

Centro Federal de Educação Tecnológica de Minas Gerais  
*Campus* Divinópolis  
Graduação em Engenharia Mecatrônica

Ronaldo Ribeiro Tavares Junior  
Vitor Ferreira Décimo

DESENVOLVIMENTO DE UM DISPOSITIVO ELETRÔNICO PARA  
MONITORAMENTO DE VAGAS DE TRÂNSITO ESPECIAIS



Divinópolis  
2018



Ronaldo Ribeiro Tavares Junior

Vitor Ferreira Décimo

DESENVOLVIMENTO DE UM DISPOSITIVO ELETRÔNICO PARA  
MONITORAMENTO DE VAGAS DE TRÂNSITO ESPECIAIS

Monografia de Trabalho de Conclusão de Curso  
apresentada ao Colegiado de Graduação em Engenharia Mecatrônica como parte dos requisitos exigidos para a obtenção do título de Engenheiro Mecatrônico.

Eixo de Formação: Computação e Eletrônica.

Orientador: Prof. Dr. João Carlos de Oliveira



Divinópolis

2018

(Catalogação - Biblioteca Universitária – Campus Divinópolis – CEFET-MG)

|       |  |
|-------|--|
| T231d | Tavares Júnior, Ronaldo Ribeiro.<br>Desenvolvimento de um dispositivo eletrônico para monitoramento de vagas de trânsito especiais. / Ronaldo Ribeiro Tavares Júnior ; Vitor Ferreira Décimo. - Divinópolis, 2018.<br>76f.; il.<br><br>Orientador: Prof. Dr. João Carlos de Oliveira.<br><br>Trabalho de Conclusão de Curso (graduação) – Colegiado de Graduação em Engenharia Mecatrônica do Centro Federal de Educação Tecnológica de Minas.<br><br>1. Eletrônica. 2. Computação. 3. Acessibilidade. 4. Visão Computacional. 5. Monitoramento. 6. Reconhecimento - Algoritmos. 7. Estacionamento – Vagas Especiais. 8. Trânsito. I. Oliveira, João Carlos de. II. Décimo, Vitor Ferreira. III. Centro Federal de Educação Tecnológica de Minas Gerais. IV. Título.<br><br>CDU: 62(043) |
|-------|--|



**Centro Federal de Educação Tecnológica de Minas Gerais**  
**CEFET-MG / Campus Divinópolis**  
**Curso de Engenharia Mecatrônica**

Monografia intitulada “*Desenvolvimento de um dispositivo eletrônico para monitoramento de vagas de trânsito especiais*”, de autoria do graduando Ronaldo Ribeiro Tavares Junior e Vitor Ferreira Décimo, aprovada pela banca examinadora constituída pelos seguintes professores:

---

Prof. Dr. João Carlos de Oliveira - CEFET-MG / Campus Divinópolis - Orientador

---

Prof. Dr. André Luiz Maravilha Silva - CEFET-MG / Campus Divinópolis

---

Prof. Me. André Luiz Paganotti - CEFET-MG / Campus Divinópolis

---

Prof. Dr. Lúcio Flávio Santos Patrício  
Coordenador do Curso de Engenharia Mecatrônica  
CEFET-MG / Campus Divinópolis

Divinópolis - Dezembro de 2018



# Abstract

The vacancies of special parking spaces are of huge help for people with disabilities, the elderly and pregnant woman. According to the Brazilian law of inclusion of people with disability, 2% of the parking spaces presents in a city have to be reserved for handicapped, and another 5% for elderly, as stated in the status of the elderly. Currently, the only way to observe this kind of parking spaces is through traffic inspectors, what requires a huge demand of professionals, turning it financially unviable to totally monitor the available parking spaces. The use of visual inspection instruments can provide a meaningful progress of a monitoring system, allowing for the access and exchange of information in a fast and safe way. It is proposed in this project the projection and development of an embedded monitoring system in order to reduce expenses with professionals and provide an effective way to monitor special parking spaces. This system will detect the parked vehicles plate, determine if it has a permission to occupy the space and execute the appropriate corrective measurement to prevent unauthorized parking. Thus providing the well being for those who really need the destined parking spaces. For the development of such a project, it will be essential to use resources of computational vision in the plate's recognition, of databases to control the plates, of programming mobile devices for the registration of vehicles, of electronics to execute all the communication of the hardware of the device, among others.

**Keywords:** accessibility, computational vision, monitoring, recognition of numerals.



# Resumo

As vagas de estacionamento especiais são de grande ajuda para pessoas com deficiência, idosos e gestantes. De acordo com a lei brasileira de inclusão da pessoa com deficiência, 2% das vagas de estacionamento presentes em um município são reservadas para deficientes físicos, e outros 5% para idosos, conforme prevê o estatuto do idoso. Atualmente o único meio de monitoramento deste tipo de vaga é através de fiscais de trânsito, o que requer uma alta demanda de profissionais da área, tornando assim inviável financeiramente o monitoramento por completo de todas as vagas disponíveis. A utilização de ferramentas de inspeção visual pode proporcionar significativo avanço no monitoramento de um sistema, permitindo o acesso e a troca de informações de forma rápida, agilizando as medidas corretivas necessárias e tornando este sistema mais rápido e seguro. A fim de minimizar as despesas profissionais e proporcionar um monitoramento efetivo das vagas especiais, é proposto neste trabalho o projeto e desenvolvimento de um sistema embarcado de monitoramento. Este sistema deve detectar a placa do veículo estacionado, determinar se este tem autorização para ocupar o devido espaço e executar as medidas corretivas cabíveis para coibir o estacionamento não autorizado. Visa proporcionar o bem estar para quem realmente necessita das vagas especiais. Para o desenvolvimento de tal projeto é necessário o uso de recursos da visão computacional no reconhecimento das placas, de banco de dados para o controle das placas, de programação de dispositivos móveis para o cadastro de veículos, da eletrônica para fazer toda comunicação de hardware do dispositivo, entre outros que serão abordados mais a frente.

**Palavras-chave:** Acessibilidade, Visão computacional, Monitoramento, Reconhecimento de Algoritmos.



# Sumário

|          |                                |          |
|----------|--------------------------------|----------|
| <b>1</b> | <b>Introdução</b>              | <b>1</b> |
| 1.1      | Definição do Problema          | 2        |
| 1.2      | Motivação                      | 2        |
| 1.3      | Objetivos do Trabalho          | 2        |
| 1.3.1    | Objetivo geral                 | 2        |
| 1.3.2    | Objetivos Específicos          | 2        |
| 1.4      | Estado da Arte                 | 3        |
| <b>2</b> | <b>Fundamentos</b>             | <b>5</b> |
| 2.1      | Revisão Bibliográfica          | 5        |
| 2.2      | Metodologia                    | 7        |
| 2.2.1    | Características do Dispositivo | 8        |
| 2.3      | Fundamentação Teórica          | 9        |
| 2.3.1    | Visão Computacional            | 9        |
| 2.3.2    | OpenCV                         | 9        |
| 2.3.3    | Processamento de Imagem        | 9        |
| 2.3.4    | Filtros                        | 10       |
| 2.3.5    | Realce de Contraste            | 11       |
| 2.3.6    | Segmentação de Imagem          | 12       |
| 2.3.7    | Reconhecimento de Padrões      | 13       |
| 2.3.8    | OCR                            | 13       |

|          |   |           |
|----------|---|-----------|
| 2.3.9    | UBUNTU . . . . .  | 14        |
| 2.3.10   | Python . . . . .  | 14        |
| 2.3.11   | Web Design . . . . .  | 15        |
| 2.3.12   | Placa de identificação Veicular . . . . .   | 15        |
| 2.3.13   | Banco de Dados (SQLite) . . . . .   | 16        |
| 2.3.14   | Comunicação Wireless . . . . .  | 17        |
| 2.3.14.1 | Bluetooth . . . . .   | 17        |
| 2.3.14.2 | Wi-Fi . . . . .   | 18        |
| 2.3.15   | Câmera de Transmissão Sem Fio . . . . .   | 18        |
| <b>3</b> | <b>Desenvolvimento . . . . .</b>  | <b>21</b> |
| 3.1      | Elaboração do Algoritmo de Reconhecimento . . . . .                                 | 21        |
| 3.1.1    | Interação entre OpenCv e sistema operacional . . . . .                              | 21        |
| 3.1.2    | Padrões técnicos para obtenção da imagem . . . . .                                  | 22        |
| 3.1.3    | Extração de imagem . . . . .  | 23        |
| 3.1.4    | Descarte de zona de não interesse . . . . .   | 24        |
| 3.1.5    | Binarização . . . . .   | 24        |
| 3.1.6    | Destacamento da figura de interesse . . . . .                                       | 26        |
| 3.1.7    | Abertura de imagem . . . . .  | 26        |
| 3.1.8    | Corte e isolamento dos caracteres alfanuméricos . . . . .                           | 27        |
| 3.1.9    | Identificação de Caracteres . . . . .   | 28        |
| 3.2      | Aplicação Móvel de Cadastro de Usuários . . . . .                                   | 29        |
| 3.2.1    | Criação do Aplicativo . . . . .   | 29        |
| 3.2.2    | Ferramentas Utilizadas . . . . .  | 29        |
| 3.2.3    | Interface Gráfica . . . . .   | 29        |
| <b>4</b> | <b>Resultados . . . . .</b>   | <b>33</b> |
| 4.1      | Desenvolvimento WEB . . . . .   | 33        |
| 4.2      | Aplicação de técnicas de visão computacional e algoritmo de identificação . . . . . | 36        |
| 4.2.1    | Identificação de placas . . . . .   | 40        |
| 4.3      | Banco de Dados . . . . .  | 43        |
| 4.4      | Periféricos . . . . .   | 45        |
| <b>5</b> | <b>Considerações Finais . . . . .</b>   | <b>49</b> |
| 5.1      | Conclusões . . . . .  | 49        |

|          |  |           |
|----------|--|-----------|
| 5.2      | Propostas de continuidade . . . . .    | 50        |
| <b>A</b> | <b>Códigos . . . . .</b>               | <b>53</b> |
| A.1      | Instalação biblioteca OpenCv . . . . . | 53        |
| A.2      | Leitura e identificação . . . . .      | 53        |
| A.3      | Cadastro Banco de dados . . . . .      | 58        |
|          | <b>Bibliografia . . . . .</b>          | <b>59</b> |



## Lista de figuras

|   |    |
|---|----|
| Figura 1 – Características do dispositivo. . . . .                          | 9  |
| Figura 2 – Matriz Máscara . . . . .   | 10 |
| Figura 3 – Binarização . . . . .  | 11 |
| Figura 4 – Segmentação por borda externa . . . . .                          | 12 |
| Figura 5 – Segmentação por região . . . . .                                 | 13 |
| Figura 6 – Ferramenta Terminal . . . . .                                    | 14 |
| Figura 7 – Tipologia dos caracteres . . . . .                               | 16 |
| Figura 8 – Dimensões da Placa de identificação . . . . .                    | 16 |
| Figura 9 – SQLite . . . . .   | 17 |
| Figura 10 – Sistema da câmera . . . . .                                     | 18 |
| Figura 11 – Câmera de transmissão sem fio com infravermelho. . . . .        | 19 |
| Figura 12 – Distâncias padrão de instalação da câmera . . . . .             | 23 |
| Figura 13 – a - imagem original. b - imagem após emprego do método. . . . . | 24 |
| Figura 14 – Imagem binária. . . . .   | 25 |
| Figura 15 – Imagem original. . . . .  | 26 |
| Figura 16 – imagem com erosão. . . . .                                      | 27 |
| Figura 17 – imagem com erosão e dilatação. . . . .                          | 27 |
| Figura 18 – Caracteres alfabéticos isolados. . . . .                        | 28 |
| Figura 19 – Caracteres numéricos isolados. . . . .                          | 28 |
| Figura 20 – Layout do Aplicativo. . . . .                                   | 31 |

|  |    |
|--|----|
| Figura 21 – Pagina Home . . . . .  | 35 |
| Figura 22 – Aba de cadastro de usuários . . . . .  | 35 |
| Figura 23 – Home aberta via PC . . . . .   | 36 |
| Figura 24 – (a) - compilação do algoritmo com placa cadastrada no banco de dados<br>(b) - placa identificada pela câmera . . . . .     | 39 |
| Figura 25 – (a) - compilação do algoritmo com placa não cadastrada no banco de<br>dados (b) - placa identificada pela câmera . . . . . | 39 |
| Figura 26 – Cadastro de placa no banco de dados . . . . .  | 40 |
| Figura 27 – Exemplo banco de imagens . . . . .   | 41 |
| Figura 28 – Placa em má conservação. . . . .   | 43 |
| Figura 29 – Modelo entidade relacionamento. . . . .  | 44 |
| Figura 30 – Notebook utilizado. . . . .  | 45 |
| Figura 31 – Câmera. . . . .  | 46 |
| Figura 32 – Mini caixa de som. . . . .   | 46 |
| Figura 33 – Características Técnicas LEDES. . . . .  | 46 |
| Figura 34 – Arduino UNO. . . . .   | 47 |

# Lista de tabelas

|   |    |
|---|----|
| Tabela 1 – Relação de placas identificadas. . . . . | 41 |
| Tabela 2 – Relação de placas identificadas. . . . . | 42 |



## Lista de acrônimos e notações

|         |                                   |
|---------|-----------------------------------|
| DETRAN  | Departamento Nacional de Trânsito |
| CI      | Circuito Integrado                |
| Ph.D    | Philosophiae Doctor               |
| WPAN    | Wireless Personal Area Network    |
| INTEL   | Integrated Eletronics             |
| SDK     | Software Development Kit          |
| CONTRAN | Conselho Nacional de Trânsito     |
| MCid    | Ministério das Cidades            |



## Introdução

A mobilidade urbana é um atributo das cidades e se refere à facilidade de deslocamentos de pessoas e bens no espaço urbano. Tais deslocamentos são feitos considerando-se veículos, vias e toda a infraestrutura que possibilitam esse ir e vir cotidiano (CIDADES, 2018). Observa-se atualmente um planejamento urbano que vem priorizando o acesso a deficientes, que por diversas enfermidades encontram maior dificuldade no acesso a lugares de interesse. Porém, tal planejamento não assegura de forma eficaz o respeito dos demais usuários ao acesso prioritário e as diversas formas de privilégios concedidos aos deficientes.

