

**CENTRO ESTADUAL DE EDUCAÇÃO TECNOLÓGICA
PAULA SOUZA**

**FACULDADE DE TECNOLOGIA DE ITU
“DOM AMAURY CASTANHO”**

**BIANCA ALVES BARBOSA
GUILHERME DE MATTOS COSSARI**

**SISTEMA DE CONTROLE DE ENTRADA E SAÍDA DE
VEÍCULOS ATRAVÉS DA LEITURA DE PLACAS.**

**BIANCA ALVES BARBOSA
GUILHERME DE MATTOS COSSARI**

**SISTEMA DE CONTROLE DE ENTRADA E SAÍDA DE
VEÍCULOS ATRAVÉS DA LEITURA DE PLACAS.**

Trabalho de Graduação apresentado à Banca Examinadora da Faculdade de Tecnologia de Itu "Dom Amaury Castanho", como exigência parcial para conclusão do Curso Superior de Tecnologia em Análise e Desenvolvimento de Sistemas, sob a orientação do Prof. Dilermando Piva Junior.

Itu
Setembro/2020

**BIANCA ALVES BARBOSA
GUILHERME DE MATTOS COSSARI**

**SISTEMA DE CONTROLE DE ENTRADA E SAÍDA DE
VEÍCULOS ATRAVÉS DA LEITURA DE PLACAS.**

Trabalho de Graduação apresentada à Banca Examinadora da Faculdade de Tecnologia de Itu "Dom Amaury Castanho", como exigência parcial para conclusão do Curso Superior de Tecnologia em Análise e Desenvolvimento de Sistemas, sob a orientação do Prof. Dilermando Piva Junior.

Aprovado em

Dilermando Piva Jr/Doutor /Fatec-Itu

Nome do convidado/ titulação/IES

Nome do convidado/ titulação/IES

DEDICATÓRIA

Dedicamos esse trabalho as nossas famílias, que esteve presente durante todo esse processo de nossas vidas. Aos professores, que nos orientaram e nos concederam todo o conhecimento necessário para a essa etapa. Ao nosso orientador Prof. Dr. Dilermando Piva Jr., que nos acolheu e acreditou no nosso potencial, orientando sobre todo o presente trabalho. Aos amigos e companheiro em especial Lucas de Moraes Souza, que estiveram conosco durante toda essa etapa.

AGRADECIMENTOS

Queremos agradecer, primeiramente, aos nossos pais, por todo o apoio e educação que foi nos dados, para conseguirmos concluir mais esse passo em nossas vidas. Ao nosso orientador Prof. Dr. Dilermando Piva Jr., por todo o apoio, ajuda e disponibilidade para nos auxiliar neste projeto. A todos os amigos e companheiro que esteve nos incentivando a nunca desistir. Aos professores Daniella Arruda Franceschinelli Borges (in memoriam) e Renato Luiz Cardoso por todo auxílio durante o desenvolvimento deste projeto.

RESUMO

Em diversas instituições de ensino incluindo a Faculdade de Tecnologia de Itu – Fatec Itu, que foi a base e referência para este presente projeto, o método utilizado para segurança e controle de entrada e saída de veículos é realizado de forma manual ou até mesmo mecânica, utilizando sistema de cancela sem o cadastro do veículo ou apenas utilizando anotações. No caso específico da Fatec Itu, o controle de acesso a instituição é feito por meio de adesivos que se encontram no para-brisa frontal do veículo para identificação do veículo e de seu ocupante. Nenhum registro de acesso é realizado. O presente projeto propõe o desenvolvimento de um sistema computacional que efetua a leitura automatizada das placas dos veículos na entrada e na saída da instituição. Tomando como base o número da placa, o sistema irá registrar as informações, incluindo o horário de entrada e saída do mesmo. Para eliminar o cadastro manual dos registros na instituição, será desenvolvido junto ao sistema de reconhecimento de placas, um sistema web de cadastro para auxiliar na identificação do proprietário do veículo e a liberação do acesso.

Palavras-chave: Segurança. Reconhecimento de placas. Inteligência Artificial. Instituição de ensino. Controle de veículos.

ABSTRACT

In several educational institutions including the itu college of technology - fatec itu, which was the base and reference for this project, the method used for safety and control of entry and exit of vehicles is performed manually or even mechanically, using a parking barrier system without registration of the vehicle or just using notes. In the specific case of fatec itu, the access control to the institution is done by means of stickers that are on the frontwindscreen of the vehicle to identify the vehicle and its occupant. No access log is performed. This project proposes the development of a computer system that performs automated reading of vehicle license plates at the entrance and exit of the institution. Based on the number of the card, the system will record the information, including the time of entry and exit. To eliminate the manual registration of records in the institution, will be developed with the plate recognition system, a web registration system to assist in identifying the owner of the vehicle and release access.

Key-words: Security. Plate recognition. Artificial Intelligence. Educational institution. Vehicles control.

LISTA DE ILUSTRAÇÕES

Figura 1 - Subconjuntos da IA	19
Figura 2 - Tipos, algoritmos e métodos de ML.	21
Figura 3 - Subáreas da IA	22
Figura 4 - Resultado dos filtros utilizados na etapa de Pré-processamento.....	24
Figura 5 - método de detecção de descontinuidades	27
Figura 6 – Detecção de contornos por derivadas	27
Figura 7 - Resultado do reconhecimento de caracteres com OpenCV e Tesseract	28
Figura 8 - Arquitetura da Tesseract	29
Figura 9 - Linha de pinos GPIO Raspberry	31
Figura 10 - Conectores da Raspberry	32
Figura 11 - Imagem da Câmera Raspberry Pi v2	32
Figura 12 – Componentes do Arduino Uno.....	34
Figura 13 – Especificação dos pinos da placa Arduino Uno	34
Figura 14 – Representações e logos das tecnologias utilizadas.....	41
Figura 15 - Diagrama de arquitetura do sistema de reconhecimento e sistema web	42
Figura 16 – Esquema de ligação do sistema físico	44
Figura 17 - Diagrama de casos de uso de alto nível	46
Figura 18 - Diagrama de classes do sistema	47
Figura 19 - Diagrama entidade-relacionamento do sistema	48
Figura 20 - Tela principal	49
Figura 21 – Tela de login do aluno	50
Figura 22 – Tela de cadastro do aluno	51
Figura 23 – Tela de recuperação de senha do aluno	51
Figura 24 – Tela de reset de senha do aluno	52
Figura 25 – Tela de cadastro do gestor	52
Figura 26 – Dashboard do aluno	53
Figura 27 – Tela de login do administrador.....	53
Figura 28 – Listagem dos veículos no Dashboard Administrativo.....	54
Figura 29 - Listagem dos acessos no Dashboard Administrativo.....	55
Figura 30 – Informações Geral do respectivo acesso	55
Figura 31 – Resultado da etapa de pré-processamento do projeto	56

Figura 32 – Contornos encontrados na etapa de identificação da placa	57
Figura 33 – Placa identificada na etapa de identificação da placa	57

LISTA DE QUADROS

Quadro 1 - Caracterização dos trabalhos relacionados.....	36
Quadro 2 – Resultados obtidos a partir do Sistema Computacional	57

LISTA DE TABELAS

Tabela 1 - Alguns Modelos atuais do Raspberry Pi	30
Tabela 2 - Especificação de hardware	33
Tabela 3 - Requisitos funcionais do sistema.....	44
Tabela 4 - Requisitos não funcionais do sistema.....	45
Tabela 5 - Diagrama de casos de uso baixo nível do requisito efetuar login	65
Tabela 6 - Diagrama de casos de uso baixo nível do requisito realizar cadastro	65
Tabela 7 - Diagrama de casos de uso baixo nível do requisito recuperar senha.....	66
Tabela 8 - Diagrama de casos de uso baixo nível do requisito cadastrar veículo.....	66
Tabela 9 - Diagrama de casos de uso baixo nível do requisito listar controle.....	67
Tabela 10 - Diagrama de casos de uso baixo nível do requisito captar vídeo	68
Tabela 11 - Diagrama de casos de uso baixo nível do requisito reconhecer placa.....	68
Tabela 12 - Diagrama de casos de uso baixo nível do requisito consultar placa.....	69
Tabela 13 - Diagrama de casos de uso baixo nível do requisito registrar controle	69
Tabela 14 - Diagrama de casos de uso baixo nível do requisito acionar cancela	70
Tabela 15 - Dicionário de dados do diagrama entidade relacionamento	70

LISTA DE ABREVIATURAS E SIGLAS

IA	Inteligência Artificial
ML	<i>Machine Learning</i>
DL	<i>Deep Learning</i>
GPIO	<i>General Purpose Input/Output</i>
MP	<i>Mega Pixels</i>
DC	<i>Direct current</i>
PDI	Processamento Digital de Imagens
ALPR	<i>Automated License Plate Recognition</i>
VC	Visão Computacional
CL	<i>Classical Learning</i>
HTTP	<i>HyperText Transfer Protocol</i>
HTML	<i>HyperText Markup Language</i>
JWT	Json Web Token
PWM	Modulação por Largura de Pulso
RGB	<i>Red, Green, Blue</i>

SUMÁRIO

INTRODUÇÃO	14
CAPÍTULO 1 REFERENCIAL TEÓRICO	17
1.1 INTELIGÊNCIA ARTIFICIAL	18
1.2 VISÃO COMPUTACIONAL	22
1.3 RECONHECIMENTO DE PADRÕES	23
1.3.1 <i>Pré-processamento</i>	23
1.3.2 <i>Identificação da placa</i>	24
1.3.2.1 <i>Segmentação por contornos</i>	26
1.3.3 <i>Extração dos caracteres</i>	28
1.3.4 <i>Reconhecimento dos caracteres</i>	28
1.4 BIBLIOTECA OPENCV	28
1.5 BIBLIOTECA PYTESSERACT	29
1.6 RASPBERRY	30
1.6.1 <i>Raspicam</i>	32
1.7 ARDUINO	33
1.8 TRABALHOS RELACIONADOS	35
CAPÍTULO 2 TECNOLOGIAS UTILIZADAS	37
2.1 PYTHON	37
2.2 PROTOCOLO HTTP	37
2.3 JAVASCRIPT	38
2.4 HTML5	38
2.5 REACT JS	38
2.6 NODE JS	39
2.7 EXPRESS	39
2.8 BYCRYPT	39
2.9 MYSQL	40
2.10 JSON WEB TOKEN (JWT)	40
2.11 MATERIAL-TABLE	40
2.12 CLOUDINARY	40
CAPÍTULO 3 MODELAGEM DE DADOS DA APLICAÇÃO	42
3.1 DIAGRAMA DE ARQUITETURA DE ALTO NÍVEL	42
3.2 ESQUEMA DE LIGAÇÃO DO SISTEMA FÍSICO	43
3.3 REQUISITOS FUNCIONAIS	44
3.4 REQUISITOS NÃO FUNCIONAIS	45
3.5 DIAGRAMA DE CASOS DE USO ALTO NÍVEL	45
3.6 DIAGRAMA DE CASOS DE USO BAIXO NÍVEL	46

