado) ou não (led aceso)

```
*****/
```

```
int n=0;
char arq_txt[10000];
```

```
FILE *db = fopen("placasregistradas.txt", "r");
```

```
/*if (db == NULL)
{
    printf("Não foi possível acessar o arquivo\n");
    exit(-1);
}*/
```

```
while(!feof(db))                //enquanto não chegar ao fim do arquivo
{
    fscanf(db,"%s",arq_txt);
    if(strcmp(arq_txt,part2) == 0) //se encontra uma placa igual no texto
        n++;
}
```

```
if(n==0)
{
    if(wiringPiSetup() == -1) { //when initialize wiringPi failed, print message to screen
        printf("setup wiringPi failed !\n");
        return -1;
    }
}
```

```
    pinMode(LedPin, OUTPUT);
    digitalWrite(LedPin, LOW);
    delay(2000);
    digitalWrite(LedPin, HIGH);
```

```
    printf("\nVISITANTE\n\n");
}
else
{
    printf("\nMORADOR!\n\n");
}
```

```

/*****
Essa parte salva a data atual numa string
*****/

    time_t raw;
    time(&raw);

    struct tm *time_ptr;
    time_ptr = localtime(&raw);

    char date[11];
    strftime(date, 11, "%d-%m-%Y", time_ptr);

/*****
Essa parte cria o banco de dados, ou seja, salva a foto cujo nome é a propria placa numa pasta do
dia atual cria uma planilha contendo a placa, a data e hora que chegou ao local.
*****/

    int ch;
    FILE *mv1, *mv2;
    char dest[MAX];
    char extensao[10] = ".jpg";

    strcat(part2, extensao);
    mv1 = fopen("foto.jpg", "r");          /* abre o arquivo de origem da foto para leitura */

    chdir("Fotos/");
    char cmd[100];                          /* cria a pasta com o nome sendo a data atual */
    strcpy (cmd, "mkdir ");
    strcat (cmd, date);
    strcat(cmd, "/");

    system(cmd);
    chdir(date);

    mv2 = fopen(part2, "w"); /* abre o arquivo de destino da foto para escrita */

    /* copia o arquivo da origem para o destino */
    while (!feof(mv1)) {
        ch = fgetc(mv1);
        fputc(ch, mv2);
    }

    /* fecha os arquivos e volta pra pasta de trabalho */
    fclose(mv1);
    fclose(mv2);
    chdir("..");
    chdir("..");
}

```