Com o aumento do número de veículos nos grandes centros, estimado em cerca de 120% (DENATRAN, 2018), é necessária uma maior eficiência na identificação de veículos. Com o reconhecimento de placas de identificação de veículos através de câmeras, os custos operacionais de fiscalização podem ser reduzidos drasticamente, além do monitoramento se tornar mais extensivo e eficiente.

A popularização dos smartphones é considerada por muitos, a revolução tecnológica de maior impacto nos últimos tempos após a revolução causada pela Internet e pelas redes sociais (IDC, 2013). As vendas de smartphones no país totalizaram 47,7 milhões de unidades em 2017 (IDC (2017)). Por meio dos dados citados, pode-se notar o potencial existente para a obtenção de dados dos usuários. Pode-se obter um portal capaz de cadastrar seus dados e uma base de cadastro capaz de gerar uma forma efetiva de consulta de dados de usuários.

## 1.1 Definição do Problema

O problema fundamental neste trabalho de conclusão de curso trata-se do acesso ilegal a vagas de trânsito especiais, tais como as ofertadas a idosos, deficientes e gestantes. O uso indevido destas vagas acaba gerando um desconforto mental e de mobilidade a quem realmente necessita destas vagas.

## 1.2 Motivação

A escolha deste tema para o trabalho de conclusão de curso apresentado visa aplicar os conhecimentos adquiridos ao longo do curso, em computação e eletrônica para o desenvolvimento de um sistema capaz de identificar e notificar infratores de trânsito. Existe o interesse dos membros em desenvolver um projeto capaz de gerar um ganho em qualidade de vida a quem sofre devido a deficiências, se enquadrando assim, no tema proposto a este trabalho. Além disso, esta proposta pode se tornar um produto comercial, gerando uma primeira interação com o mercado de trabalho, e pode gerar novos trabalhos de conclusão de curso no futuro.

## 1.3 Objetivos do Trabalho

### 1.3.1 Objetivo geral

Tem-se como objetivos principais deste trabalho desenvolver um sistema de monitoramento de vagas de trânsito especiais utilizando visão computacional, na identificação e análise de placas de veículos. Além disso, visa elaborar uma interface de cadastro de dados em um dispositivo móvel.

### 1.3.2 Objetivos Específicos

- Desenvolver algoritmo de identificação e reconhecimento de algarismos em placas de identificação veículos.
- Aplicar filtros de captura e isolamento de imagem de forma correta.
- Desenvolver um web site responsivo para recolhimento e associação de dados entre veículos e usuários.

- Elaborar banco de dados com informações associadas às placas de identificação veiculares.
- Elaborar comunicação entre câmera e sistema.
- Elaborar comunicação entre website desenvolvido e sistema de cadastro de dados.
- Elaborar comunicação entre dados de cadastro e dispositivo.
- Realizar ajustes entre a comunicação dos dispositivos.

## 1.4 Estado da Arte

Nos tempos atuais, podem ser encontradas diversas pesquisas que abordam aplicações de visão computacional na identificação de objetos, isto se justifica pela vasta gama de aplicações para esta tecnologia.

Dentre as pesquisas, podemos destacar as relacionadas à detecção e identificação de caracteres alfanuméricos, das quais podemos citar (Damore (2015)), que desenvolveu um modelo robusto de análise e identificação dos caracteres em condições complexas de aquisição. O modelo possui a finalidade de gerar recursos computacionais capazes de identificar os caracteres presentes em placas de identificação de veículos com situações adversas, tais como: reflexos, inclinações, mau estado de conservação, dentre as variantes.

O modelo conta com várias etapas, que iniciam em um pré-processamento e segmentação da região de interesse, que terminam com a segmentação e reconhecimento dos caracteres.

Também se destacam pesquisas relacionadas à utilização de aplicativos móveis para obtenção de dados. Referente a este tema, podemos citar (Moraes (2016)), este, teve como propósito o desenvolvimento de um aplicativo móvel para coleta de informações relacionadas à saúde. Tal aplicação tem como objetivo a coleta de dados e elaboração de uma base de dados, referentes a saúde dos usuários.

Considerando o aspecto comercial como primordial, pode-se referenciar protótipos de produtos que se assemelham ao trabalho pesquisado neste documento. Um deles é citado por (Shimoskai (2017)), este se trata de um controlador eletrônico de vagas especiais (CEVE). O protótipo de Shimoskai tem o objetivo de fiscalizar vagas de trânsito, através de um sensor de presença, que indica a ocupação da vaga, é acionado por um sensor que

visa identificar uma etiqueta eletrônica previamente instalada nos veículos autorizados. Da mesma forma, também podemos destacar (Yamawaki (2018)), ele descreve um sistema de reconhecimento de veículos estacionados em vagas privadas. Através de sensores instalados na massa asfáltica, o veículo é detectado e identificado por um sistema de análise do sinal emitido por componentes instalados preliminarmente no veículo autorizado.

Atualmente, o investimento empregado em atividades de ganho em qualidade de vida para deficientes é cada vez mais alto. De acordo com (Goncalves (2016)), a fundação Google.Org fez uma doação de cerca de US\$20.000.000,00 à diversas fundações, para o financiamento de pesquisas em tecnologia de assistência a deficientes físicos. Da mesma forma existem diversas outras linhas de investimentos públicos e privados para financiar pesquisas sobre tecnologia aplicada a melhoria da qualidade de vida de pessoas com algum tipo de deficiência.

De acordo com as referências mencionadas, nota-se que existem diversas pesquisas direcionadas a utilização da visão para identificação de veículos comuns, do mesmo modo, existem trabalhos baseados no monitoramento de vagas utilizando uma diversidade de sensores unidos a dispositivos implantados em veículos.

Sendo assim, a utilização de sistemas embarcados voltados para o monitoramento de vagas especiais utilizando visão computacional é uma nova solução para um problema já identificado anteriormente. Tal aplicação deve ser desenvolvida para possibilitar uma fiscalização mais simples e eficiente, agregando assim, mais qualidade de vida aos usuários das vagas reservadas por meio da fiscalização eficiente.

## Fundamentos

Este capítulo apresenta os marcos históricos relevantes para a realização deste projeto assim como os conceitos fundamentais para o aprofundamento da pesquisa sobre o mesmo.

### 2.1 Revisão Bibliográfica

Projetos como o proposto são exequíveis devido as grandes evoluções tecnológicas ocorridas décadas atrás. Tecnologias que foram criadas, e hoje são de suma importância para o desenvolvimento humano. Dentre diversas criações, podemos mencionar uma das mais importantes, a que impulsionou toda a evolução da tecnologia atual e que é de suma relevância para este trabalho, o Computador Pessoal. Os computadores pessoais e laptops são ícones da era da informação; estes equipamentos são o que muitas pessoas consideram como "computador". Entretanto, atualmente as formas mais comuns de computador em uso são os sistemas embarcados, pequenos dispositivos usados para controlar outros dispositivos, como robôs, câmeras digitais ou brinquedos (SILVEIRA, 2004). Com a vinda dos computadores, abriu-se um universo de oportunidades para novos projetos, levando em conta que a partir daí, as pessoas passaram a ter em suas mãos uma ferramenta capaz de criar tecnologias jamais vistas. É disso que se trata este trabalho, uma tecnologia que pode ser considerada inovadora e que só é possível graças a ferramenta computador.

Mais um marco da evolução tecnológica foi o advento dos Smartphones, revolução tecnológica que entregou acessibilidade para a população, facilitando o cotidiano através de aplicativos. O termo "smartphone" foi idealizado no final da década de 1990 – naquela época, uma fração pequena de funções adicionais além de chamadas era desenvolvida por

dispositivos (VOLTOLINI, 2017).

Com a possibilidade de conexão em praticamente qualquer lugar, cresce o número de pesquisas e investimentos na usabilidade de interfaces de dispositivos móveis, de modo a tentar transportar o poder computacional presente em computadores “de mesa” para dispositivos de fácil mobilidade (Ribeiro (2007)). Devido a esse avanço nas tecnologias móveis que este projeto se torna mais acessível, pois usa o smartphone como o seu principal meio de comunicação com o cliente, proporcionando mais acessibilidade e praticidade na hora que o cliente quiser fazer o seu cadastro, de maneira simples e rápida, sem a necessidade do cliente se locomover até um local específico.

Mas antes dos primeiros computadores pessoais serem inventados, estudos já estavam sendo feitos a respeito de visão computacional, que em 1960 começava a ser idealizada por *Larry Roberts*, Ph.D no *Massachusetts Institute of Technology (MIT)*, que escreveu sobre as possibilidades de extrair informações geométrica 3D de uma vista de perspectiva 2D de blocos. Assim, nota-se as possíveis aplicabilidades de tais informações, pesquisadores do *MIT* e de outras instituições do mundo seguiram os estudos no contexto da visão computacional dos blocos (HUANG, 1996). A visão computacional é um dos principais artifícios deste projeto, pois através dela que acontece todo o reconhecimento da placa do veículo, proporcionando assim um projeto tecnológico e inovador, diminuindo as chances de erros e otimizando as fiscalizações de trânsito, levando em conta de que todos os carros que pararem nas vagas serão monitorados.

Para obter sucesso através da visão computacional, deve se utilizar periféricos que estejam de acordo com a finalidade do projeto, como a câmera, que é o dispositivo utilizado na visão computacional e que deve estar bem calibrada para a realização de sua função. O problema geral referente a calibração de câmeras tem sido estudado de forma exaustiva para aplicações em visão computacional, realidade virtual e realidade aumentada. Um exemplo clássico é o algoritmo de (Tsai (1991)), o qual pode ser usado para obter a configuração completa dos parâmetros de uma dada câmera num ambiente controlado. (Wang (1991)) exploradas as linhas de fuga (conhecidas com *vanishing lines*) em uma cena com o intuito de obter os parâmetros de câmera com base num padrão hexagonal focado no plano do chão. (Douxchamps (2009)) apresenta uma nova abordagem para calibração de câmera com maior acurácia, reportando erros na ordem de 0.002 pixels quando marcas planares são utilizadas.

Para a interação de todos estes dispositivos, é necessário um banco de dados capaz de armazenar todas as informações que são obtidas, este banco de dados recebe todas as informações passadas pelo usuário através de um aplicativo. O primeiro Sistema Gerenciador de Banco de Dados (*SGBD*) comercial surgiu no final de 1960 com base nos primitivos sistemas de arquivos disponíveis na época, os quais não controlavam o acesso concorrente por vários usuários ou processos. Os *SGBDs* evoluíram desses sistemas de arquivos de armazenamento em disco, criando novas estruturas de dados com o objetivo de armazenar informações. Com o tempo, os *SGBD's* passaram a utilizar diferentes formas de representação, ou modelos de dados, para descrever a estrutura das informações contidas em seus bancos de dados. Hoje, os seguintes modelos de dados são normalmente utilizados pelos *SGBD's*: modelo hierárquico, modelo em redes, modelo relacional (amplamente usado) e o modelo orientado a objetos (TAKAI, 2005).

Atualmente, com a junção de várias tecnologias faz-se possível a criação de dispositivos que tem o intuito de melhorar a qualidade de vida das pessoas. E como essas tecnologias estão sempre em constante evolução, há de se esperar cada vez mais por projetos e sistemas inovadores e otimizados.

## 2.2 Metodologia

Para o desenvolvimento de um projeto mecatrônico, algumas etapas de trabalho devem ser respeitadas, dentre estas, se pode destacar duas, que em específico são fundamentais para o desenvolvimento deste trabalho. A primeira, trata do estudo do projeto, onde se busca a obtenção do conhecimento teórico necessário para o desenvolvimento das atividades propostas, tais como: características do sistema; condições de operação; custo computacional; especificações de projeto e definições do sistema físico adotado. A segunda etapa consiste no desenvolvimento conceitual do projeto, onde permite-se o desenvolvimento teórico dentro de uma arquitetura ideal. Através de cálculos e simulações, avalia-se ideias e conceitos previamente determinados.

Portanto, para o prosseguimento deste trabalho são levantadas informações sobre: microcontroladores e suas respectivas linguagens de programação; visão computacional e suas diversas técnicas de segmentação; elaboração de aplicativos para dispositivos móveis e sua respectiva comunicação com servidores; interatividade entre dispositivos e transmissão

de informações com mínimo ruído associado. Os parâmetros obtidos são de fundamental importância para o desenvolvimento do projeto físico e determinação dos materiais e softwares a serem utilizados, tais como, posicionamento do dispositivo em vias públicas e seleção de câmera e microcontrolador, definição de plataforma de elaboração para banco de dados, interação entre aplicativo e banco de dados.

As listagens destas variáveis funcionais do projeto são de suma importância para aferição da execução pelo sistema de suas funções, e com isto, mensurar sua real capacidade de realização das tarefas propostas. Por sequência, é executada a fase de amostragem com testes funcionais de métodos propostos anteriormente, alguns destes são: posicionamento do dispositivo de captura de imagens em diversas amplitudes de luz, análise das variáveis impostas ao tratamento de imagem pela variação de amplitudes de luz, segmentação de imagem relacionada aos algarismos alfanuméricos, funcionalidades do aplicativo, transferência de dados obtidos para o banco de dados.