3.7 DIAGRAMA DE CLASSES.....	46
3.8 DIAGRAMA ENTIDADE-RELACIONAMENTO	48
3.9 DICIONÁRIO DE DADOS	48
CAPÍTULO 4 MODELAGEM DO PROTÓTIPO.....	49
4.1 MODELAGEM DAS TELAS	49
4.2 RESULTADOS OBTIDOS	56
CONSIDERAÇÕES FINAIS.....	60
REFERÊNCIAS BIBLIOGRÁFICAS	61
APÊNDICES.....	65
APÊNDICE A – SEQUÊNCIA DE DIAGRAMAS DE CASOS DE USOS BAIXO NÍVEL	65
APÊNDICE B – DICIONÁRIO DE DADOS	70

INTRODUÇÃO

Analisando a segurança atual da Faculdade de Tecnologia de Itu – Dom Amaury Castanho e das demais faculdades ou instituições de ensino, é notório que em algumas delas não existe controle e registro de dados algum, sobre a entrada e saída de alunos e veículos mesmo que seja somente uma breve identificação.

Com a falta de segurança em muitas instituições de ensino, na maioria públicas, é possível se deparar com acontecimentos como, furto ou roubo aos bens públicos voltados ao ensino, furto ou roubo aos bens materiais dos alunos e gestores da organização, furto ou roubo das informações confidenciais ou não da instituição, estragos no patrimônio público escolar, tragédias extremas envolvendo ataques aos alunos e gestores etc. Um simples controle de entrada pode evitar que pessoas de fora da instituição acabem não sendo identificadas e fazendo o que lhe convém, podendo trazer grandes prejuízos às instituições de ensino.

Portanto esse projeto visa auxiliar as instituições, e em específico a Fatec Itu, a solucionar parte da segurança, controlando a entrada e saída de veículos dos alunos, funcionários e visitantes da instituição de forma automatizada.

Considerando que o controle de entrada e saída de veículos é uma segurança física, surge o seguinte problema de pesquisa: ***Como as Tecnologias de Informação, em especial, computadores de baixo custo e a área da computação voltada ao reconhecimento de padrões/imagens, pode auxiliar no controle de entrada e saída de veículos nas instituições de ensino?***

Para responder essa questão, foram pensadas soluções que coletassem dados do veículo (e, em um segundo momento, da pessoa que está conduzindo o veículo) de forma automatizada e associasse esses dados à pessoa responsável pelo veículo, registrando todos os dados e armazenando essas informações em bases de dados, possibilitando a consulta, pela instituição, *a posteriori*. Tais informações ficariam registradas em servidores externos (nuvem ou *cloud* em inglês), possibilitando assim, segurança na guarda e manutenção dessas informações.

Diante disso, o objetivo geral do presente trabalho de graduação é o desenvolvimento de um sistema computacional e físico que controle a entrada e saída de veículos por meio do reconhecimento e identificação das placas dos automóveis. O protótipo do sistema físico deve conter, a princípio, câmera e sinalizador de acesso (Leds com cores que sinalizem a permissão do acesso, fazendo as vezes de uma cancela). O sistema computacional, realizará a obtenção,

identificação e registro das placas dos automóveis, e assim, autorizar ou não a sua entrada na instituição. O registro será feito em banco de dados localizado em servidor (nuvem), e poderá ser acessado remotamente, para verificação *a posteriori*. Também contará com um sistema web para alunos e gestores cadastrarem seus veículos e os administradores consultarem todas as informações de controle dos veículos que tiveram acesso a instituição.

Para atingir esse objetivo geral, foram cumpridos os seguintes objetivos específicos ou parciais:

- Realização de pesquisa conceitual sobre os principais temas a serem trabalhados ao longo deste projeto, destacando-se: inteligência artificial, reconhecimento de padrões, desenvolvimento em Raspberry etc.
- Desenvolvimento de um sistema computacional que comporta as seguintes funcionalidades:
 - Leitura de placa do veículo que estiver no alcance da câmera, identificando qual estado o veículo se encontra, entrada ou saída.
 - Busca os dados do aluno ou gestor referente ao cadastro de veículos já realizado no sistema de cadastro *web* que também foi implementado neste presente trabalho.
 - Identifica se há cadastro do veículo para o registrar o acesso.
 - Identifica veículos ainda não cadastrados pela instituição, para realizar a identificação na portaria para a liberação da entrada.
 - Registra dia e horários de entrada e saída do veículo na instituição.
- Desenvolvimento de um protótipo físico com as seguintes funcionalidades:
 - Realiza a sinalização com leds, sobre a permissão de acesso do veículo (fazendo as vezes de uma cancela).
- Promove, pela automatização proposta, agilidade na entrada e saída dos veículos, eliminando tempo de procura manual dos dados para liberação do veículo.

A relevância do tema, está em expor a segurança em instituições, com intuito de mostrar os riscos que podem trazer a ausência de controles adequados e incentivar os alunos a desenvolverem seus próprios projetos em prol da instituição, por meio de artefatos tecnológicos, melhorando assim o ambiente educacional. Para a Faculdade de Tecnologia Itu o tema é relevante pois o presente projeto irá auxiliar no controle dos veículos que circulam pela instituição e facilitar, se e quando implementada pela instituição, o trabalho de registro

das entradas e saídas dos automóveis, trazendo assim maior segurança aos alunos e gestores e, consequentemente, melhoria na qualidade da gestão institucional.

Para obter os dados necessários para o estudo foram realizadas pesquisas bibliográficas para detalhar e exemplificar os principais conceitos e temas técnicos (computacionais) necessários para o desenvolvimento do presente projeto.

O trabalho em questão segue a seguinte estrutura: No capítulo 1, é realizado o referencial teórico, onde são estabelecidas definições dos termos que se fazem importantes para o entendimento do trabalho. No capítulo 2, é abordada a metodologia de desenvolvimento e com enfoque nas tecnologias e metodologias adotadas para o planejamento e desenvolvimento do Sistema de controle de veículos. No capítulo 3, o sistema em questão é modelado, e assim, detalhado os requisitos funcionais e não funcionais do sistema, os casos de uso de alto e baixo nível, o diagrama de entidade e relacionamento e o protótipo do sistema. Por fim, no capítulo 4, são tecidas algumas considerações finais sobre o presente projeto.

CAPÍTULO 1 REFERENCIAL TEÓRICO

Segundo Dias (2005) um ser humano devidamente treinado necessitaria de um tempo aproximado de 5 segundos para conseguir identificar a placa de um automóvel, para somente assim conseguir inseri-la no sistema, além disso pode haver erros no processo de inserção por cansaço, erro na digitação ou distração.

Para realizar esse processo de forma automatizada e diminuir o erro humano, seria necessário que um artefato tecnológico fosse capaz de “enxergar” e compreender o mundo da forma que nós seres humanos compreendemos, pois são vários os processos que nós, como humanos, fazemos para extrair qualquer informação que enxergamos. É claro que isso é feito de forma tão instantânea e automática, que não é perceptível.

A tecnologia capaz de aplicar estratégias para modelar e replicar a visão humana é a área de computação conhecida como visão computacional, sendo ela uma subárea da inteligência artificial (IA), a inteligência capaz de solucionar problemas e automatizar as atividades humanas.

Segundo Russell e Norvig (1995) a Inteligência Artificial tem por finalidade construir artefatos computacionais, como programas e sistemas que são controlados por um computador, e que realizam tarefas que necessitam de inteligência e racionalidade características do humano.

Porém ainda precisamos dar visão a máquina e fazê-la reconhecer os padrões das placas veiculares. Segundo Marr (1982 apud GONÇALVES, 2019) “[...] visão é o método ou processo de descobrir por meio de imagens o que está presente no mundo e onde se está localizado, sendo assim, uma tarefa de processamento de informação por meio de imagens.”. Nesse contexto que é inserida a Visão Computacional.

Entretanto, é necessário cuidado para não confundir visão Computacional com Reconhecimento de Imagens, Reconhecer a imagem é um processo dentro da visão computacional, pois só ela não é suficiente para dar capacidade de percepção e respostas a uma máquina, tão semelhante à de um humano. (INSTITUTO ATLÂNTICO, 2018)

Nesse capítulo abordaremos com mais profundidade esses conceitos e os demais processos de visão computacional, reconhecimento de padrões e aprendizagem de máquina, identificando e detalhando bibliotecas públicas que implementam parte dessas funcionalidades e que serão utilizadas ao longo deste projeto.

Abordaremos também os *hardwares* que serão utilizados para a criação do sistema de controle, sendo eles a Raspberry responsável pelo processamento de todo nosso algoritmo de reconhecimento de placas, toda a entrada de dado dos veículos será responsável pela Raspicam, câmera de própria integração com a Raspberry, e para o controle de acesso do veículo, usaremos um Arduino para realizar toda a lógica e sinalização.

1.1 Inteligência Artificial

Segundo Nakabayashi (2009) a Inteligência Artificial (IA), pode-se ser interpretada como a inteligência produzida pelo homem, para prover as máquinas de alguma habilidade que simule a própria inteligência do homem.

Uma outra definição de IA, pode ser visualizada sob a ótica de Durkin, onde define IA como a “[...] ciência de fazer máquinas fazerem coisas que requereriam inteligência, caso fossem feitas pelo homem.” (DURKIN, 1994 apud NAKABAYASHI, 2009).