Dando continuidade, com todas as informações acumuladas, o aplicativo referente ao ambiente de cadastro de usuários e o banco de dados concernente aos dados obtidos dos mesmos, serão desenvolvidos, de mesma forma o sistema de reconhecimento de caracteres e comparação com dados previamente informados. Após o desenvolvimento dos itens anteriores a comunicação entre o banco de dados, câmera e sistema de análise é realizada.

As atividades foram realizadas nos laboratórios de Protótipos e Eletrônica do CEFET-MG, Campus V Divinópolis.

### **2.2.1 Características do Dispositivo**

Como pode ser visto na Figura 1, o dispositivo funciona da seguinte forma: Primeiramente o usuário cadastra-se através da plataforma de obtenção de dados, em seguida ele estaciona na vaga especial desejada, e o dispositivo faz o reconhecimento da placa do veículo através da visão computacional. Caso o usuário tenha autorização para utilizar a vaga, o dispositivo acende uma luz verde indicando que ele pode continuar naquela vaga, caso o usuário não tenha autorização para estacionar, o dispositivo acende uma luz vermelha e soa um alerta sonoro por cerca de 60 segundos, se o usuário ainda assim permanecer no local, o dispositivo tira uma foto da placa do veículo e essa foto é encaminhada para o órgão responsável por multar o veículo naquele município.

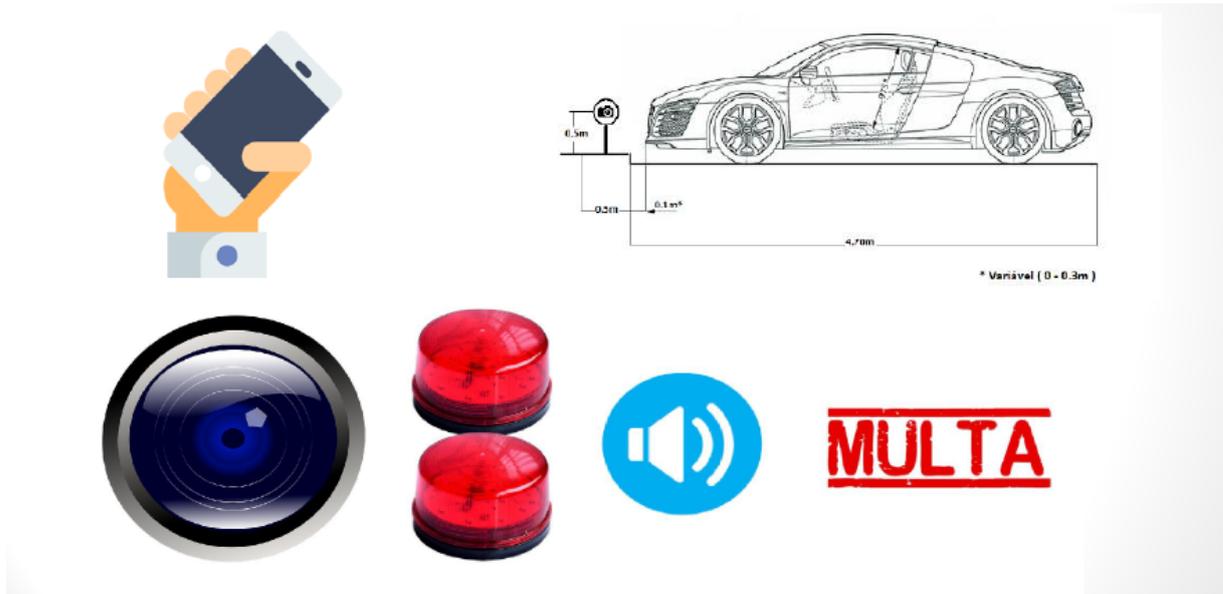


Figura 1 – Características do dispositivo.

## 2.3 Fundamentação Teórica

### 2.3.1 Visão Computacional

A visão computacional trata-se de um processo de modelagem e replicação da visão humana usando software e hardware. Este processo visa compreender uma cena 3D a partir de imagens 2D. Visão computacional e reconhecimento de imagens são termos frequentemente tratados como sinônimos, porém o primeiro não se limita apenas a se analisar imagens, envolve outras práticas.

### 2.3.2 OpenCV

A biblioteca OpenCV (*Open Source Computer Vision Library*) foi desenvolvida no ano de 2000 pela Intel. Ela se trata de uma biblioteca multiplataforma para o desenvolvimento de aplicativos na área de Visão computacional, de livre acesso ao uso acadêmico e comercial. A OpenCV possui diversos módulos de processamento de imagens e vídeo, além de mais de 350 algoritmos de Visão computacional como: filtros de imagem, calibração de câmera, reconhecimento de objetos, análise estrutural e outros.

### 2.3.3 Processamento de Imagem

O processamento de imagens é qualquer forma de tratamento de dados relativos a imagens ou quadros de vídeo. Ao contrário do tratamento de imagens, que visa a mani-

pulação de figuras para sua representação final. O processamento de imagens visa tratar dados de modo que estes possam viabilizar tarefas tais como a aprendizagem de máquina ou reconhecimento de padrões.

O processamento é dividido em três etapas nomeadas pré-processamento, realce e classificação. O pré-processamento se refere ao processamento inicial dos dados brutos para calibração radiométrica da imagem, correção de distorções geométricas e remoção de ruído. O realce visa evidenciar detalhes na imagem, permitindo assim uma melhor discriminação de objetos presentes nas imagens. Por fim, na classificação são atribuídas classes aos objetos presentes nas imagens.

#### 2.3.4 Filtros

(OpenCv (2018)) apresenta uma gama de conteúdo a respeito de filtros, relatando que estes podem ser aplicáveis através de funções e classes algorítmicas que resultam em filtragens lineares e não lineares em figuras 2D.

O processo de filtragem é feito utilizando matrizes denominadas máscaras, as quais são aplicadas sobre a imagem. A aplicação da máscara consiste em posicionar seu centro na posição  $(I, J)$ , sendo  $I$  o número de uma dada linha e  $J$  o número de uma coluna sobre a imagem. Então, o valor do pixel na posição  $(I, J)$  é substituído por um valor genérico obtido a partir dos valores encontrados nos pixels vizinhos.

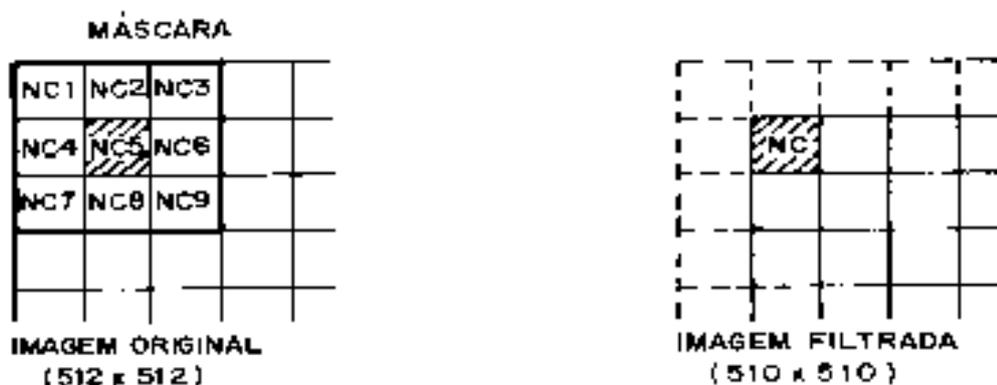


Figura 2 – Matriz Máscara

Os filtros lineares são responsáveis por suavizarem e realçarem detalhes da imagem e minimizarem os efeitos do ruído, sem alterar a média da imagem. Dentre os filtros lineares podemos citar os que mais se destacam como: Passa-baixa; Passa-Alta; Realce de borda; Realce não direcional; Realce de imagens.

Os filtros não lineares podem tanto minimizar quanto realçar ruídos e bordas, alterando a média da imagem, sendo os principais operadores para detecção de bordas e os filtros morfológicos. Dentre os filtros não lineares podemos citar os que mais se destacam como: Operador de detecção de borda; Operador de Roberts; Operador de Sobel.

### 2.3.5 Realce de Contraste

A técnica de realce de contraste tem como objetivo a melhora da qualidade de imagens por critérios subjetivos a visão humana. A manipulação do contraste consiste numa transferência radiométrica em cada pixel, com o objetivo de aumentar a discriminação visual entre os objetos presentes na imagem, esta operação é realizada pixel a pixel independente da vizinhança. O processo é complementado através da binarização da imagem, onde são atribuídos valores de '0' e '1' para cada pixel de acordo com sua condição de contraste.



Figura 3 – Binarização

### 2.3.6 Segmentação de Imagem

Segundo (Erpen (2004)), o reconhecimento de padrões é parte essencial de qualquer sistema de análise de imagens de alto nível. Um sistema de reconhecimento flexível deve ser capaz de reconhecer um objeto independentemente de sua orientação, tamanho e localização no campo de visão.

Na etapa de segmentação de imagens buscam-se informações de interesse ou que sejam básicas para a discriminação entre classes de objetos (GONZALES, 2000). Para tal é necessária a escolha da forma de representação dos dados, sejam de fronteiras externas estas determinadas por contornos, ou por regiões internas.

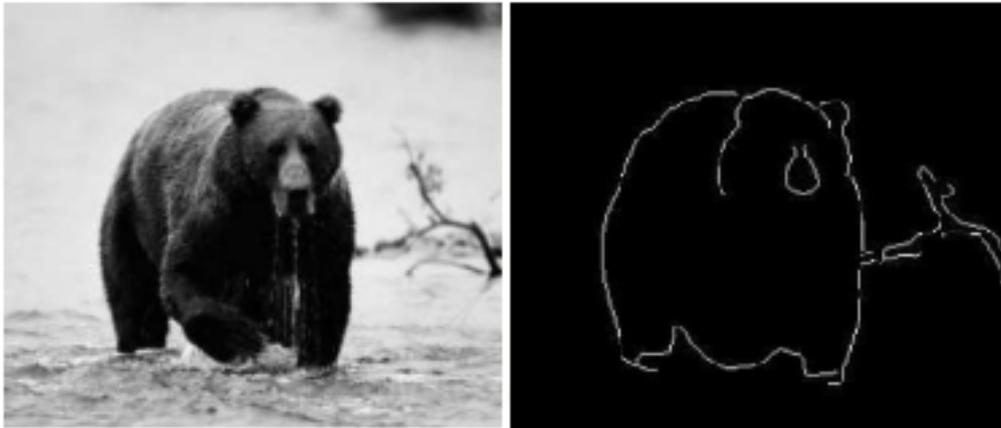


Figura 4 – Segmentação por borda externa



Figura 5 – Segmentação por região

A identificação por fronteira é mais adequada quando se visa identificar feições externas da forma, tais como contornos não preenchidos uniformemente. A identificação por regiões é mais efetiva quando se tratam de formas com características internas de interesse, tais como cores ou texturas. Contudo, nada inviabiliza a utilização de ambas as formas de identificação em conjunto para uma melhor solução da tarefa a ser realizada, lembrando que em qualquer um dos casos as características utilizadas como descritivas devem ser minimamente afetadas por variações como mudanças de tamanho, rotação ou translação.

### 2.3.7 Reconhecimento de Padrões

O reconhecimento em visão computacional trabalha com padrões, isto implica que os arquivos de vídeo ou imagem devem conter algo padronizado, que contemple um volume de variação dentro dos limites estipulados. Padrões podem ser aplicados em diversos componentes da figura, tais como: cor, forma, textura, distância de pontos, entre outros.

O processo de reconhecimento de padrões é comumente dividido em 5 etapas: Sensoriamento; Segmentação; Extração de características; Classificação e Pós-processamento.

### 2.3.8 OCR

Segundo (OpenCv (2018)), OCR ( *Optical Character Recognition* ) são arquivos da biblioteca openCV que possibilitam a identificação de caracteres em uma imagem ou mapa de bits sem necessidade de intervenção humana, ou seja, ela é capaz de extrair informações

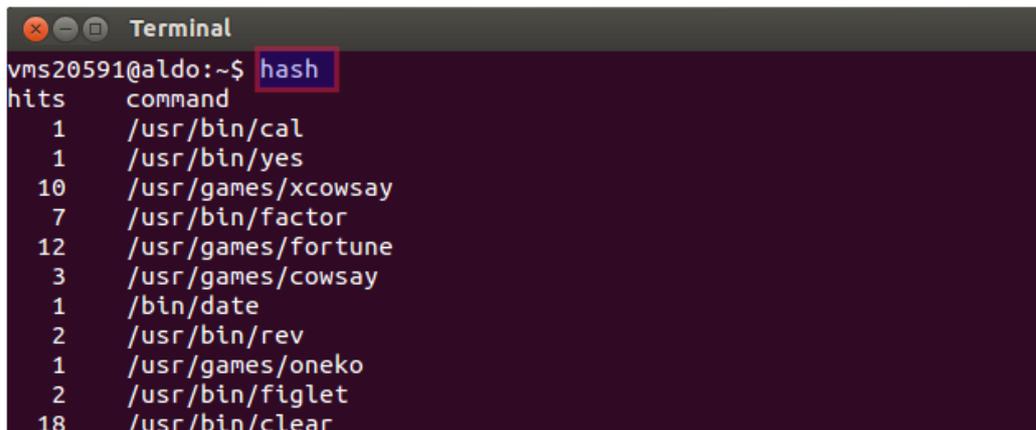
a partir de uma imagem qualquer. Um exemplo muito comum de aplicação do OCR são os radares de trânsito, onde câmeras de alta sensibilidade capturam a imagem do veículo e utiliza a biblioteca OCR para extração dos caracteres referentes ao veículo.