Acrescentando, temos a definição de Elias (2020) onde a IA é:

[...] a possibilidade das máquinas (computadores, robôs e demais dispositivos e sistemas com a utilização de eletrônica, informática, telemática e avançadas tecnologias) executarem tarefas que são características da inteligência humana, tais como planejamento, compreensão de linguagens, reconhecimento de objetos e sons, aprendizado, raciocínio, solução de problemas etc. Em outras palavras, é a teoria e desenvolvimento de sistemas de computadores capazes de executar tarefas normalmente exigindo inteligência humana, como a percepção visual, reconhecimento de voz, tomada de decisão e tradução entre idiomas, por exemplo.

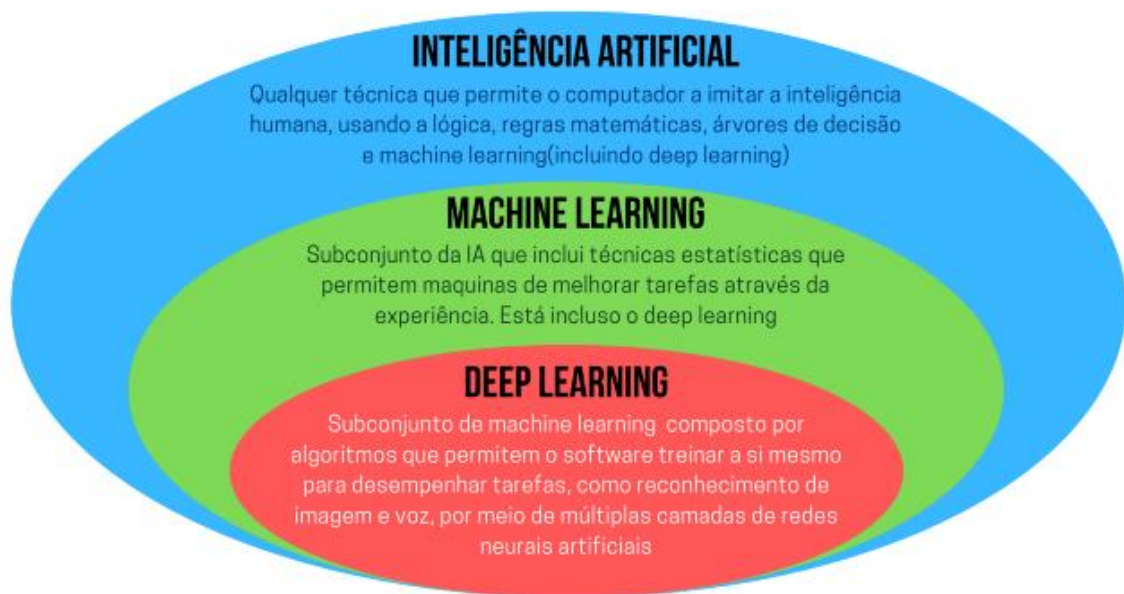
Contudo não se pode confirmar com a certeza de que uma máquina irá pensar exatamente como o ser humano. (GANASCIA, 1993 apud VALLE FILHO, 2003).

Uma IA deve ser capaz de realizar 3(três) coisas: Armazenar conhecimento, aplicar esse conhecimento que lhe foi armazenado para solucionar problemas e adquirir cada vez mais conhecimentos de acordos com suas experiências, além de possuir 3(três) componentes fundamentais, que são a representação, o raciocínio e a aprendizagem (SAGE, 1990, apud HAYKIN, 2001)

Para a aprendizagem existe uma área de estudo mais recente, que compõe a IA, conhecida como *Machine Learning* (ML, em português, *Aprendizagem de Máquina*), que é a capacidade de uma máquina em se manter melhorando o próprio desempenho sem que alguém necessite explicar como realizar suas próprias tarefas que lhe foram atribuídas. (BISHOP, 2006)

Existe uma subárea específica em ML chamada de *Deep Learning* (DL, em português, *Aprendizagem Profunda*), que segundo a IBM é uma aprendizagem que incorpora redes neurais com camadas contínuas, fazendo com que aprenda de maneira iterativa com os dados. As Redes Neurais e a DL são usadas com grande frequência em sistemas de reconhecimento de imagem e visão computacional. (HURWITZ; KIRSCH, 2018). A Figura 1 ilustra essa relação entre IA, ML e DL.

Figura 1 - Subconjuntos da IA



Fonte: <https://medium.com/>

Existem diversas abordagens a ML. Dentre essas abordagens, uma delas pertencem a categoria de Aprendizagens Clássicas (em inglês, *Classical Learning* (CL)). Segundo Lorena (2017) e Pereira de Figueiredo (2020) a definição de duas aprendizagens presentes na CL são:

- **Aprendizado supervisionado:** existe a figura de um ‘professor’ ou especialista orientando o processo, ou seja, existe exemplos rotulados na forma de entrada e saída desejada. Algumas de suas principais abordagens ou algoritmos são:
 - **Regressão:** Necessário quando se precisa prever um valor numérico específico, detectar peso, altura ou metragem de um determinado objeto. Ou seja, quando há números infinitos de valores.

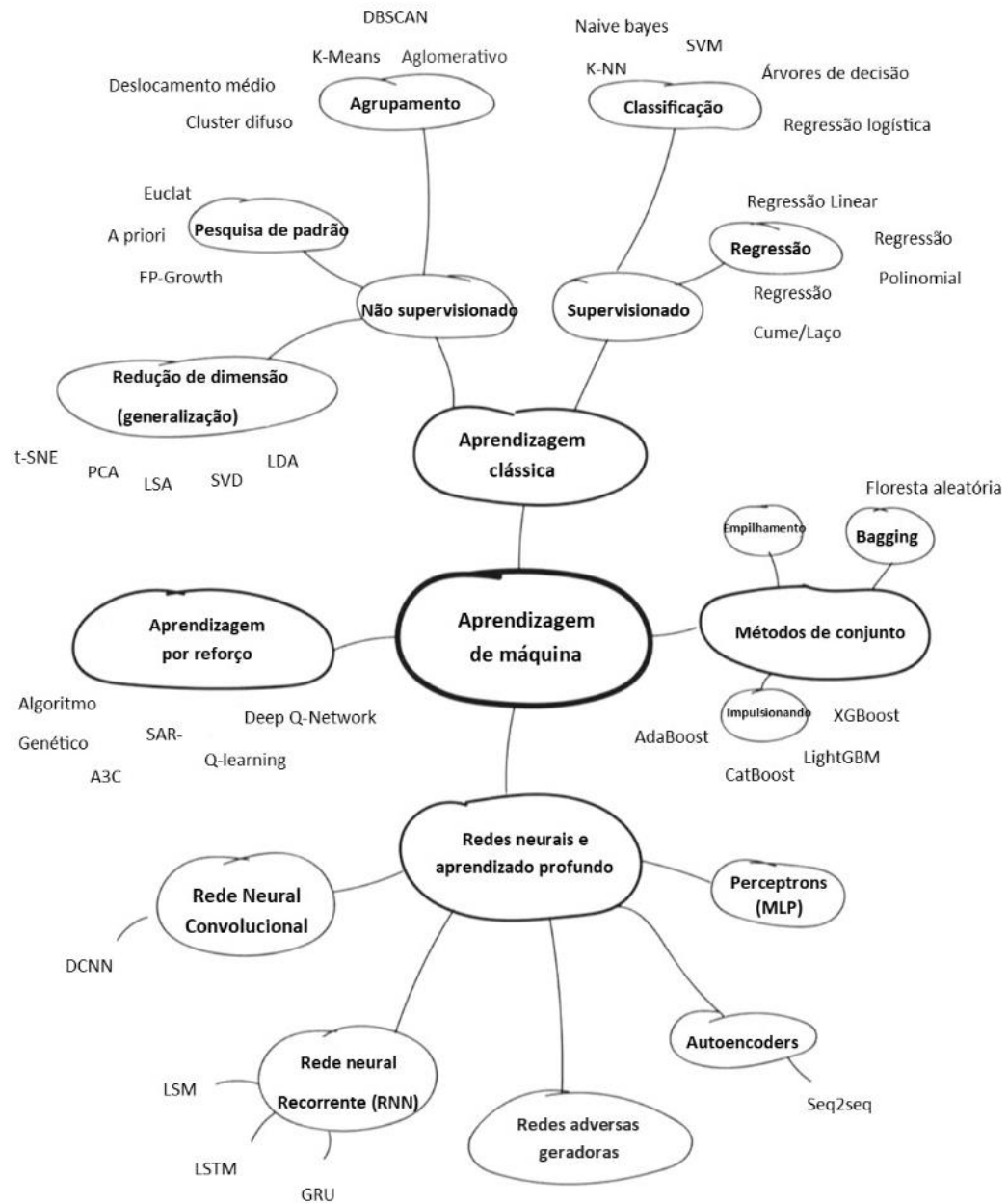
- **Classificação:** Necessário quando se precisa encontrar uma classe dentro de possibilidades limitadas. Ou seja, quando há conjunto finito de valores.
- **Aprendizado não-supervisionada:** não existe a figura de ‘professor’ ou especialista conduzindo o processo de aprendizagem. Não existindo assim exemplos rotulados. Ou seja, as máquinas não têm informações do que aprender. Os problemas que ela trata são a dimensionalidade, detecção de anomalias e regras de associação.

Outro tipo de aprendizagem existente é o:

- **Aprendizado por reforço:** Essa aprendizagem não se tem dados de treinamento, o algoritmo deve aprender através de interações como a tentativa e erro, aprendendo por si só qual a melhor estratégia para obter maior recompensa.

Existem diversos algoritmos de aprendizagem para cada um dos tipos de aprendizagem, A Figura 2 pode ser utilizada para melhor visualizar a dimensão da ML, seus tipos, algoritmos e métodos. Notem que além da Aprendizagem Clássica (CL) e da Aprendizagem por Reforço, descritos acima, um dos tipos mais relevantes e que merece maior destaque atualmente é a aprendizagem baseada em Redes Neurais (em específico, e já comentada anteriormente, a Aprendizagem Profunda). É especialmente esse tipo ou subárea, que atualmente está sendo mais utilizada para o processo de reconhecimento de padrões/visão computacional.