### 2.3.9 UBUNTU

(Ubuntu (2018)) o Ubuntu se trata de um sistema operacional de código aberto. A proposta do ubuntu é oferecer um sistema que qualquer pessoa possa utilizar sem dificuldades ou custos, independentemente de nacionalidade, nível de conhecimento ou limitações físicas.

Um dos focos principais é a usabilidade, incluindo o uso de diversas ferramentas para tarefas administrativas e a oferta de uma gama de recursos a partir de uma instalação padrão.

Embora hajam ferramentas gráficas para praticamente todos os programas presentes neste sistema operacional, a ferramenta ‘*Terminal*’ responsável por executar tarefas administrativas em todo o sistema via linhas de código é de grande valia quando se trata de compilação das mais diversas linhas de códigos em uma ampla gama de linguagem.



```
vms20591@aldo:~$ hash
hits  command
  1    /usr/bin/cal
  1    /usr/bin/yes
 10    /usr/games/xcowsay
  7    /usr/bin/factor
 12    /usr/games/fortune
  3    /usr/games/cowsay
  1    /bin/date
  2    /usr/bin/rev
  1    /usr/games/oneko
  2    /usr/bin/figlet
 18    /usr/bin/clear
```

Figura 6 – Ferramenta Terminal

### 2.3.10 Python

(Python (2018) ), Python é uma linguagem de programação criada por Guido Van Rossum em 1991. Os objetivos do projeto da linguagem eram: produtividade e legibili-

dade. Em outras palavras, Python é uma linguagem que foi criada para produzir código bom e fácil de manter de maneira rápida. Python tem uma biblioteca padrão imensa, que contém classes, métodos e funções para realizar essencialmente qualquer tarefa, desde acesso a bancos de dados a interfaces gráficas com o usuário.

Python é uma linguagem livre e multiplataforma, isso significa que os programas escritos em uma plataforma são executados sem nenhum problema na maioria das plataformas existentes e sem nenhuma modificação. E, caso a plataforma objetivo não tenha uma versão de Python, desenvolvedores têm a liberdade de estudar e modificar o código da linguagem para fazer com que ela rode onde quer que seja.

A plataforma Python é comumente nativa do Ubuntu, em casos excepcionais de inexistência dos pacotes, sua instalação é simples e de fácil alcance.

### 2.3.11 Web Design

O web design é uma extensão da prática do design gráfico, onde o foco do projeto é a criação de web sites e documentos disponíveis no ambiente da World Wide Web.

O web design tende à multidisciplinaridade, uma vez que a construção de páginas web requer subsídios de diversas áreas técnicas, além do design propriamente dito. Áreas como a arquitetura da informação, programação, ergonomia, usabilidade, acessibilidade entre outros.

A preocupação fundamental do web designer é agregar os conceitos de usabilidade com o planejamento da pessoa em destaque, garantindo que o usuário final atinja seus objetivos de forma agradável e intuitiva.

### 2.3.12 Placa de identificação Veicular

A partir de 1990, foi adotado o Sistema renavam, este sistema consiste no padrão tipo ABC-1234, sendo que cada Estado passou a ter seus intervalos de combinações de letras. Cada veículo recebe uma combinação alfanumérica que o acompanhará mesmo se posteriormente ele for licenciado em outro Estado.

A tipologia e dimensões das placas de identificação podem ser observadas nas seguintes imagens.

**I234567890**  
**ABCDEFGHIJKLM**  
**NOPQRSTUVWXYZ**

Figura 7 – Tipologia dos caracteres



Figura 8 – Dimensões da Placa de identificação

### 2.3.13 Banco de Dados (SQLite)

Programadores Java que visam desenvolver aplicativos móveis para o Android podem se beneficiar imediatamente do uso do SQLite em seus programas, pois se trata de um banco de dados existente em qualquer instalação do sistema operacional móvel do Google. A criação de programas que utilizem este banco é facilitada por um conjunto poderoso de classes existentes no Android SDK, que dão ao desenvolvedor amplo controle sobre o armazenamento dos dados (DEV MEDIA, 2018).

Uma das escolhas certas do Google ao elaborar o Android SDK, foi embutir o pequeno (mas poderoso) banco de dados SQLite no sistema operacional. Com ele, os desenvolvedores podem armazenar dados no aparelho de maneira rápida e eficiente, sem nenhuma necessidade de bibliotecas adicionais, pois existem classes específicas dentro do Android SDK para uso direto do SQLite.

Para quem não conhece o SQLite, o seu nome pode soar um pouco desanimador. Afinal de contas, a palavra “lite” sempre dá a ideia de algo reduzido, menor, mais limitado. Isso realmente pode se aplicar ao SQLite em virtude do fato de ele ser realmente pequeno em

tamanho e quantidade de memória que ocupa, mas não em relação a sua capacidade e desempenho.



Figura 9 – SQLite

### 2.3.14 Comunicação Wireless

As chamadas comunicações sem fios são o segmento de mais rápido crescimento no setor de comunicações. Além da utilização em dispositivos móveis e residências, há diversas utilizações em meio industriais: redes de sensores sem fio, rodovias e fábricas automatizadas, casas e eletrodomésticos inteligentes e telemedicina. Há diversos meios de comunicação Wireless, dentre ele pode-se destacar: o Bluetooth, o Wi-Fi e o ZigBee.

#### 2.3.14.1 Bluetooth

Bluetooth, nome em homenagem ao rei da Dinamarca entre 940 e 985 DC, Harald I Bluetooth, que uniu a Dinamarca e Noruega, isso, devido ao fato que o Bluetooth propõe unir dispositivos através de conexões de rádio, daí surge a inspiração para o nome. Estes raios fornecem conexão de curto alcance entre dispositivos sem fio, juntamente com redes rudimentares. O Bluetooth é o principal meio de comunicação de baixo alcance, operando em uma faixa de frequência de 2,4GHz, portanto, pode ser utilizada sem licença. Esta comunicação fornece um canal de dados assíncronos em 723.2 Kbps. Este, também utiliza saltos de frequências para acesso múltiplo com um espaçamento de suporte de 1MHz, normalmente, até 80 frequências diferentes são usadas para uma largura de banda de 80MHz (GOLDSMITH, 2005).

(Lee, Su e Shen (2007)) já descrevem os termos técnicos desta comunicação, ressaltando que o Bluetooth também é conhecido como padrão IEEE 802.15.1 com base em um

sistema de rádio sem fio projetado para curto alcance e dispositivos baratos para substituir canais para periféricos de computadores, tais como: Mouses, teclados, joysticks e impressoras. O autor descreve que essa gama de aplicativos são conhecidos como rede de área pessoal sem fio.

#### 2.3.14.2 Wi-Fi

Wi-Fi é um padrão de rede de área local desenvolvido pelo grupo de trabalho IEEE 802.11 e foi projetado para ser usado em ambientes fechados, para distribuir o acesso à Internet a diversos computadores em uma casa ou em um escritório. (Lee, Su e Shen (2007)) diz que Wi-Fi possui padrões para redes de área local sem fio (WLAN). Esta, permite que usuários naveguem na Internet a velocidades de banda larga quando conectados por um ponto de acesso (AP). A arquitetura do Wi-Fi consistem em vários componente que interagem para fornecer uma rede sem fio que suporte a mobilidade da estação de forma transparente para as camadas superiores.

#### 2.3.15 Câmera de Transmissão Sem Fio

As câmeras são equipamentos destinados a converter níveis de iluminação e cor em sinais elétricos, seguindo certo padrões. Todas as câmeras possuem elementos (sensores) os quais são atingidos pela luz. Todo o sistema de visualização tem como ponto de inicio a câmera. A câmera cria a imagem CCD, essa imagem capturada é então processada e transmitida para o sistema de controle, como um quad, multiplexador ou DVR. Atualmente existem diversos tipos diferentes de câmeras projetados para aplicações e ambientes específicos. Existem micro câmeras para aplicações simples, câmeras profissionais para aplicações de grande porte e grande versatilidade, entre outras (PERES, 2006).

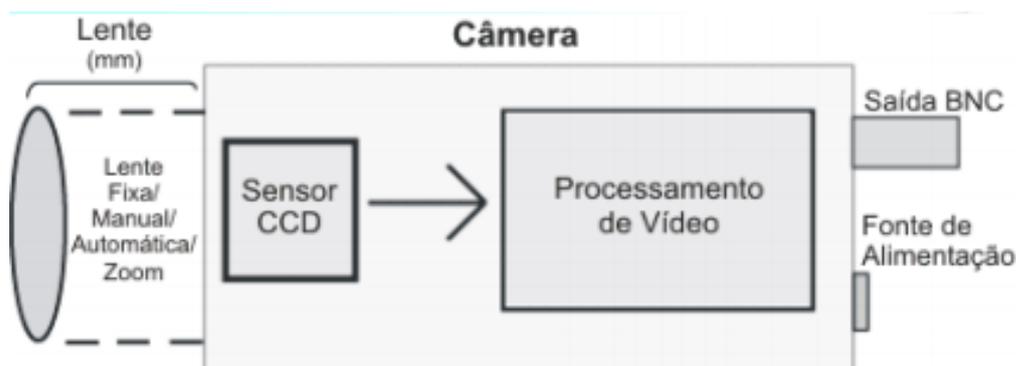


Figura 10 – Sistema da câmera

Quando existe dificuldade de passar o cabeamento para transmissão do sinal de vídeo da câmera para o monitor, uma das alternativas a serem consideradas é a transmissão sem fio. Porém, os sinais sem fio também podem sofrer uma série de atenuações ou interferências, pois todo o sinal de radiofrequência é atenuado ou bloqueado por paredes ou estruturas de metal pesado, linhas de transmissão de alta tensão, equipamentos de microondas e outros.

Atualmente, os sistemas sem fio mais comercializados são:

- a) Frequência de 900MHz com uma alcance de transmissão de 100 metros,
- b) Frequência de 2.4Ghz com uma alcance de transmissão de 100 metros,e
- c) Frequência de 2.4Ghz com uma alcance de transmissão de 250 metros.



Figura 11 – Câmera de transmissão sem fio com infravermelho.



## Desenvolvimento

De acordo com os fundamentos apresentados, pôde-se obter conhecimentos necessários para se iniciar o desenvolvimento deste projeto. Tais conhecimentos foram empregados na determinação das condições de operação, características físicas e as especificações do projeto.

Este projeto foi dividido em etapas de desenvolvimento, são elas: elaboração de software de reconhecimento; aplicação móvel de cadastro de usuários; comunicação entre dispositivos não cabeados; desenvolvimento de protótipo para obtenção de imagens e sinalização e por fim, o processo de junção entre todos os processos anteriores.

### 3.1 Elaboração do Algoritmo de Reconhecimento

#### 3.1.1 Interação entre OpenCv e sistema operacional

Inicialmente, foram efetuadas buscas por uma interface de programação que melhor colaborasse para o livre desenvolvimento da aplicação. A interface selecionada foi o sistema operacional *UBUNTU 16.4*, escolha esta justificada por se tratar de uma plataforma que tem como compromissos a isenção de qualquer custo de software e a livre plataforma de desenvolvimento técnico sem o empecilho de direitos do desenvolvedor. Neste sistema operacional é desnecessária a instalação de qualquer ambiente de programação, seu próprio terminal de comandos é capaz de compilar arquivos com o auxílio da instalação de pacotes de extensões.

Para desenvolvimento do algoritmo, a linguagem de comandos escolhida é a *PYTHON*, esta definida por sua larga amplitude de documentação e simplicidade de execução. A ver-

são Ubuntu 16.4 já conta com a biblioteca Python nativa, simplificando assim o processo de instalação, o que tornou necessário apenas a instalação da biblioteca básica OpenCV e algumas de suas extensões de maior complexidade. Dentre eles podemos citar a título de exemplo, o pacote de execução de cálculos matemáticos Numpy e o de reconhecimentos de caracteres em padrão internacional OCR.

#### 3.1.2 Padrões técnicos para obtenção da imagem

A detecção de placas em uma imagem demanda uma série de análises sobre diferentes características. A eficiência em termos de tempo é de suma importância em cada uma destas etapas.

Para uma melhor elaboração do algoritmo, características físicas do projeto podem ser dimensionadas, de modo que, facilitem a obtenção das variáveis necessárias para funcionamento do mesmo, poupando assim custos computacionais e conseqüentemente tempo. Para um dimensionamento vantajoso das características de obtenção de imagem, são levadas em consideração o posicionamento da câmera em relação ao veículo fiscalizado e a possível altura da placa de acordo com as variações estruturais dos veículos.

De acordo com (Park (2018)), a vaga média de estacionamento nas cidades brasileiras consiste em 4,70 metros de comprimento por 2,20 metros de largura.

Em pesquisas in loco foi possível averiguar que a altura máxima de posicionamento da placa dianteira de identificação em veículos originais se tratava de 0,7 metros e de mesma forma a altura mínima observada 0,3 metros.

De acordo com os dados obtidos, para um melhor alcance e simplificação de remoção de zonas mortas ( zonas com ausência de dados relevantes ao interesse da visão ) presentes nas imagens, são definidos os padrões de fixação da câmera em relação a vaga, como se pode observar na figura 12.

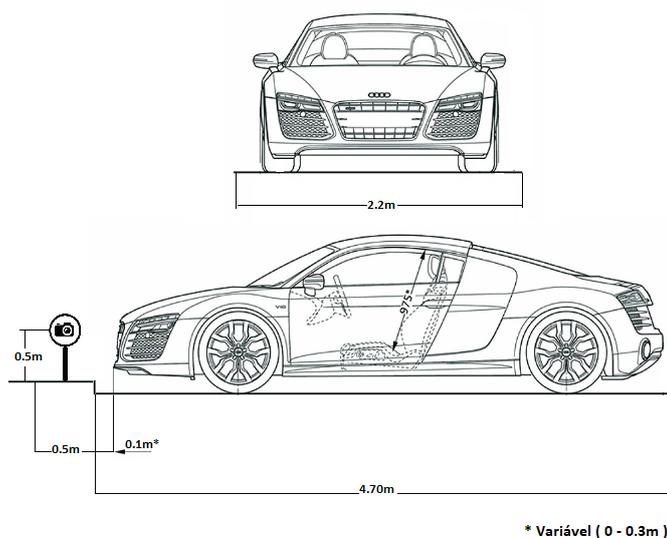


Figura 12 – Distâncias padrão de instalação da câmera

### 3.1.3 Extração de imagem

Nesta etapa são utilizadas imagens subtraídas de um banco de dados com imagens no padrão de distâncias estabelecidos anteriormente. O dispositivo utilizado para execução das imagens foi um smartphone modelo 5s com resolução 3264x2448 pixels.

O acesso ao banco de imagens é realizado utilizando funções da biblioteca OpenCV, inicialmente são chamadas apenas as funções necessárias para realização desta tarefa. A extensão *cv2* torna possível a abertura, manipulação e salvamento de imagens.

Através do método *cv2.imread("imagem.formato",X)* é possível 'chamar' uma imagem presente no banco de dados e atribuí-la a uma variável qualquer no algoritmo, os parâmetros necessários para este método são o nome atribuído à esta imagem no banco de dados seguido por seu formato, representados no exemplo anterior por *'imagem.formato'*. Além disso é possível determinar o parâmetro de cores RGB (*Red Green Blue*) através do valor de *'X'* também representado no exemplo anterior, neste caso utilizaremos o valor de *'X'* como 0, pois, este corresponde à representação somente em escala de cinza, etapa que se torna ideal mais a frente do processo.



Figura 13 – a - imagem original. b - imagem após emprego do método.

#### 3.1.4 Descarte de zona de não interesse

Através dos padrões de posição da câmera é possível se estimar a zona de interesse da imagem.

Neste trabalho, com o padrão de câmera proposto e o estudo de altura da placa de identificação de automóveis é possível determinar matematicamente quais pixels podem ser descartados por não constarem na área de interesse da figura.

O corte em imagens é orientado via a posição dos pixels em sua matriz, deve-se também observar que a posição cartesiana na biblioteca *openCV* é invertida, ou seja,  $(y,x)$ , portanto, se torna necessário um maior zelo com estas operações. A operação de Corte é extramente simples quando se sabe a região a ser destacada, com a extensão *cv2* ativada, o corte é executado com o simples comando `'var=img[y:(y+h), x:(x+k)']`, onde `'var'` representa a variável de alocação da imagem cortada, `'img'` define a imagem a ser cortada, `'y'` e `'x'` são respectivamente as posições da coluna inicial e linha inicial do corte, e por fim temos `'h'` e `'k'` que definem o valor de colunas e linhas internas que devem ser somadas a área de corte.

#### 3.1.5 Binarização

O método de binarização consiste em encontrar um limiar entre um histograma de 256 níveis de cinza que melhor separa as fases associadas em questão, portanto ele evidencia

as características que deseja-se detectar na imagem (os algarismos e a placa).

Para aplicação deste método se utilizou da função `cv2.threshold(corte,240,255,cv2.THRESHBINARY)` que transforma e armazena os dados da imagem em uma variável definida previamente, abaixo se pode observar uma imagem com características binárias.



Figura 14 – Imagem binaria.

### 3.1.6 Destacamento da figura de interesse

Após a execução da binarização a característica retangular da placa é evidenciada, ou seja, é passível de identificação. Para localização da placa se utiliza de padrões que caracterizam os objetos de interesse, neste caso podemos referenciar como padrões o formato e as medidas do retângulo.

Utiliza-se um filtro que neste caso específico localiza todos os possíveis retângulos presentes na imagem analisada e em seguida utiliza uma máscara com o tamanho esperado do objeto convertido em pixels. Com um objeto semelhante encontrado a posição de seus pixels é armazenada, e com um comando de corte simples a placa é removida da imagem e armazenada em outra variável.

### 3.1.7 Abertura de imagem

A abertura é obtida pelo encadeamento do filtro de erosão, seguido pelo de dilatação.

O filtro de erosão observa a vizinhança de cada pixel reduzindo a escala de cor em caso de vizinhança neutra, tornando assim a imagem cada vez mais sucinta e livre de ruídos e ilhas a cada aplicação do filtro, de forma inversa o filtro de dilatação aumenta a escala de cor determinada pela vizinhança.

Para aplicação do filtro de erosão foi utilizado o seguinte comando '*cv2.erode(dilation, kernel, iterations = X)*', para aplicação do filtro de dilatação o comando é '*cv2.dilate(corte2, kernel, iterations = X)*' onde para ambos o valor X é substituído pelo valor de vezes de aplicações seguidas do filtro. De acordo com os testes o numero indicado de aplicações para obtenção de um resultado satisfatório são 3 aplicações para erosão seguidas de 4 aplicações de dilatação.

Podemos observar abaixo a aplicação dos filtros.

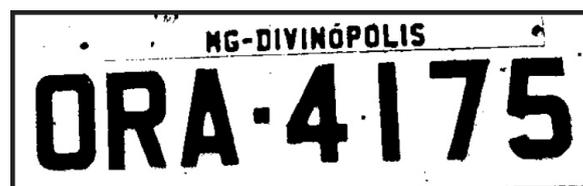


Figura 15 – Imagem original.

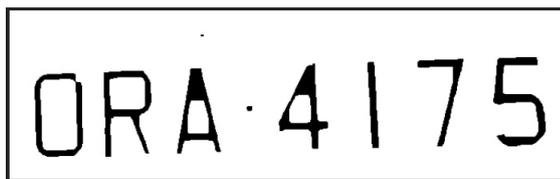


Figura 16 – imagem com erosão.

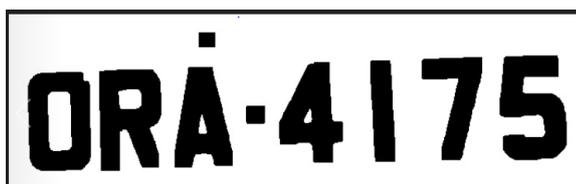


Figura 17 – imagem com erosão e dilatação.

### 3.1.8 Corte e isolamento dos caracteres alfanuméricos

Com a placa devidamente identificada em seu contorno, é possível apontar a posição dos caracteres dentro da placa.

A posição dos caracteres dentro de placas veiculares é tabelada nacionalmente, portanto cada letra ocupa o mesmo espaço físico das demais, e também a distância entre a borda da placa e o início do caractere são equivalentes para as posições análogas, logo, com o recorte da placa em sua integridade é possível localizar com precisão cada caractere e assim isolá-los para facilitar sua identificação pela biblioteca responsável.

Através de cálculos que relacionam a posição da câmera e o número de pixels da figura é possível definir na matriz de pixels  $(y,x)$  as posições de corte para isolamento dos caracteres.

Optou-se neste presente trabalho pelo isolamento dos caracteres alfabéticos e numéricos em duas imagens, pois através do desenvolvimento desta prática é notório que o isolamento de caractere por caractere representava um trabalho desnecessário já que a identificação de caracteres isolados não apresentava melhora significativa para tal trabalho.

Com as posições definidas na matriz, o corte é efetuado através do código citado anteriormente para ambos os tipos de caracteres, podemos observar subsequentemente os referentes cortes.



Figura 18 – Caracteres alfabéticos isolados.



Figura 19 – Caracteres numéricos isolados.

### 3.1.9 Identificação de Caracteres

A identificação de caracteres se trata de uma atividade com um grau de complexidade superior, onde todas as etapas apresentadas anteriormente são de suma importância para que esta atividade tenha sucesso.

As pesquisas se iniciaram com a busca de uma biblioteca interna a OpenCV com capacidade teórica de identificação de caracteres, esta se tratou da *tesseract ocr*, biblioteca com capacidade de reconhecimento de padrões de caracteres no padrão *ASCII*.

A biblioteca trabalha analisando imagens e identificando caracteres que estejam evidenciados. Um padrão não uniforme de cor ou uma taxa excessiva de ruído no caractere representa uma barreira a identificação do mesmo e conseqüentemente o retorno de uma leitura defeituosa.

Quando tratamos da identificação de placas de veículos, várias são as possibilidades de influência negativa nos resultados, estas variam de mal estado de conservação até a tarjeta de identificação de município.

Os tratamentos de imagem apresentados anteriormente ficam a cargo de remoção de possíveis falhas na imagem e evidenciação do conteúdo a ser identificado, assim, cortando drasticamente a quantidade de erros presentes das identificações.

O padrão de caracteres utilizado em placas de identificação automotiva conta com falhas estruturais relativas a identificação, a mais evidente ocorre com a semelhança entre caracteres alfabéticos e numéricos, como exemplo podemos observar o numeral '1', este

apresenta alta semelhança com a letra em caixa baixa 'l', portanto, causando certo conflito de identificação pela biblioteca, com isso, a necessidade de segregação de caracteres numéricos se torna vital, pois, assim essas falhas de identificação podem ser evitadas.

## 3.2 Aplicação Móvel de Cadastro de Usuários

### 3.2.1 Criação do Aplicativo

Para o desenvolvimento do aplicativo de interação com o usuário, o primeiro passo é pesquisar qual seria a plataforma de programação mais adequada para a implementação de tal aplicação. Depois de alguns estudos e testes feitos, foi decidido a elaboração de um site de cadastro.

A plataforma de um site é acessível a qualquer usuário com acesso a internet de forma simples e rápida, sem a necessidade de download de qualquer arquivo da rede.

### 3.2.2 Ferramentas Utilizadas

A IDE (Ambiente de Desenvolvimento Integrado) de desenvolvimento web proporciona ao desenvolvedor, várias ferramentas capazes de arquitetar o layout de um web site. No ambiente temos etapas pré dispostas onde um modelo é fornecido para alterações que caracterizem o web site de acordo com o requerido pelo usuário, tais etapas variam de arquitetura (onde o site é moldado) até o modo de obtenção de informações do usuário.

### 3.2.3 Interface Gráfica

A acessibilidade e o fácil entendimento da interface são os requisitos de maior importância para a construção do design do site, visando a possibilidade de que o usuário pode não possuir uma certa familiaridade com o manuseio de tais aplicações. O projeto contém 8 classes do tipo `activity_main.xml` que são responsáveis pelo layout do aplicativo, cada tela possui sua devida funcionalidade de interação com o usuário, são elas:

Tela Inicial: A tela inicial é onde acontece o primeiro contato com o usuário, ela possui um `TextView` de boas vindas e 4 `Buttons` para o usuário escolher entre fazer cadastro, consultar normas, como funciona e central de ajuda.

Tela Cadastro: Caso o usuário clique em fazer cadastro na tela inicial, esta tela aparecerá, e nela possui 3 Buttons que são as opções referentes a classe a qual o usuário pertence, são elas idoso, portador de deficiência ou gestante.

Tela do Idoso: Essa tela é referente ao cadastro do idoso, que possui campos do tipo Plain Text, onde ele deve fornecer informações como nome completo, idade, nº de registro e placa do veículo.

Tela do Portador de Deficiência: Essa tela refere-se ao cadastro do portador de deficiência, que possui campos do tipo Plain Text, onde ele deve fornecer informações como nome completo, idade, nº de registro e placa do veículo.

Tela da Gestante: Essa tela é referente a gestante, que possui campos do tipo Plain Text, onde ela deve fornecer informações como nome completo, idade, data do término da gestação e placa do veículo.

Tela Normas: Essa tela é responsável por passar para o usuário todas as normas referentes ao funcionamento tanto do aplicativo quanto do dispositivo que se encontra nas vagas especiais.

Tela Como Funciona: Essa tela é responsável por explicar ao usuário como é o funcionamento do dispositivo, tirando todas as dúvidas que o usuário possa ter em relação ao dispositivo e ao cadastro no aplicativo.

Tela Central de Ajuda: Essa tela é responsável por prestar ajuda ao usuário, caso ocorra algum erro no cadastro ou alguma dúvida referente ao projeto.

Na figura 22 a seguir podemos ver uma simulação das telas com a interface na aplicação.

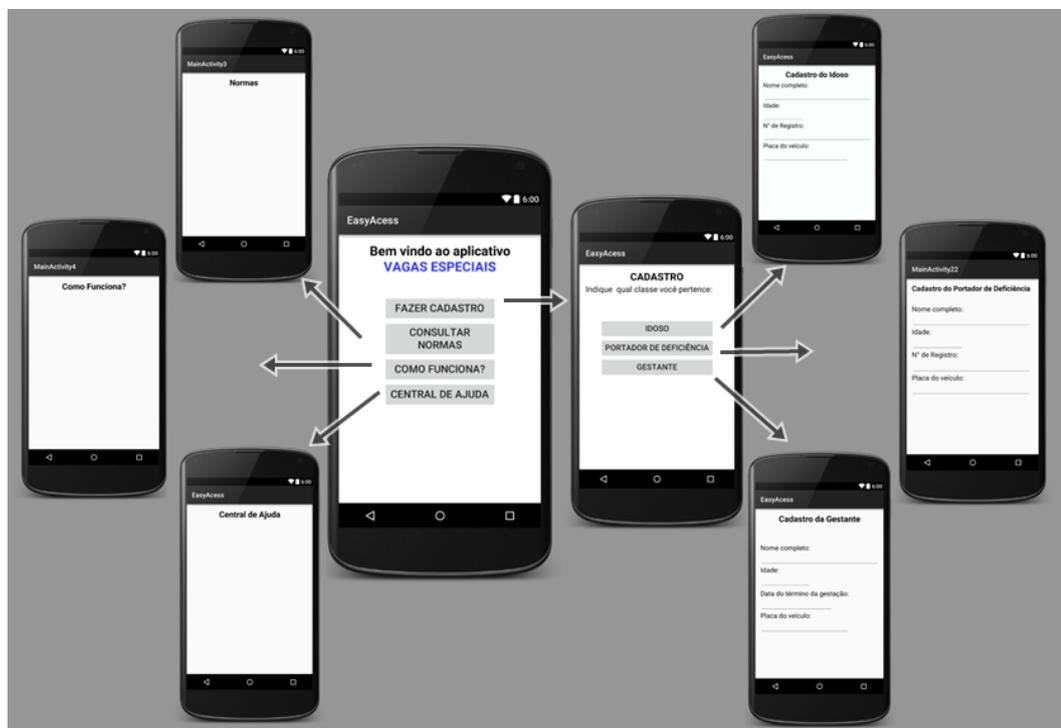


Figura 20 – Layout do Aplicativo.