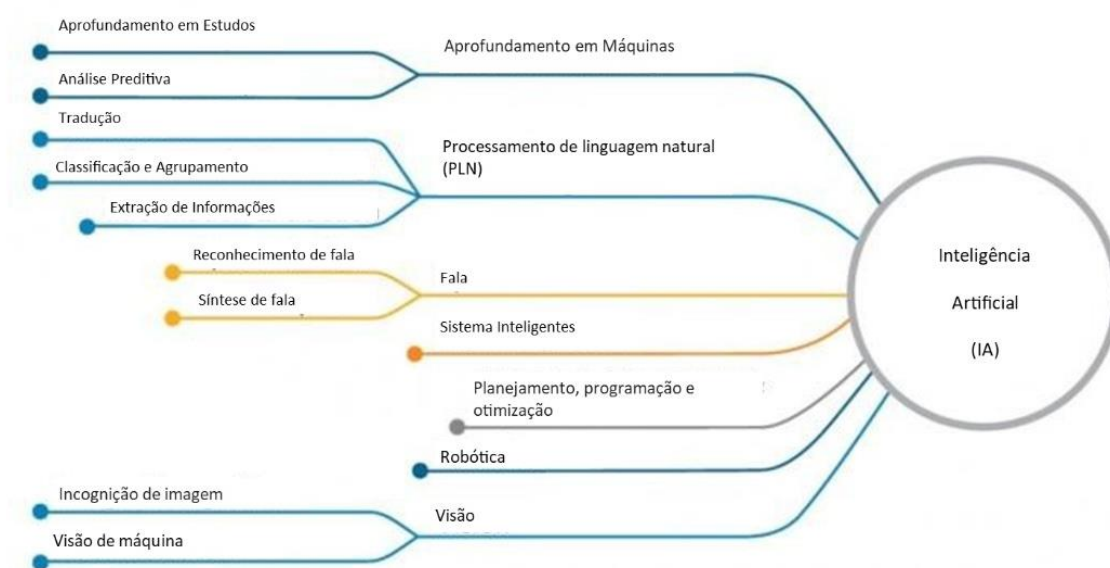
Figura 2 - Tipos, algoritmos e métodos de ML.



Fonte: Adaptado de <https://br.pinterest.com/>

Dentro da IA existem diversas subáreas, a visão computacional que será utilizada nesse presente trabalho também se encontra nesse conjunto, e atualmente recebe especial destaque. Note que todas as subáreas se utilizam da Aprendizagem de Máquina para melhorar seu desempenho e performance.

Figura 3 - Subáreas da IA



Fonte: Adaptado de <https://startecexp.com/>

1.2 Visão computacional

Se a IA é a simulação da inteligência Humana, então, pode-se dizer que, a visão computacional é o campo que ensina máquinas a compreender e interpretar o mundo visual, como é mencionado pelos pesquisadores do MIT “ensinar computadores a enxergarem como humano” (GREENEMEIER, 2008)

De acordo com Milano e Honorato (2010) a visão computacional (VC) é a ciência que estuda a forma como a máquina enxerga o meio a sua volta, coletando informações por meio de imagens extraídas de câmeras, sensores, satélites, entre outros.

A abordagem da VC dentro da área de aplicação de segurança, segundo Zhanga *et al.*, (2017) é a “detecção automática de eventos precursores de transposição de cercas baseado na análise de dados de câmeras de vigilância por visão computacional”. Existem algumas subáreas dentro da VC, sendo as principais:

- **Reconhecimento:** classificação e identificação de objetos encontrados em uma imagem por meio de algoritmos.
- **Análise de movimento:** realizar cálculos feitos sobre a velocidade, posição, referenciais etc. do objeto identificado na imagem de entrada.

- **Restauração 3D e de imagem:** Utilização de algoritmos de reconstrução e modelagem, para geração de modelos tridimensionais e melhoramento da qualidade da imagem.

Para a realização do presente projeto será utilizada a subárea de reconhecimento, onde será realizado o reconhecimento de padrões, extraindo assim os caracteres da placa dos veículos a partir de uma imagem de entrada por meio de uma câmera.

1.3 Reconhecimento de padrões

Reconhecimento de Padrões é a” ciência que tem por objetivo a classificação de objetos em um determinado número de categorias ou classes a partir da observação de suas características” (THEODORIDIS; KOUTROUMBAS, 2003). “[...]Refere-se a classe ao conjunto de padrões com características semelhantes, de acordo com algum critério”. (JÚNIOR, 2010)

A área de reconhecimento de padrões é importante devido aos acontecimentos humanos terem formas de padrões, como o modo de falar, figuras em uma imagem, uma linguagem, tudo tem o envolvimento de padrões. (PAO, 1989 apud MORAIS,2010)

Para a realização do reconhecimento de placas são necessárias 4 etapas: Pré-processamento, identificação da placa, extração dos caracteres e reconhecimento desses caracteres. (BIAO, 2016)

1.3.1 Pré-processamento

Segundo Trentini *et al.* (2010), para que ocorra sucesso nas fases de identificação, é necessário realizar esse pré-processamento para implementar correções de distorções geométricas e eliminações de ruídos, já que as fotos são capturadas por outros dispositivos.

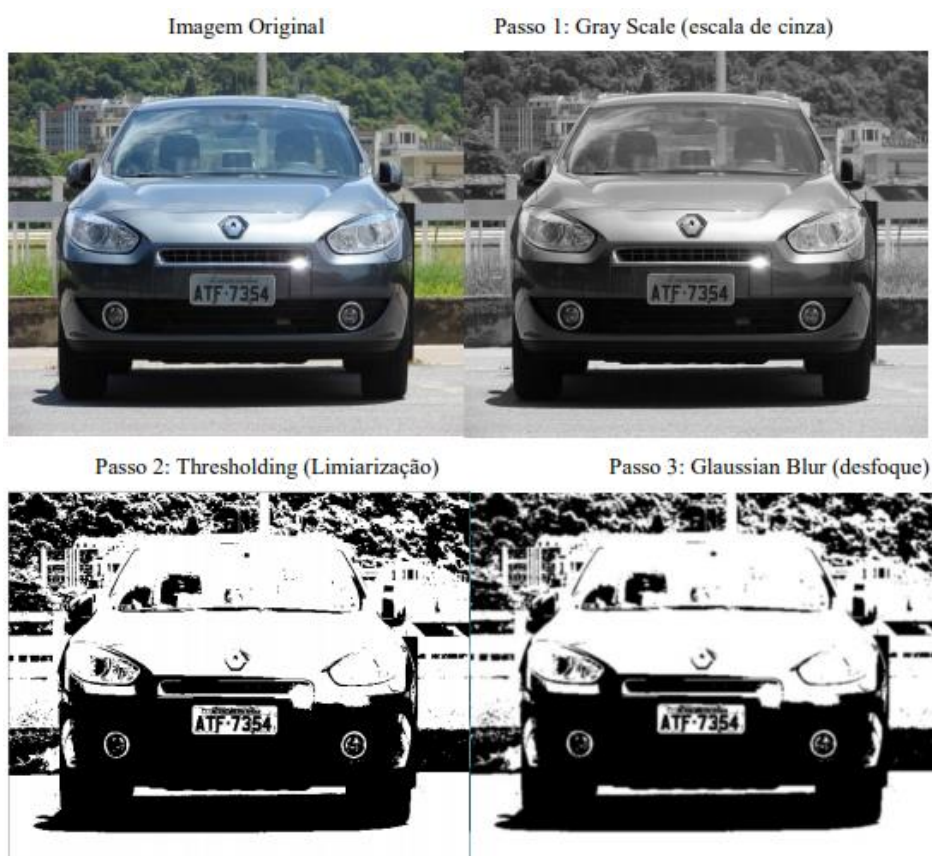
Para Campos (2001), essa etapa se utiliza de algoritmos que realcem os dados necessários e elimine os dados que atrapalhariam a segmentação. Alguns dos algoritmos tem como função realizar a limiarização, exclusão de linhas e a suavização.

Barbosa (2017) no seu trabalho de conclusão de curso voltado a reconhecimento de placas veiculares utilizou filtros nessa etapa como *Gray Scale* (Escala de Cinza), *Thresholding* (limiarização) e *Gaussian Blur* (desfoque), utilizaram de funções existentes desses filtros dentro da biblioteca OpenCV

“A Limiarização ou *Thresholding*, é uma das abordagens para a segmentação de imagens das mais importantes, essa é uma técnica de análise por região particularmente útil para cenas que contêm objetos sobre um fundo contrastante” (CUNHA, T. N. 1988 apud BARBOSA, 2017). Para essa etapa é necessário a entrada de uma imagem em escala cinza, que foi resultado da aplicação de um método específico que transforma a imagem em escala de cinza (como em OpenCV *Gray Scale*). O resultado desse processo é uma imagem binarizada.

Para eliminação de ruídos que ainda podem atrapalhar a segmentação da placa, podem ser utilizados filtros responsáveis pela suavização. Nesse contexto do referido trabalho, foi utilizado do filtro *Gaussian Blur*. O resultado de sua aplicação são, bordas borradas.

Figura 4 - Resultado dos filtros utilizados na etapa de Pré-processamento



Fonte: Barbosa (2017)

1.3.2 Identificação da placa

Segundo Biao (2016) essa fase tem como objetivo isolar a placa do automóvel de todo o restante da imagem, também conhecida como identificação da região da placa. Para isso se

segue algumas características da placa que a destacam sobre a figura do carro como um todo. Esse processo é conhecido como segmentação.

O processo de segmentação pode ser entendido como o particionamento de uma imagem em regiões que apresentem propriedades semelhantes, como textura ou cor. O princípio da segmentação foi apresentado por psicólogos alemães, quando foi mostrado que o ser humano, no processo de visão, realiza naturalmente o agrupamento de regiões por critério baseados na proximidade, similaridade e continuidade. (FACON, 1998 apud BARBOSA, 2017).