## Resultados

A partir deste capítulo são apresentadas as descrições dos projetos, testes e experimentos realizados, acompanhados dos seus respectivos resultados.

### 4.1 Desenvolvimento WEB

O desenvolvimento da plataforma de cadastro web é executado a partir de uma plataforma interativa online monetizada, nesta, todas as texturas e funcionalidades do site puderam ser testadas, inclusive com o site momentaneamente na web.

Na elaboração do site, critérios de cadastro para cada tipo de usuário são definidos de acordo com barreiras para que usuários não autorizados não burlarem o sistema, é importante salientar que uma base de apoio para consulta da veracidade dos dados cadastrados é necessária, base esta que existe em poder das agências governamentais responsáveis e como tal é de acesso restrito.

Os critérios de cadastro se definiram como:

**Para Idosos:**

- Nome.
- E-mail.
- Telefone de Contato.
- Numero do documento de identidade.
- Idade.

- Placa do veículo.
- RENAVAM (numeração individual atribuída a cada veículo.).
- Numeração do chassi.

### **Para Gestantes:**

- Nome.
- E-mail.
- Telefone de Contato.
- Período de Gestaçãõ.
- Idade.
- Cadastro pré natal,
- Placa do veículo.
- RENAVAM (numeração individual atribuída a cada veículo.).
- Numeração do chassi.

### **Para Deficientes:**

- Nome.
- E-mail.
- Telefone de Contato.
- Cadastro municipal de deficiente físico.
- Idade.
- Placa do veículo.
- RENAVAM (numeração individual atribuída a cada veículo.).
- Numeração do chassi.

O site elaborado se trata de uma aplicação responsiva, ou seja, esta se molda de acordo com a plataforma utilizada para acessá-lo, então seja em um smartphone ou um PC de mesa a plataforma atende os usuários da mesma forma.

A versão final do web site aberto via smartphone ficou da seguinte forma:



Figura 21 – Pagina Home

Clicando no ícone 'menu' que é caracterizado pelos três traços horizontais na parte superior direita do *print* presente na figura 20 temos acesso as abas de cadastro como pode se observado abaixo.

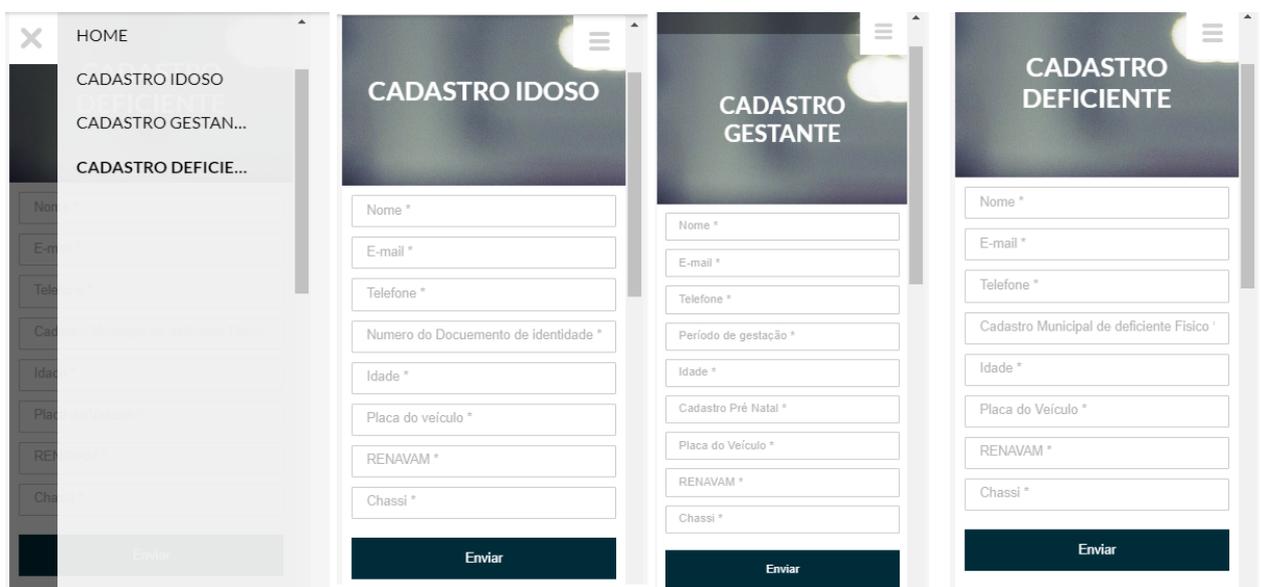


Figura 22 – Aba de cadastro de usuários

De forma análoga podemos observar a pagina inicial do site aberto via um PC comum.

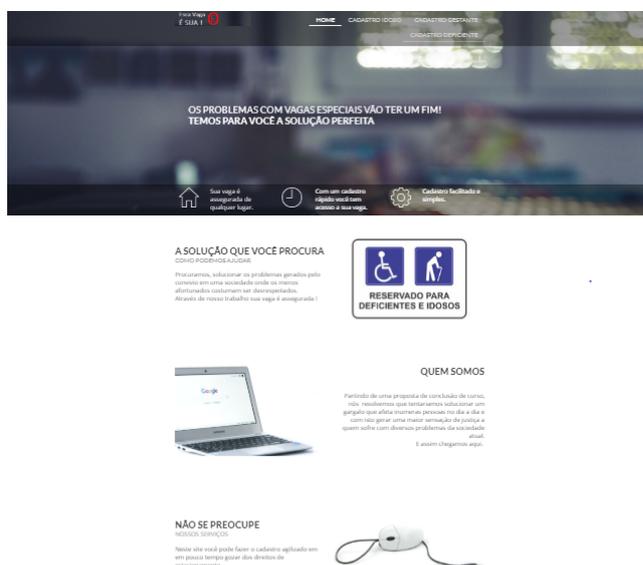


Figura 23 – Home aberta via PC

Os menus apresentam o mesmo formato ao apresentado na versão via smartphone.

No que se trata do acesso às informações coletadas no site, estas são enviadas para um servidor de e-mail que faz a coleta dos dados e armazenamento no banco de dados. É importante salientar que como a ativação tanto do site quanto do servidor de e-mail próprio é inviável financeiramente, por tanto essa funcionalidade está defasada e é necessária a inserção manual dos dados no banco de dados.

## 4.2 Aplicação de técnicas de visão computacional e algoritmo de identificação

Conforme apresentado nos capítulos anteriores diversas etapas de tratamentos de imagem são aplicadas para o reconhecimento dos caracteres. Contudo ainda é necessário unir estas aplicações a uma rotina de programa capaz de executar a análise da placa e analisar os dados presentes no banco de dados. Para isso escrevemos um código que além de executar as rotinas de identificação dos caracteres executa a comunicação entre o banco de dados, análise de cadastro, e execução de sinais de identificação.

O programa executado pode ser resumido com as seguintes etapas e seus respectivos resultados:

- **Localização da câmera e captura de imagem.**

O index da câmera é identificado e a mesma acionada, subsequentemente a imagem é capturada e salva no sistema em uma variável padrão.

- **Aplicação de Filtros.**

Em uma segunda etapa, os filtros de imagem são aplicados a variável padrão com base na teoria proposta em capítulos anteriores e com isto as imagens obtidas também foi armazenadas com um nome padrão.

- **Segmentação.**

Adiante a imagem tratada é editada de modo a realizar a remoção das zonas de não interesse. De acordo com os padrões da câmera utilizada foi executada entre os frames [200:400, 200:1010].

- **Identificação.**

Para a etapa de identificação de caracteres a variável com a última alteração feita é submetida a biblioteca *OCR-openALPR*, o código fonte executa a análise sobre a imagem e retorna o *string* no formato 'AAA1111' (três letras seguido de quatro numerais) ou uma *string* vazia em caso de haver veículo com placa na imagem obtida.

- **Banco de Dados.**

Em um caso onde a identificação é positiva, a placa encontrada é submetida a uma consulta no banco de dados do sistema e este retornou se a placa estava devidamente cadastrada.

- **Elo de veículo ausente.**

Como o algoritmo trabalha de forma interativa fazendo análises constantes de imagens capturadas em pequenos intervalos, quando não houve identificação de placas ele entrou em um elo que sinaliza de forma interna (mensagem em tela) que não há veículo na vaga, e este seguido de um pequeno delay até a próxima avaliação.

- **Elo de veículo cadastrado.**

Com a placa identificada, e esta devidamente cadastrada no banco de dados de acesso, o algoritmo entra em um elo de veículo cadastrado onde há uma sinalização interna (mensagem em tela) e outra externa. Por sinalização externa temos: a sonora que se tratou do soar da seguinte mensagem "Veículo autorizado" e a visual onde luzes de identificação foram acionadas.

- **Elo de veículo não cadastrado.**

Também com a placa identificada, e a ausência de cadastro o algoritmo entra em um elo de três etapas: Primeiro se trata de uma notificação sonora onde o usuário estacionado é avisado da situação irregular e indicado a retirar seu veículo em um tempo  $T$ .

A Segunda notificação é gerada no intervalo  $T/2$  onde o usuário é alertado que seu tempo está se encerrando.

Por fim uma notificação sonora é gerada informando da autuação do veículo e a placa identificada é então cadastrada em um banco de placas autuadas. A identificação luminosa é intermitente durante todo o elo.

- **Algoritmo de autorização de acesso**

Com a placa alocada em determinada variável do tipo *string*, é realizada uma consulta as linhas do banco de dados e este retorna ao programa um valor lógico *true* ou *false* ao sistema, e de acordo com este, a resposta é gerada. Este sistema foi simulado utilizando como servidor para o *root* do programa um notebook Dell inspiron.

Podemos ver nas Figuras 24 e 25 a compilação do programa no sistema operacional Ubuntu e a respectiva identificação da placa para tal leitura, e os dois casos, um de autorização e um de negação do acesso a vaga.

```
ronaldo@ronaldo-PC: ~/Documentos/agoraval
##### FINALIZA PROGRAMA #####
ronaldo@ronaldo-PC:~/Documentos/agoraval$ python agoravat.py
HIGHGUI ERROR: V4L: index 1 is not correct.

##### L E I T U R A #####
HJY-4547
PLACA OTIMIZADA:
HJY4547

##### CONSULTA BANCO DE DADOS #####
VEICULO AUTORIZADO
PLACA LIBERADA
##### FINALIZA PROGRAMA #####
```

a



b

Figura 24 – (a) - compilação do algoritmo com placa cadastrada no banco de dados (b) - placa identificada pela câmera

```
ronaldo@ronaldo-PC: ~/Documentos/agoraval
ronaldo@ronaldo-PC:~/Documentos/agoraval$ ^C
ronaldo@ronaldo-PC:~/Documentos/agoraval$ ^C
ronaldo@ronaldo-PC:~/Documentos/agoraval$ python agoravat.py
HIGHGUI ERROR: V4L: index 1 is not correct.

##### L E I T U R A #####
AWB-9874
PLACA OTIMIZADA:
AWB9874

##### CONSULTA BANCO DE DADOS #####
ATENCAO: VEICULO NAO AUTORIZADO
PLACA REGISTRADA PARA AUTUACAO
ALERTA ACIONADO
##### FINALIZA PROGRAMA #####
```

a

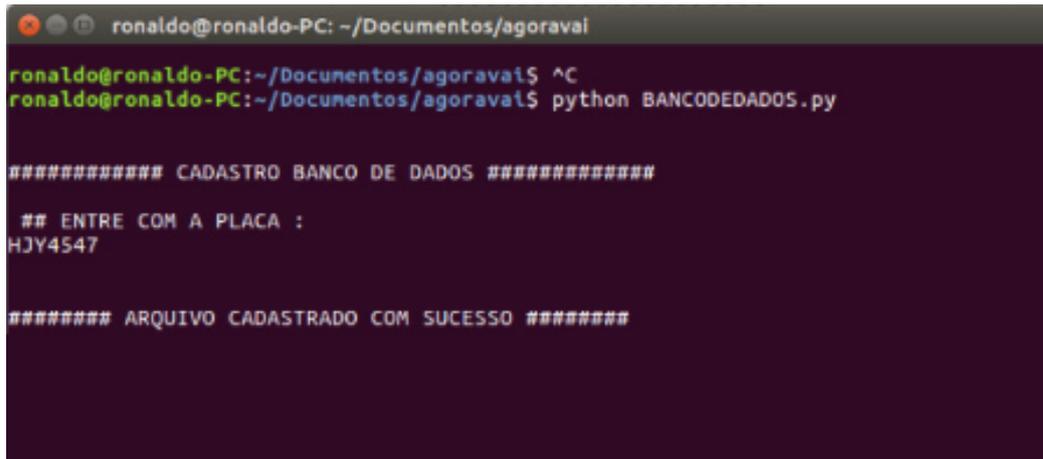


b

Figura 25 – (a) - compilação do algoritmo com placa não cadastrada no banco de dados (b) - placa identificada pela câmera

É criado também um algoritmo para cadastro direto no banco de dados onde se pode cadastrar diretamente no sistema qualquer placa desejada.