São vários os tipos de segmentação segundo Facon (2002):

- **Segmentação por região:** Região é um conjunto de pontos ligados, as regiões são ‘homogêneas’, elas apresentam alguma propriedade local, aproximadamente, constante. Alguns métodos utilizados para particionamento em região é a limiarização ou divisão e fusão.
 - **Divisão e fusão:** “Permite agrupar pixels para criar regiões as quais um predicado de homogeneidade é verificado”.
- **Segmentação por textura:** “Deve permitir a colocação em evidência dos motivos da imagem considerando-se as propriedades de regularidade e de repetição”. Os métodos principais são a estatística, estruturais e espectrais.
 - **Estatísticas:** Permitem caracterizar textura suave, irregular, granulada etc. “Essa técnica trata da disposição primitivas em uma imagem.”
 - **Estruturais:** “Construir, a partir de uma primitiva de textura simples, texturas mais complexas, usando regras que permitem limitar o número de arranjos possíveis das primitivas.”
 - **Espectrais:** Permite a detecção da direcionalidade de motivos periódicos em uma imagem.
- **Segmentação por contorno:** “Contorno é uma mudança brusca do nível de cinza entre duas regiões relativamente homogênea”. Se utiliza de métodos como detecção dos pontos e detecção de bordas.
 - **Detecção de pontos:** Segmentar por pontos particulares relevantes, esses são pontos isolados que devem conter bastante informações e não podem ser deteriorados por ruídos. “Um ponto ele é altamente caracterizável se e somente se ele possuir uma variância alta segundo quatro direções: vertical, horizontal e as duas diagonais principais.”
 - **Detecção de bordas:** Consiste em detectar descontinuidade nos níveis de cinza e assim conseguir fornecer contornos em região distintas.

1.3.2.1 Segmentação por contornos

Segundo Chagas *et al.* (2021) A segmentação é uma das etapas mais importantes para realizar uma análise de uma imagem e obter informações qualitativas. Primeiramente, segmentar uma imagem consiste em separá-la em regiões com objetos na mesma imagem ou fundo com base em algum critério.

A segmentação pode seguir por duas principais estratégias sendo elas a similaridade e a descontinuidade onde se encontra a detecção de contornos (SOBRAL, 2002).

Um contorno se trata da união de pontos de fronteira de um objeto, para realizar esta união normalmente se leva em consideração uma área específica de relação entre pixels da imagem com as cores e intensidade parecidas. A biblioteca OpenCV possui duas funções que trazem uma facilidade para encontrar e desenhar os contornos na imagem, conhecidas como *findContours* e *drawContours* além de outros métodos que também facilitam as detecções como o *CHAIN_APPROX_SIMPLE* e *CHAIN_APPROX_NONE* (RATH, 2021). Geralmente utilizada em um “roi” que consiste em uma sub-região onde nosso objeto de interesse se encontra (SONAWANE, 2020).

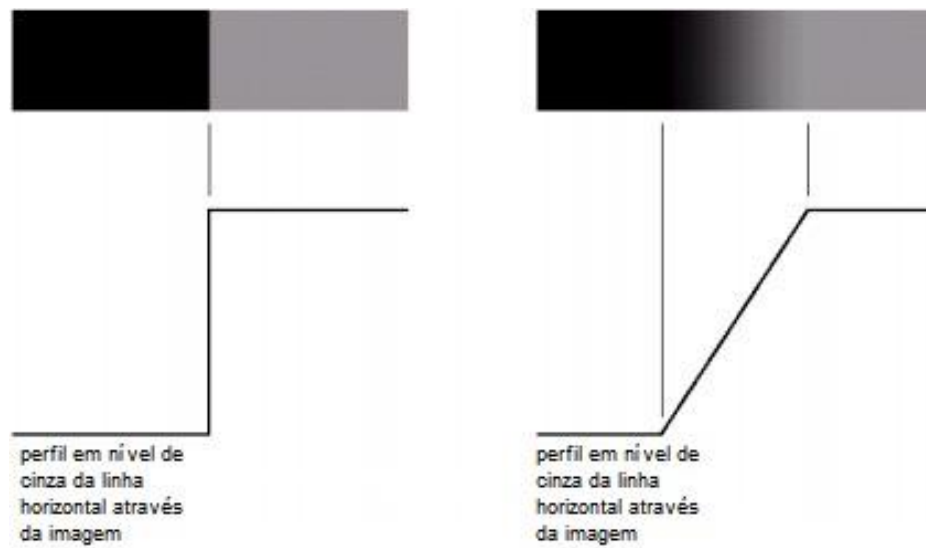
Segundo Barbosa (2017) para realizar uma boa segmentação por contornos é importante que alguns passos e aplicações de filtros sejam realizados na imagem para que melhore a taxa de sucesso na detecção dos contornos, entre elas estão a leitura da imagem e conversão para cinza, etapa que é importante para a próxima que consiste em limiar a imagem onde os objetos tornam-se totalmente pretos e brancos fazendo com que o objeto de interesse tenha suas bordas brancas e com mesma intensidade. Após esta etapa é utilizada as funções para encontrar os contornos e desenhá-los na imagem em RGB² utilizando as bibliotecas do OpenCV.

A Figura 5 representa o método mais comum com a finalidade de detecção de descontinuidades, esse método localiza certas imperfeições na imagem e transforma os contornos em tipos de “rampas”.

¹ Roi (*Region Of Interest*) é a região de interesse de uma imagem, todo processamento terá total concentração nessa região.

² RGB (*Red, Green, Blue*) é um sistema de cores onde combina as cores vermelha, verde e azul de diversas formas, formando outras diversas cores.

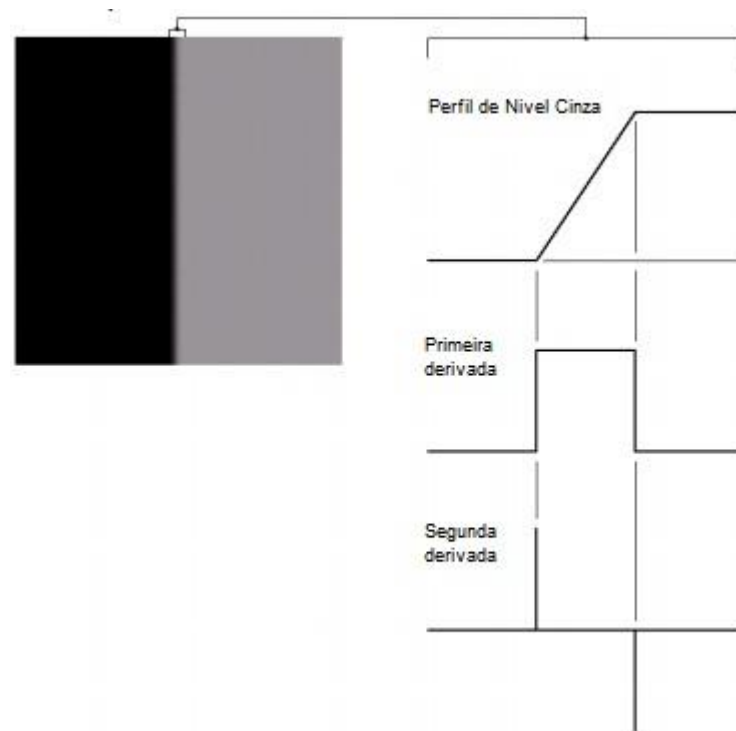
Figura 5 - método de detecção de descontinuidades



Fonte: Adaptado de Sobral (2002).

A Figura 6 é utilizado a magnitude da 1ª derivada e as passagens por zero da 2ª derivada para detecção de contornos.

Figura 6 – Detecção de contornos por derivadas



Fonte: Adaptado de Sobral (2002)

1.3.3 Extração dos caracteres

Segundo Biao (2016) essa fase é conhecida como segmentação dos caracteres, tem por objetivo separar cada caractere em imagens distintas, sendo permitido assim a identificação de cada letra e número que compõe a placa separadamente. O sucesso desta etapa está ligado ao resultado do pré-processamento.

Para essa etapa a revisão bibliográfica nos indica a utilização de uma biblioteca chamada Tesseract. A função dela nesse contexto é analisar o texto da imagem e quebrá-la em palavras a partir do espaçamento que existe entre os caracteres. (TESSERACT-OCR, 2014).

1.3.4 Reconhecimento dos caracteres

Segundo Oliveira e Gonzaga (2012) essa é a etapa onde se utiliza de rede neurais para o processo de segmentar o objeto e analisá-lo a um único caractere alfanumérico. Um sistema com base em redes neurais utiliza de um banco de dados específico para o treinamento, para realizar a análise das placas dos veículos.

Com a biblioteca Tesseract, os caracteres encontrados na etapa anterior são classificados pelo classificador de idioma uma por uma. Assim que reconhecida vira um dado de treinamento. (TESSERACT-OCR, 2014).

O resultado que será obtido dessa etapa será uma *string*, no caso um texto, como pode-se visualizar na Figura 7.

Figura 7 - Resultado do reconhecimento de caracteres com OpenCV e Tesseract



“ATF7354”

Fonte: Barbosa (2017)

1.4 Biblioteca OpenCV

Segundo o site oficial (<http://opencv.org/>), o OpenCV é uma biblioteca para visão computacional de código aberto, com o objetivo de fornecer uma infraestrutura comum em

aplicativos de visão computacional. Essa biblioteca conta com mais de 2500 algoritmos otimizados, com a intenção de serem utilizados para detectar e reconhecer rostos, identificar objetos, classificar ações humanas em vídeos, rastrear objetos em movimento, extrair modelos 3D de objetos entre outros. Também tem atuações em diversas plataformas como Windows, Linux, Mac OS, IOS e Android, e linguagens como C e Python.

Sendo desenvolvida pela Intel, ela foi criada para dar a oportunidade de tornar a Visão computacional acessível a programadores com um baixo investimento e custo de operação.

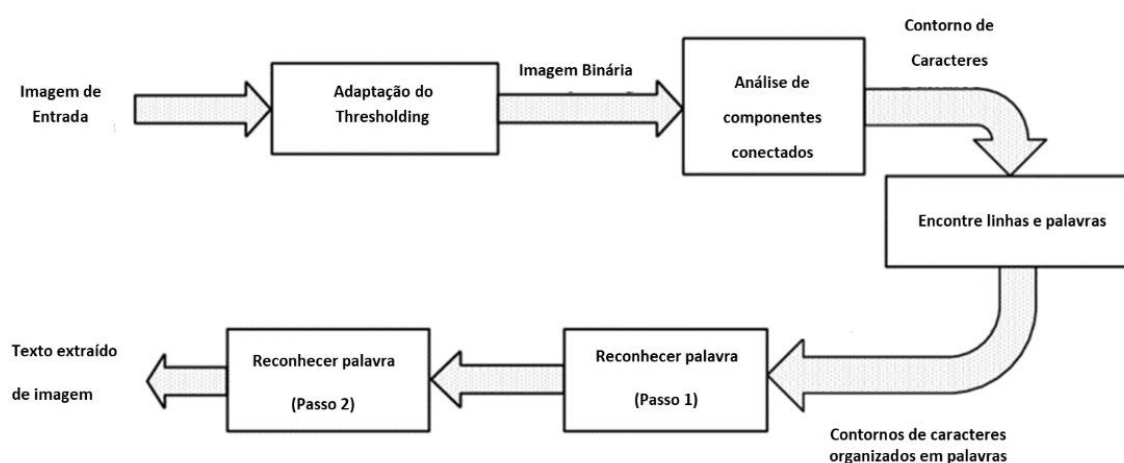
1.5 Biblioteca pytesseract

Segundo o site oficial (<https://pypi.org/>), “Python-tesseract é uma ferramenta de reconhecimento óptico de caracteres (OCR) para a linguagem Python, ou seja, ele reconhecerá e “lerá” o texto embutido nas imagens.”

É uma tecnologia de código aberto com objetivo de reconhecimento de caracteres. Foi desenvolvido pela empresa *Hewlett Packard* entre 1985 e 1995 e pelo Google a partir de 2006. É uma das API que é considerada uma das melhores da área de OCR e pode ser utilizada como um software independente. (HOSTING, 2014 apud NARDI et al., 2015)

O funcionamento do Tesseract, utiliza-se de uma entrada de dados (imagem) na qual é binarizada, caso já não esteja pré-processada. Separando os objetos em blocos de texto, linhas e figuras onde esses objetos são analisados delimitando linhas em torno dos objetos referentes ao texto. (BARBOSA, 2017)

Figura 8 - Arquitetura da Tesseract



Fonte: Adaptado de <https://www.researchgate.net/>

Pode-se ser observado que na Figura 8 ocorre um segundo processamento das palavras. Isso ocorre quando as palavras são consideradas perigosamente ambíguas (TESSERACT-OCR, 2014). Um exemplo disso seria uma interpretação confusa entre uma letra “O” e o número 0 (BARBOSA, 2017).

1.6 Raspberry

Com o avanço tecnológico de novos *hardwares*, surgiu nos últimos tempos a computação conhecida de ultrabaixo custo, que utilizam hardware periféricos de entrada e saída em única placa. (HEEKS; ROBINSON, 2014)

O Raspberry é um computador em placa desenvolvido pela Raspberry Pi Foundation, geralmente de um tamanho pequeno como um cartão. O objetivo inicial é promover ensino básico de computação com um baixo custo nas escolas com baixa renda.

Por ser total *OpenSource* ele utiliza uma distribuição Unix com licença GNU, o que o ajudou a crescer em utilização e aplicação rapidamente e ser utilizado em várias áreas como processamento de dados, servidores *web* (*webserver*), robótica e outras series de projetos embarcados. (VON ATZINGEN, 2017). A Tabela 1 dá uma visão dos principais modelos do Raspberry Pi, sendo que os mais utilizados atualmente é o Pi 3 B.

Tabela 1 - Alguns Modelos atuais do Raspberry Pi

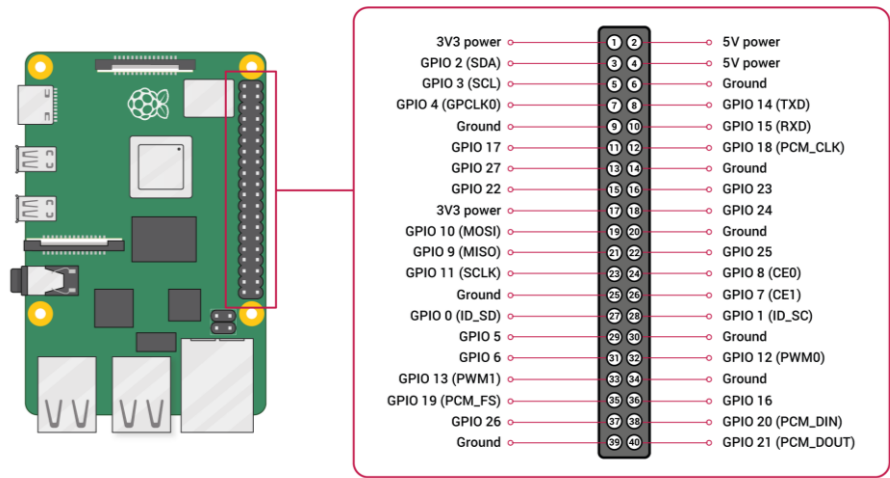
	Pi 4	Pi 3 B+	Pi 3 A+	Pi 3 B	Pi 2 B
Lançamento	Junho de 2019	Março de 2018	Novembro de 2018	Fevereiro de 2016	Fevereiro de 2015
Preço	R\$ 464,90	R\$ 269,90	R\$ 160,90	R\$350,00	-
Processador	Broadcom (CortexA72 quad-core de até 1,5 GHz	Broadcom 1,4 GHz quad-core	Broadcom 1,4 GHz quad-core	Broadcom 1,2 GHz quad-core	Broadcom 900 MHz quad-core
Memória RAM	1GB, 2GB OU 4GB de DDR4	1GB	512MB	1GB	1GB
Conectividade	USB-C (energia), 2x USB 3.0, 2x USB 2.0, 2x micro HDMI,	HDMI, conector de câmera CSI, conector de display DSI,	HDMI, conector de câmera CSI, conector de display DSI,	HDMI, conector de câmera CSI, conector de display DSI,	HDMI, conector de câmera CSI, conector de display DSI,

	fone de ouvido, micros, Gigabit Ethernet, Wi-Fi, Bluetooth e GPIO	saída de áudio P2, microSD, GPIO	saída de áudio P2, microSD, GPIO	saída de áudio P2, microSD, GPIO	saída de áudio P2, microSD, GPIO
--	---	---	---	---	---

Fonte: Adaptado de Raspberry Pi Foundation

Alguma das características importantes dos Raspberry são a interface de entrada e saída GPIO (do inglês, *General Purpose I/O*, ou seja, Entrada/Saída de Propósito Geral), sendo a principal maneira para conectá-las a outros dispositivos. Existem pinos no qual são organizados em duas colunas que aceitam comandos de entrada e saídas. São por esses pinos que se recebe dados externos. Alguns pinos em específicos têm funções distintas, como algumas com entrada e saída digital e algumas que servem para alimentação. (ALVES FILHO, 2017). A Figura 9 apresenta a relação da pinagem da GPIO de um Raspberry padrão.

Figura 9 - Linha de pinos GPIO Raspberry



Fonte: Raspberry Pi Foundation

Além dos pinos GPIO existem outras portas na placa para uso de um HD externo, cartões microSD ou uma câmera por exemplo.

Figura 10 - Conectores da Raspberry



Fonte: <https://www.filipeflop.com>

1.6.1 Raspicam

De acordo com o site oficial (www.raspberrypi.org) a Raspicam é um produto oficial da Raspberry Pi Foundation, tendo seu modelo original de 5MP lançado em 2013, e em 2016 o lançamento da câmera module V2 com 8MP e em 2020 (ano atual deste trabalho), foi lançada a câmera HQ de 12MP. Como algumas webcams não são compatíveis com a placa Raspberry, optamos pela utilização da Raspicam pela certeza da compatibilidade. A Figura 11 ilustra um modelo intermediário da Raspicam.

Figura 11 - Imagem da Câmera Raspberry Pi v2



Fonte: <https://www.filipeflop.com>

Tabela 2 - Especificação de hardware

	Módulo de câmera v1	Módulo de câmera v2	Câmera HQ
Preço líquido	\$ 25	\$ 25	\$ 50
Tamanho	Cerca de 25 × 24 × 9 mm		38 x 38 x 18,4 mm (excluindo lente)
Peso	3g	3g	
Ainda em resolução	5 megapixels	8 megapixels	12,3 megapixels
Modos de vídeo	1080p30, 720p60 e 640 × 480p60 / 90	1080p30, 720p60 e 640 × 480p60 / 90	1080p30, 720p60 e 640 × 480p60 / 90
Integração Linux	Driver V4L2 disponível	Driver V4L2 disponível	Driver V4L2 disponível
API de programação C	OpenMAX IL e outros disponíveis	OpenMAX IL e outros disponíveis	
Sensor	OmniVision OV5647	Sony IMX219	Sony IMX477
Resolução do sensor	2592 × 1944 pixels	3280 × 2464 pixels	4056 x 3040 pixels
Área da imagem do sensor	3,76 × 2,74 mm	3,68 x 2,76 mm (4,6 mm na diagonal)	6,287 mm x 4,712 mm (7,9 mm na diagonal)

Fonte: Raspberry Pi Foundation

1.7 Arduino

O Arduino é uma ferramenta simples de *hardware* e *software* que auxilia a programação de um microcontrolador capaz de se comunicar com outros eletrônicos, tendo o objetivo de realizar funções manuais do nosso dia a dia de forma automatizada. (RODRIGUES, 2018)

Segundo o Soldafria (2019), o Arduino Uno é uma placa que tem como microcontrolador o ATmega328P. Ele possui 14 pinos digitais e 6 pinos analógicos, totalizando 20 pinos, sendo eles utilizados para entrada e saída (6 desses 14 pinos digitais podem ser utilizados como saída PWM). O Arduino serve para ensinar pessoas a programar, orientar na produção de projetos de eletrônica, podendo envolver robótica, ou até projetos voltados a internet.