Quando o algoritmo é compilado ele requiere a placa do veículo e com isto ela é automaticamente cadastrada e liberada para estacionar em qualquer vaga. A interface compilada deste algoritmo pode ser observado na Figura 26.



```
ronaldo@ronaldo-PC: ~/Documentos/agoraval
ronaldo@ronaldo-PC:~/Documentos/agoraval$ ^C
ronaldo@ronaldo-PC:~/Documentos/agoraval$ python BANCOEDADOS.py

##### CADASTRO BANCO DE DADOS #####

## ENTRE COM A PLACA :
HJY4547

##### ARQUIVO CADASTRADO COM SUCESSO #####
```

Figura 26 – Cadastro de placa no banco de dados

- **Repetição.**

Para sensoriamento de presença o algoritmo utiliza a própria câmera. Para isto o algoritmo trabalha de forma recursiva onde analisa incessantemente as imagens geradas com base em delays temporais. Tal processo se apresenta vantajoso por eliminar possíveis sensores porém existe certo custo energético para funcionamento quase que incessante do sistema.

### 4.2.1 Identificação de placas

São impressas diversos caracteres no padrão de placas de veículos automotivos e estas submetidas a análise via o software desenvolvido, podemos observar então os seguintes resultados para uma amostra de 20 placas.

Para a amostra demonstrada na Tabela 1 temos as seguintes porcentagens de resposta: 45% acertos, 35% erros de identificação e 20% de leituras inconclusivas. Como erros de identificação podemos definir enganos na análise de letras, estes erros se tornam mais comuns com placas que contenham letras com traços de certa forma semelhantes, como exemplo podemos citar *E* em relação a *F* ou *T*, de mesma forma com os numerais *9* em relação a *8* ou *6*.

Os casos inconclusivos se relacionam a não identificação de caracteres ou a identificação de caracteres fora do padrão 3 letras 4 números.

Os resultados apresentados na tabela anterior apresentaram o funcionamento da aplicação, porém com uma baixa margem de acerto, com pesquisas relacionadas a alta margem de erro, se associou a utilização de placas impressas. Para solução se monta um banco de

#### 4.2. Aplicação de técnicas de visão computacional e algoritmo de identificação

| PLACA    | LEITURA OTIMIZADA | RESULTADO    |
|----------|-------------------|--------------|
| HJK-7634 | HIK7634           | ERRO         |
| IOP-7354 | IOB7853           | ERRO         |
| JOP-6120 | JOP6120           | ACERTO       |
| CAT-8712 | CAI8712           | ACERTO       |
| BAR-0873 |                   | INCONCLUSIVO |
| HUG-7851 | HUG7851           | ACERTO       |
| HAI-0709 | HAI0709           | ACERTO       |
| UOY-7236 | UOY7236           | ACERTO       |
| PAJ-7811 | BAI7911           | ERRO         |
| TFQ-7612 | EEO7813           | ERRO         |
| HJI-1303 |                   | INCONCLUSIVO |
| UOX-1789 | UOX1789           | ACERTO       |
| ZWE-0316 | ZWE0316           | ACERTO       |
| IUQ-9456 | TUO8458           | ERRO         |
| VCR-3268 | UCB8269           | ERRO         |
| KSE-0987 |                   | INCONCLUSIVO |
| YRT-0734 |                   | INCONCLUSIVO |
| HBS-7252 | HBS7252           | ACERTO       |
| URL-8762 | UBI7252           | ERRO         |
| TYZ-8532 | TYZ8532           | ACERTO       |

Tabela 1 – Relação de placas identificadas.

imagens de veículos estacionados, onde as imagens seguem o padrão imposto. Algumas destas podem ser observadas na Figura 27.



Figura 27 – Exemplo banco de imagens

## 4.2. Aplicação de técnicas de visão computacional e algoritmo de identificação

---

Utilizando o banco de imagens a margem de acerto subiu drasticamente, com exceção de pequenas falhas esporádicas o sistema apresenta funcionalidade quase perfeita, portanto trabalhando com placas reais o sistema apresenta margem de funcionalidade superior a 98%. Podemos observar na Tabela 2 uma relação de 20 placas do banco de imagens e seus respectivos resultados.

| PLACA    | LEITURA<br>OTIMIZADA | RESULTADO |
|----------|----------------------|-----------|
| OQP-3477 | OQP3477              | ACERTO    |
| HAG-9176 | HAG9176              | ACERTO    |
| HDF-3801 | HDF3801              | ACERTO    |
| PVP-6584 | PVP6584              | ACERTO    |
| GWH-0523 | GWH0523              | ACERTO    |
| DJL-7088 | DJL7088              | ACERTO    |
| OQX-6214 | OQX6214              | ACERTO    |
| PYC-4172 | PYC4172              | ACERTO    |
| HKT-1182 | HKT1182              | ACERTO    |
| CKX-1014 | CXX1014              | ERRO      |
| GOT-5224 | GOT5224              | ACERTO    |
| OXE-6629 | OXE6629              | ACERTO    |
| OPJ-5336 | OPJ5336              | ACERTO    |
| HKQ-3020 | HKQ3020              | ACERTO    |
| GYC-7145 | GYC7145              | ACERTO    |
| QNU-9186 | QNU9186              | ACERTO    |
| OQF-1140 | OQF1140              | ACERTO    |
| ORA-7252 | PRA7252              | ACERTO    |

Tabela 2 – Relação de placas identificadas.

O erro apresentado na tabela anterior pode se justificar através da qualidade de conservação da placa obtida junto a semelhança entre as letras K e X neste modelo, como pode ser observado na Figura 28.



Figura 28 – Placa em má conservação.

### 4.3 Banco de Dados

Para uma melhor organização das informações obtidas pela plataforma de obtenção de dados, é necessária a criação de um banco de dados do tipo relacional. O banco de dados relacional possibilita que os usuários utilizem uma grande variedade de abordagens no tratamento das informações.

De acordo com os dados levantados nos requisitos, foi criado um modelo entidade relacionamento (MER). Esse modelo, utilizado na Engenharia de Software e em Banco de Dados, é responsável por representar as entidades do sistema, bem como seus atributos e os relacionamentos entre essas entidades. Através do MER, conseguimos ter uma visão melhor de como os objetos se relacionam dentro do sistema de banco de dados.

A Figura 29 ilustra o modelo entidade relacionamento criado para o desenvolvimento web desse projeto.

| <b>Cliente</b>            |
|---------------------------|
| Id INT                    |
| Nome VARCHAR              |
| <u>Email</u> VARCHAR      |
| Telefone VARCHAR          |
| <u>N_Registro</u> VARCHAR |
| Idade INT                 |
| Placa VARCHAR             |
| <u>Renavam</u> INT        |
| Chassi INT                |
| Classe VARCHAR            |
| Pré-Natal VARCHAR         |
| Período DATE              |

Figura 29 – Modelo entidade relacionamento.

Com o banco de dados devidamente esquematizado e organizado, o próximo passo é a etapa de conexão entre a interface web, o banco de dados e o software de visão computacional. O processo funciona da seguinte forma: Primeiramente o usuário cadastra os seus dados através no site disponibilizado na web, essas informações é recolhida pela central, onde há uma pessoa responsável por fiscalizar as informações fornecidas pelo usuário, como por exemplo, número de registro no cadastro municipal de deficientes ou idosos, como também placa, chassi e renavam do veículo.

Então, com os dados obtidos e devidamente verificados, é feito o cadastro do cliente no banco de dados, com o intuito de gerenciar melhor as informações de cada usuário, assim, a placa deste veículo é cadastrada no algoritmo de autorização de acesso, dando assim a permissão para o usuário usufruir de todas as vagas onde se encontra o dispositivo de monitoramento.

## 4.4 Periféricos

De acordo com as exigências do projeto, e visando a viabilidade do mesmo, todo o desenvolvimento do dispositivo foi feito utilizando periféricos de fácil acesso, com o intuito de enfatizar o projeto mais na funcionalidade do que na prototipagem do mesmo.

Levando em consideração que o projeto está na fase de estudos e aprimoramento, para obter um melhor processamento de imagens foi utilizado um notebook modelo Dell Inspiron 14, com o sistema operacional Linux, suas configurações de hardware podem ser vistas a seguir:

|   |
|---|
| <b>Notebook Dell Inspiron 14<br/>série 3000</b> |
| 4GB de RAM                                      |
| Disco Rígido 1 TB                               |
| Tela 14"  |
| Placa de vídeo Intel HD<br>Graphics 520         |
| Processador Intel Core i5-<br>7200U             |



Figura 30 – Notebook utilizado.

Para fazer a obtenção de imagem, foi utilizado uma câmera de ação filmadora com resolução de vídeo 4k (30 fps) e resolução de imagem de 16 megapixels (4640 x 3480), com o intuito de obter uma melhor captura e facilitar o processamento da visão computacional. Esta câmera possui conexão usb, onde ela é conectada no notebook, fazendo assim, uma interação em tempo real com o software de visão computacional.

Para fazer a simulação do alarme sonoro do dispositivo, é utilizado uma mini caixa de som com 4W RMS. Esse alarme sonoro é acionado caso o usuário não tenha autorização para estacionar na vaga de trânsito especial. Como se trata apenas de uma simulação do que irá ocorrer na prática, esse periférico foi capaz de cumprir todas as exigências do projeto. As suas configurações podem ser vistas a seguir:

No quesito de estrutura eletrônica, optamos por fazer uma montagem mais informativa e simulatória, sem se preocupar com robustez e estética, levando em consideração que se trata do "grosso" do projeto. Essa montagem é feita com o intuito de mostrar que



Figura 31 – Câmera.

| Mini Caixa de Som       |             |
|-------------------------|-------------|
| Fonte de alimentação    | 5V USB      |
| Saída de áudio:         | 3,5 mm(P2)  |
| Potência de saída:      | 2W          |
| Sensibilidade:          | -65dB       |
| Impedância:             | 40hms       |
| Frequência de resposta: | 150Hz~14kHz |



Figura 32 – Mini caixa de som.

realmente o projeto funciona, e ela simula a funcionalidade do dispositivo final que se encontrará nas vias. Para isso, é utilizado 2 leds difuso, na cor verde e vermelha, esses leds tem o intuito de simular o alarme visual do dispositivo, as configurações do mesmo pode ser vista na figura a seguir:

| CARACTERÍSTICAS TÉCNICAS*                                      | Unid. | MÍN. | TIP. | MÁX. |
|--|-------|------|------|------|
| Potência Dissipada   | mW    |      |      | 60   |
| Corrente de Pico<br>(1/10 Ciclo, PW < 1ms)                     | mA    |      |      | 80   |
| Corrente de Operação   | mA    |      | 15   | 20   |
| Coefficiente de Redução acima de 50°C                          | mA/°C |      |      | 0,4  |
| Tensão Reversa   | V     |      |      | 6    |
| Temperatura de Operação  | °C    | -40  |      | +85  |
| Temperatura de Armazenamento                                   | °C    | -40  |      | +100 |
| Temperatura de Soldagem<br>( $\leq$ 5 seg a 3mm da base epost) | °C    | +255 |      | +265 |



Figura 33 – Características Técnicas LEDs.

Para fazer a interação entre o computador e os periféricos, é necessário a utilização

de um Arduino Uno, esse arduino trata-se de uma plataforma de prototipagem eletrônica de hardware livre e de placa única, projetada com um microcontrolador Atmel AVR com suporte de entrada/saída embutido, no qual possui uma linguagem de programação padrão. Este arduino é responsável por fazer o acionamento dos leds e da caixa de som, no qual, ele recebe a informação do software de autorização de acesso, e aciona os componentes responsáveis por passar a informação visual e sonora para o usuário. O modelo escolhido foi o modelo UNO, pelo fato de ser o melhor em relação ao custo benefício do projeto, levando em consideração que não a necessidade de um hardware com um processamento superior em relação a função na qual ele executa no projeto.

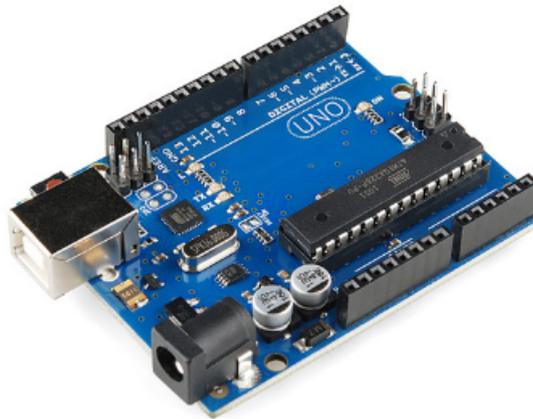


Figura 34 – Arduino UNO.



## Considerações Finais

O presente capítulo contém as considerações a respeito de todas as decisões tomadas para o projeto e desenvolvimento deste trabalho de conclusão. São apresentadas as propostas de continuidade, além do cronograma das atividades.

### 5.1 Conclusões

O trabalho apresenta algumas objeções em sistemas de monitoramento e as vantagens da utilização da visão computacional tanto para operações industriais quanto para solução de diversos problemas cotidianos. Outro conceito abordado é a utilização de sistemas de apoio que trabalham em segundo plano para amplificação das soluções propostas. Em tal contexto se observou a necessidade do desenvolvimento de um sistema capaz além de identificar variáveis, interagir com o ambiente e assim efetivamente atuar na solução do problema observado.