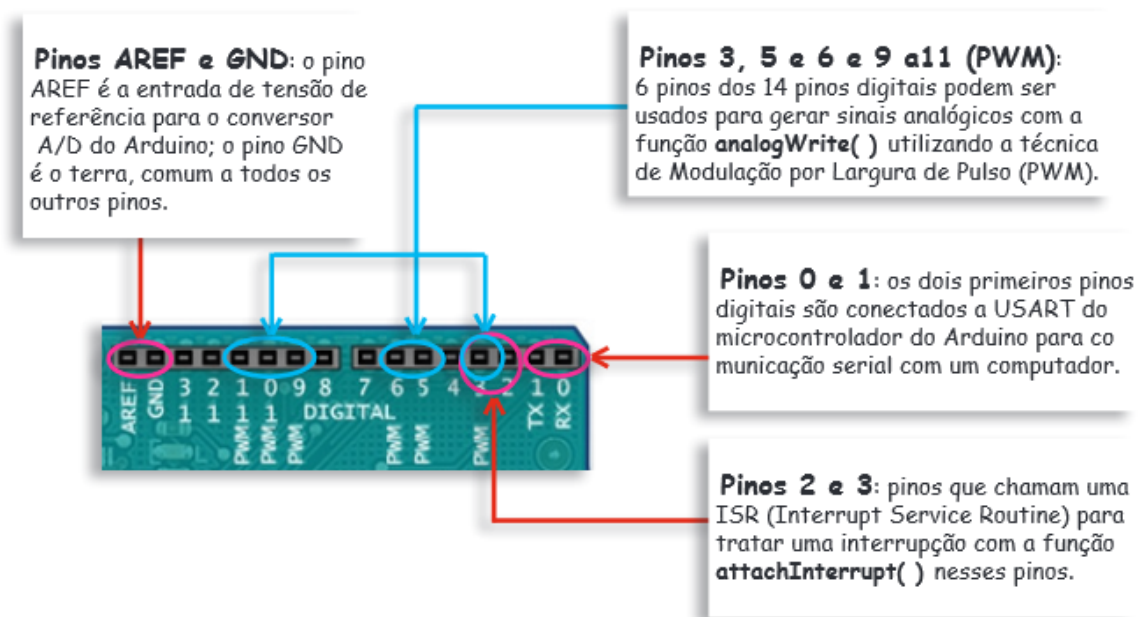
A Figura 12 ilustra todos os componentes do Arduino Uno e a Figura 13 especifica os pinos digitais.

Figura 12 – Componentes do Arduino Uno



Fonte: Daniel Paz de Araújo

Figura 13 – Especificação dos pinos da placa Arduino Uno



Fonte: <https://wiki.sj.ifsc.edu.br/>

O microcontrolador ATmega328P é o responsável pelo funcionamento da placa Arduino UNO. É ele quem possibilita o upload do código (binário) que é gerado assim que a compilação do programa é feita pelo usuário. A placa Arduino UNO é programada através da comunicação serial. (SOUZA, 2013)

1.8 Trabalhos Relacionados

Neste tópico serão apresentados estudos relacionados com o tema de nosso trabalho. O levantamento foi orientado para a busca de pesquisas científicas ou tecnológicas.

Almeida (2018) abordou um estudo de caso para reconhecimento automático de placas veiculares usando a linguagem Python onde aplica técnicas de PDI (Processamento Digital de Imagens), técnica na qual segundo Queiroz (2016) envolve a captura de uma imagem através da iluminação que é refletida nas superfícies dos objetos e é realizada através de um sistema de aquisição. Após isto a imagem deve ser representada de forma apropriada para o tratamento computacional. E utilizaram abordagem de redes neurais para solucionar o problema de reconhecimento de placas em ambientes de iluminação variada. Também mostrou alguns algoritmos de aprendizado de máquinas, *deep learning* e localização de veículos. Utilizou de bibliotecas como OpenCV e o TensorFlow.

Nardi et al (2015) realizou um projeto de controle de acesso utilizando o reconhecimento de caracteres em placas de automóveis, onde utilizaram unidades controladoras do sistema Raspberry Pi de baixo custo. Utilizou a biblioteca ALPR (*Automated License Plate Recognition*) desenvolvida em linguagem C++ para reconhecer automaticamente os caracteres de uma imagem, segundo o autor ela já é capaz de reconhecer uma grande quantidade de caracteres alfanuméricos de placas de diversos países, modificando alguns pontos para melhorar o uso para o projeto. Obtendo uma margem de acerto de 70% no reconhecimento das placas com uma velocidade de 30 segundos.

Ainda, segundo Nardi houve a conclusão de que não foi possível ter 100% do reconhecimento de caracteres e em alguns casos nem foi possível o reconhecimento e que a biblioteca ALPR possui falhas no reconhecimento de alguns caracteres, porém o grupo pretende em projetos futuros incluir layouts e códigos para o reconhecimento de placas brasileiras para a biblioteca.

Paes (2017) realizou um projeto para reconhecimento automático de placas automobilísticas em tempo real no Brasil capturados por câmeras de segurança visando a segurança como tema importante devido aos altos níveis de violência no país. Utilizou de capturas de vídeo na entrada e saída de um condomínio. O projeto mostrou que a utilização do método *Haar* foi satisfatório, segundo o autor os atributos *Haar* foram baseados na utilização de formas retangulares simples para realçar as características presentes em rostos, como

simetria, diferença de tonalidade e outras peculiaridades. E se mostrou muito boa para aplicar em imagens por conta das mudanças nos pixels.

Quando a iluminação se encontrava de forma adequada tiveram objetivo alcançado em 98,4% das vezes segundo o autor. A etapa de pré-processamento obteve sucesso acima de 90%. Também chegaram à conclusão de que a câmera deveria ser de maior resolução para melhor qualidade de vídeo e assim poder utilizar as técnicas de pré-processamento.

Cerqueira (2016) realizou seu trabalho de conclusão de curso para a implementação de detecção de placas utilizando a biblioteca OpenCV, realizando uma revisão bibliográfica para mostrar todo o tema e suas abordagens. Obteve sucesso em 40 imagens com iluminação variada em 100 imagens testadas.

O Cerqueira mostra que o estudo tinha como foco explorar todas as etapas do processo de reconhecimento de identificação, mas realizou apenas a detecção da placa em si. Também diz que o algoritmo utilizado consegue bons resultados apenas com alto nível de luminosidade e também ressalta a possibilidade de realizar novos projetos utilizando outras técnicas descritas em diversas referências para produzir um sistema mais completo.

O Quadro 1 apresenta um resumo dos trabalhos correlatos analisados, destacando seus objetivos e a peculiaridade de cada um de todos os estudos.

Quadro 1 - Caracterização dos trabalhos relacionados

Autor/Ano	Objetivo	Peculiaridade
Almeida (2018)	Realizar o reconhecimento de placas utilizando a linguagem de programação Python.	Aplicação técnicas de PDI e abordagem de redes neurais.
Nardi et al (2015)	Controlar de acesso utilizando o reconhecimento de caracteres em placas de automóveis.	Uso da biblioteca ALPR.
Paes (2017)	Reconhecimento automático de placas automobilísticas em tempo real no Brasil capturados por câmeras de segurança.	Método Haar em cascata.
Cerqueira (2016)	Implementação de detecção de placas utilizando a biblioteca OpenCV.	Utiliza técnicas de operação morfológicas.

Fonte: Próprios autores.

CAPÍTULO 2 TECNOLOGIAS UTILIZADAS

Após realizada a modelagem do sistema verificou-se com maior precisão as tecnologias necessárias para melhor desenvolvimento do projeto, para melhor compreensão do projeto será explanado as principais características de cada uma das tecnologias a serem utilizadas neste projeto.

2.1 Python

Trata-se de uma linguagem de programação de alto nível, orientada a objetos de scripts, funcional de forma que produza códigos que sejam fáceis de manter de maneira rápida, atualmente é mantida de forma colaborativa e aberta.

Segundo Borges (2014) “O Python tem uma sintaxe clara e concisa que favorece a legibilidade do código-fonte, tornando a linguagem mais produtiva.”. No próprio site oficial da linguagem Python temos a seguinte definição: “Python é uma linguagem de programação que permite trabalhar rapidamente integrar sistemas de forma mais eficaz.” (Python, 2020).

Para compreender melhor a escolha da linguagem usada nos projetos, estudamos outros trabalhos que a linguagem Python foi utilizada. Segundo Almeida (2017), “[...] a linguagem é uma das mais utilizadas no âmbito de visão computacional”.

2.2 Protocolo HTTP

O HTTP é um protocolo cliente-servidor, onde o cliente (*client-side*), que no caso poderia ser um usuário da web solicita uma requisição para o servidor(*server-side*) e o servidor vai lidar com a requisição e devolver uma resposta para o cliente.

Este protocolo permite a obtenção de recursos, utilizado para sistemas de informações de hipermídia e distribuídos. Desenvolvido para a comunicação entre os navegadores podendo ser utilizados para outras finalidades.

É utilizado para buscar links Hipertexto (HTML), como também enviar imagens, vídeos e até publicar conteúdos no servidor. (MDN, 2020).

2.3 JavaScript

Projetada para programar o comportamento de uma página web a partir de algum evento, é uma linguagem baseada em protótipos, dinâmica e multi-paradigma que suporta estilos orientados a objetos ou procedural, quanto a orientação de objetos, esses objetos são criados em tempo de execução onde métodos e propriedades são anexados a objetos vazios, ou seja, não há definições de classes como encontramos em outras linguagens.

O JavaScript está entre as três principais tecnologias da *World Wide Web*, também é a principal linguagem de programação *client-side* em navegadores web, mas atualmente já existe a possibilidade de utilizá-la como uma linguagem *scripting* para o *server-side* (MDN, 2019).

2.4 HTML5

Hyper Text Markup Language (ou Linguagem de marcação de hipertexto) também conhecido como HTML, consiste em uma linguagem de marcação para a construção de páginas web, anotando textos, imagens e outros conteúdos para exibição. Esta tecnologia foi escolhida para trabalhar em conjunto com o React mais precisamente com o HTML sendo inserido dentro do projeto React.