Visto isso, para realização deste projeto foi utilizada uma metodologia composta de duas importantes etapas: Estudo do projeto e Projeto conceitual. Diante de tal, deu-se início ao levantamento do conteúdo literário necessário para o desenvolvimento deste, etapa descrita no capítulo dois do presente trabalho, intitulado Revisão Bibliográfica. Neste tópico, conceitos fundamentais para todo o desenvolvimento do trabalho tais como, conceitos de programação e software; processamento de imagem; reconhecimento de padrões, dentre outros foram abordados para levantamento de conceitos e definições a respeito do projeto proposto.

O desenvolvimento do trabalho se atentou estritamente aos passos abordados no capí-

tulo 3 seguindo os conceitos literários obtidos anteriormente. Estes foram fundamentais para o desenvolvimento das atividades e conseqüentemente geração dos resultados. Portanto, é possível afirmar que tendo em vista todos conceitos impostos no pré-projeto os resultados são satisfatórios perante a eficiência dos métodos utilizados.

O site desenvolvido apresenta um comportamento esperado, onde sua praticidade se mostrou compatível e essencial ao público destinado.

O banco de dados apresenta funcionalidade ideal para trabalho em pequenos níveis, o que se sustenta pelo trabalho se tratar apenas de um processo a pequeno nível de cadastro.

Observando-se o custo computacional imposto pela carga de análises geradas pelo desenvolvimento da visão computacional se concluiu que esse custo pode ser suprido sem limitações pelo periférico computacional *Raspberry Pi 1*.

Mais um fato de interesse, se trata da abordagem de conteúdo de programação para implementação dos algoritmos necessários (Identificação e reconhecimento de caracteres, desenvolvimento website, dentre outros). E o conteúdo de eletrônica para realização do processamento digital das imagens, tratamento de sinais, aplicação de filtro de ruídos e utilização de hardware sinalizador.

Considerando então todo o processo, os resultados do protótipo consistem com o esperado, apresentando baixa margem de erro de identificação, inclusive estas são inferiores as esperadas inicialmente de acordo com a realidade do hardware utilizado para desenvolvimento do mesmo. Portanto, com os resultados obtidos pode-se afirmar que o sistema elaborado age de maneira satisfatório correspondendo minimamente as expectativas do projeto.

## 5.2 Propostas de continuidade

Considerando as etapas cumpridas até o momento e o alcance dos objetivos propostos inicialmente para finalização do projeto. Para este trabalho, há propostas de atividades futuras.

A primeira recomendação é o desenvolvimento de um algoritmo capaz de conferência dos dados fornecidos pelo usuário para comparação com o banco de dados municipal, estadual ou federal que assegure a segurança necessária para que o usuário realmente utilize dados verídicos e assim aumente a confiabilidade do sistema de cadastro, vale

ressaltar que o acesso aos bancos de dados governamentais é de certa forma restrito para um acesso massivo, ou seja, atualmente esse acesso deve ser feito a um usuário por vez e manualmente.

Em segundo plano temos a realização de um melhor sensoriamento ao protótipo de modo a evitar a atividade incessante de análise da câmera a imagem obtida, portanto com um sensor que tenha menor custo computacional e energético, destinado a verificar a presença de veículos o trabalho executado seria mais econômico ao sistema.



## Códigos

### A.1 Instalação biblioteca OpenCv

No terminal do linux o seguinte comando deve ser aplicado

```
sudo apt-get install git -y
```

Em seguida os scripts devem ser baixados

```
git clone https://github.com/bishoppython/Install_OpenCV_Script
```

A partir disto o script de instalação ode ser executado. Para isso, você precisa entrar no diretório criado ao fazer o “git clone” e depois entrar no diretório da distro que você utiliza. Agora basta executar o script `opencv_latest.sh`, utilizando super usuário e aguardar ele finalizar a instalação, neste caso os comandos ficaram da seguinte forma:

```
bishop-elliott # cd /home/bishop/Install-OpenCV/Ubuntu/bishop-elliott/Ubuntu  
./opencv_latest.sh
```

### A.2 Leitura e identificação

```
1 #RECONHECIMENTO AUTONOMO DE PLACAS DE VEICULOS E APLICACAO DE MEDIDAS  
   CORRETIVAS  
2 #AUTORES: RONALDO R. TAVARES JUNIOR & VITOR FERREIRA DECIMO  
3 #ORIENTADOR: PROF. DR. JOAO CARLOS
```

```
4
5 import pytesseract as ocr
6 import numpy as np
7 import cv2
8 import re
9 import pygame
10 import time
11 from PIL import Image
12 from cv2 import *
13 from openalpr import Alpr
14
15
16 #INICIA PROGRAMA DE RECONHECIMENTO
17
18 print ("\n\n***** RECONHECIMENTO DE PLACAS ***** ")
19 # inicializa a camera
20 cam = VideoCapture(0) # localiza camera no index determonado
21 parado = 'x';
22 while True :
23     cam = VideoCapture(0)
24     e='';
25     s, img = cam.read()
26     imwrite("placa.jpg",img)
27     imagem = cv2.imread("placa.jpg",0)
28     cam.release ()
29     cv2.imwrite("log/escaladecinza.jpg",imagem)
30     ret, binaria = cv2.threshold(imagem, 240, 255,
31     cv2.THRESH_BINARY | cv2.THRESH_OTSU )
32     cv2.imwrite("log/binaria.jpg",binaria)
33     imgb = cv2.imread("log/binaria.jpg")
34     corte = imgb[200:400, 200:1010]
35     cv2.imwrite("log/corte.png",corte)
36
37     # DILACAO E EROSAO
```

```
38
39     kernel = np.ones((2,2),np.uint8)
40     dilation = cv2.dilate(corte,kernel,iterations = 0)
41     erosion = cv2.erode(dilation,kernel,iterations = 3)
42     cv2.imwrite("log/erosao.jpg",erosion)
43
44     Redimensionamento da imagem
45     im = Image.open("log/corte.png")
46     im.thumbnail((150,150))
47     im.save("log/miniatura.png")
48
49     #LEITURA OCR - openALPR
50     alpr = Alpr("br", "/usr/share/openalpr/config/openalpr.
        defaults.conf", "/usr/share/openalpr/runtime_data")
        localiza diretorio e seta linguagem.
51     if not alpr.is_loaded():
52     print("Error loading OpenALPR")
53     sys.exit(1)
54     alpr.set_top_n(1)
55     alpr.set_default_region("md")
56     results = alpr.recognize_file("corte.jpg")
57     i = 0
58     for plate in results['results']:
59     i += 1
60     for candidate in plate['candidates']:
61     prefix = "-"
62     if candidate['matches_template']:
63     prefix = "*"
64     e = candidate['plate']
65     alpr.unload() #zera memoria
66     print ("PLACA RECOLHIDA ")
67     print e
68     if e == '':
69         if parado == e:
```

```
70         time.sleep(5)
71     else:
72         print ("                VAGA VAZIA")
73         time.sleep(5)
74         parado = e;
75     else:
76         def pesquisar_registro( arq, placa ):
77             with open( arq, 'r' ) as a:
78                 for linha in a:
79                     if linha.strip() == placa:
80                         return True
81                     return False
82         v = pesquisar_registro( 'BANCODEDADOS/placas.txt', e
83             )
84     if v:
85         if parado == e:
86             time.sleep(5)
87         else:
88             print "                PLACA LIBERADA "
89             pygame.init()
90             pygame.mixer.music.load("alertas/AVIS00.mp3")
91             pygame.mixer.music.play()
92             time.sleep(5)
93             pygame.mixer.music.stop()
94             parado = e;
95     else:
96         if parado == e:
97             if beta ==0:
98                 beta = 1;
99                 pygame.init()
100                 pygame.mixer.music.load("
                    alertas/AVIS02.mp3")
                    pygame.mixer.music.play()
```

```
101         time.sleep(8)
102         pygame.mixer.music.stop()
103     if beta == 1:
104         arquivo = open('BANCODEDADOS/autuacao
105             .txt', 'r')
106         conteudo = arquivo.readlines()
107         conteudo.append('\n')
108         conteudo.append(parado)
109         arquivo = open('BANCODEDADOS/autuacao
110             .txt', 'w')
111         arquivo.writelines(conteudo)
112         arquivo.close()
113         print("Veiculo Autuado")
114         pygame.init()
115
116         pygame.mixer.music.load("alertas/
117             AVIS03.mp3")
118         pygame.mixer.music.play()
119         time.sleep(8)
120         pygame.mixer.music.stop()
121         beta=2;
122
123     if beta == 2:
124         time.sleep(8);
125     else:
126         beta = 0
127         parado = e;
128
129         print "          VEICULO NAO CADASTRADO
130             "
131         pygame.init()
132
133         pygame.mixer.music.load("alertas/
134             AVIS01.mp3")
```

```
127         pygame.mixer.music.play()
128         time.sleep(8)
129         pygame.mixer.music.stop()
130     alpr.unload()
```

## A.3 Cadastro Banco de dados

```
1     print ("CADASTRO BANCO DE DADOS ")
2     s = raw_input("ENTRE COM A PLACA :")
3     arquivo = open('placas.txt', 'r')
4     conteudo = arquivo.readlines()
5     conteudo.append(' ')
6     conteudo.append(s)
7     arquivo = open('placas.txt', 'w')
8     arquivo.writelines(conteudo)
9     arquivo.close()
10    print (" ARQUIVO CADASTRADO COM SUCESSO")
```

# Bibliografia

CIDADES, Ministerio das. **Cartilha Urbana**. [S.l.: s.n.], 2018.

DAMORE, Luiz Angelo. **Robustez na segmentacao de placas veiculares em condicoes complexas de aquisicao**. [S.l.: s.n.], 2015.

DENATRAN. **Departamento Nacional de Transito**. [S.l.: s.n.], 2018.

DEVMEDIA. **Criando aplicativos android com banco de dados SQLite**. 2018.  
Disponível em: <[www.devmedia.com.br](http://www.devmedia.com.br)>.

DOUXCHAMPS, Chihara. High accuracy and robust localization of large control markers for geometric camera calibration. **IEEE transactions on pattern analysis and machine intelligence**, 2009.

ERPEN, Luis Renato Cruz. **Reconhecimento de Padroes em imagem por descritores de forma**. 2004. Mestrado – Universidade Federal do Rio Grande do Sul.

GOLDSMITH, Andrea. **Wireless communications**. [S.l.: s.n.], 2005.

GONCALVES, Jeferson. **Google investe vinte milhoes em tecnologia para deficientes**. 2016. Disponível em: <[www.olhardigital.com.br](http://www.olhardigital.com.br)>.

GONZALES, Rafael. **Digital image processing**. 2000. Mestrado – Universidade of Tennessee.

HUANG, Thomas. **Computer Vision Evolution and promise**. [S.l.: s.n.], 1996.

IDC. **Crescimento da venda de smartphones**. 2017. Disponível em:  
<[www.idc.com](http://www.idc.com)>.

\_\_\_\_\_. **Popularizacao dos smartphones**. 2013. Disponível em: <[www.idc.com](http://www.idc.com)>.

- 
- LEE, Jin-Shyan; SU, Yu-Wei; SHEN, Chung-Chou. A Comparative Study of Wireless Protocols: Bluetooth, UWB, ZigBee, and Wi-Fi. **IEEE Industrial Electronics Society (IECON)**, 2007.
- MORAES, Douglas Alves de. Prototipo para coleta de informacoes em saude utilizando dispositivos moveis. **Departamento de informatica da saude**, 2016.
- OPENCV. **Open source computer vision library**. 2018. Disponível em: <[www.opencv.org](http://www.opencv.org)>.
- PARK, Brasil. **Parametros de vagas de estacionamento**. 2018. Disponível em: <[www.brasilpark.com.br](http://www.brasilpark.com.br)>.
- PERES, Marcelo Pereira. **Guia CFTV**. 2006. Disponível em: <[www.guiadocftv.com.br](http://www.guiadocftv.com.br)>.
- PYTHON. **Biblioteca Python**. 2018. Disponível em: <[www.pythonsciencelibrary.com](http://www.pythonsciencelibrary.com)>.
- RIBEIRO, Daniel. **Estudo de interface humano maquina em dispositivos moveis**. 2007. Mestrado – Universidade Federal de Santa Catarina.
- SHIMOSKAI, Ricardo. **Vagas de estacionamento reservadas para pessoas com deficiencia**. 2017. Disponível em: <[www.turismoadaptado.com.br](http://www.turismoadaptado.com.br)>.
- SILVEIRA, Sergio Amadeu da. **Inclusao digital e software livre**. [S.l.: s.n.], 2004.
- TAKAI, Oracio. **Introducao ao banco de dados**. 2005. Doutorado – Universidade do Estado de Sao Paulo.
- TSAI, Robert. A versatile camera calibration technique for high accuracy 3d machine vision metrology using. **IEE journal of robotic and Automation**, 1991.
- UBUNTO. **Sistema operacional Ubuntu**. 2018. Disponível em: <[www.ubuntu.com/about](http://www.ubuntu.com/about)>.
- VOLTOLINI, Ramon. **Conheca o primeiro smartphone da historia**. 2017. Disponível em: <[www.tecmundo.com.br](http://www.tecmundo.com.br)>.
- WANG, Lor. Camera calibration by vanishing lines for 3d computer version. **IEEE transactions on Pattern Analysis and Machine Intelligence**, 1991.
- YAMAWAKI, Sergio. **Engenheiro cria equipamento para monitorar vagas exclusivas**. 2018. Disponível em: <[www.g1.globo.com/pr/parana](http://www.g1.globo.com/pr/parana)>.