A versão HTML5 abrange novos elementos e atributos como conectividade, semântica permitindo uma melhor descrição de seus conteúdos, entre outras melhorias. (MDN, 2019).

2.5 React JS

É uma biblioteca de JavaScript de código aberto que tem como foco a criação de interfaces de usuário conhecidos como *frontend* nas páginas web, onde se pode combinar o HTML e o JavaScript. Atualmente é mantido pelo Facebook, Instagram entre outras empresas e uma comunidade de desenvolvedores individuais.

Possui lógicas de estado e componentes, onde os componentes são encapsulados e podem ser reutilizados em outras aplicações de mesmas funções, os componentes são renderizados na medida em que ocorre a mudança de dados (REACT, 2020).

2.6 Node JS

O NodeJs é uma tecnologia *backend* que utiliza da linguagem JavaScript, porém voltada para o *server-side*, ou seja, essa tecnologia nos permite utilizar uma linguagem que é usada somente para o *client-side* para rodar em um servidor sem precisar do browser. O Node é utilizado por grandes empresas no mercado de tecnologia, como a Uber, LinkedIn e a Netflix. Uns dos motivos para sua utilização é a alta capacidade de escala, flexibilidade e a leveza que ela possui.

A vantagem de se utilizar JavaScript no *server-side* é que ela possui sua execução em *single-thread*, ao contrário de outras linguagens que demandam de *mult-thread*, ou seja, ela trata todas as requisições com apenas uma única thread e as chamadas as requisições são assíncronas, não dependendo da termino de outra para continuar pois são tratada como um evento, isso faz com que demande menos recursos computacionais. (LENON, 2018).

2.7 Express

Framework Node que oferece gerencia de requisições em diferentes URLs e definir configurações de servidor como porta, localização de modelos e utilizaram de middleware especificos em qualquer ponto das requisições.

Express é popular com base no número de empresas que utilizam, no número contribuição de pessoas no código base e de no numero de pessoas que dão suporte. (MDN, 2019).

2.8 Bcrypt

Um método de criptografia para senhas do tipo *hash*³ e que foi baseado no *Blowfish* que é uma cifra simétrica de blocos. Seus criadores foram Niels Provos e David Mazières apresentando na conferência da Usenix em 1999.

Sua função é 'esconder' a senha de um usuário, o seu *salt*⁴ acrescenta de forma aleatória várias sequencias de caracteres, criando senhas criptografadas complexas e seguras,

³ Hash é um algoritmo matemático que transforma qualquer bloco de dados em uma série de caracteres de comprimento fixo.

⁴ O *salt* é a adição de caracteres, palavras, termos ou mesmo números que dão uma certa aleatoriedade aos algoritmos e ajudam a torná-los indecifráveis.

podendo assim, armazenar essa senha criptografada em bancos de dados, já que a própria criptografia não libera seu acesso ao site, e sim somente a senha original.

2.9 MySql

O mySql é um sistema gerenciador de banco de dados relacional que utiliza da linguagem SQL, sendo essa a linguagem mais popular para inserir, consultar e gerenciar os registros armazenados no banco de dados.. O MySQL é utilizada na maioria das aplicações gratuitas.

2.10 Json web token (JWT)

É um padrão aberto que define um modo de transmitir informações com segurança. Essas informações são assinadas digitalmente. Os JWTs podem ser criptografados. Há uma diferença entre os tokens assinados e os criptografados, onde os assinados verificam a integridade das declarações obtidas, já os tokens criptografados ocultam essas declarações.

Duas situações em que o JWT é utilizado são para realizar autorizações e troca de informações, um exemplo é quando utilizamos de rotas seguras, onde só é possível realizar uma requisição e obter uma resposta quando o usuário está autenticado no sistema. (JSON, 2020).

2.11 Material-table

O material-table é uma tabela preparada com ações prontas de paginação, filtragem, ordenações etc. Seu componente é baseado no Material Ui, que é uma biblioteca de componentes React prontos para utilização, o que torna o desenvolvimento mais fácil e rápido, sem necessitar da criação de componentes complexos do zero.

2.12 Cloudinary

O serviço Cloudinary conta com inúmeros servidores de upload e transformação de imagens e vídeos com foco em lidar com altas taxas de uploads, também escalam automaticamente para gerenciamento facilitado de grandes picos além de contar com uma

rede de distribuição em nível empresarial por meio de vários servidores. Sua segurança é baseada em nuvem e está acessível por meio de APIs HTTPS autenticadas.

O uso principal desse serviço foi servidor de uploads para o armazenamento das imagens capturadas pela Raspicam.

A Figura 14 são representações e logos das tecnologias citadas nesse tópico seguindo sua ordem de citação.

Figura 14 – Representações e logos das tecnologias utilizadas



Fonte: Próprios autores.

CAPÍTULO 3 MODELAGEM DE DADOS DA APLICAÇÃO

Nesse capítulo trataremos a modelagem do presente projeto. Será utilizada a Linguagem de Modelagem Unificada (UML, do inglês, *Unified Modeling Language*) da OMG.

Segundo o site oficial da UML (<https://www.uml.org/>) um sistema deve ser estruturado de forma que permita estabilidade, segurança e execução robusta, além de ser clara para outros programadores, assim como os quais irão fazer a manutenção, para compreender e encontrar de forma mais rápida erros e corrigi-los.

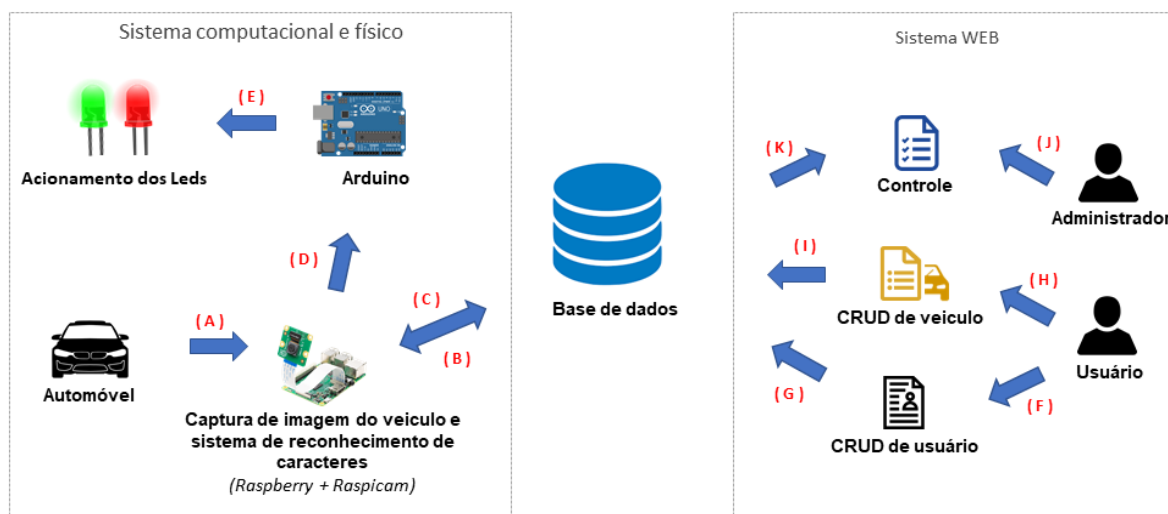
A parte de modelagem é essencial para ajudar a especificar, visualizar e documentar o software, de forma universal.

3.1 Diagrama de arquitetura de alto nível

Diagrama de arquitetura de alto nível ajuda a compreender toda a composição e funcionamento do sistema.

A Figura 15 ilustra a arquitetura física do funcionamento do sistema e, logo em seguida, a explicação de ação desses elementos componentes.

Figura 15 - Diagrama de arquitetura do sistema de reconhecimento e sistema web



Fonte: Próprios autores

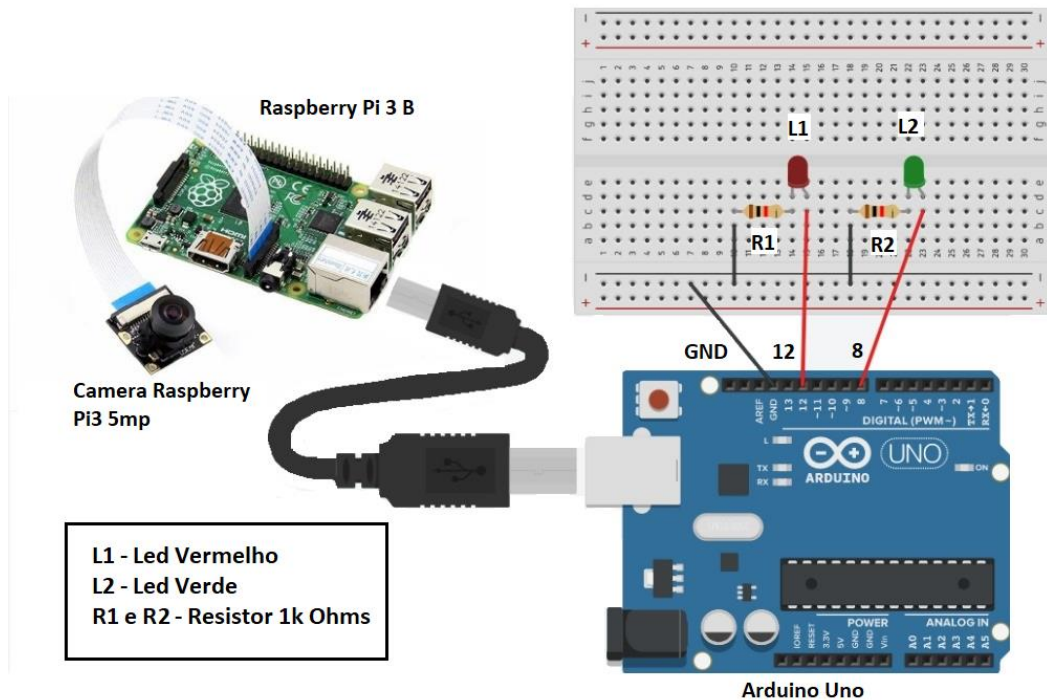
- (A) - Após a aproximação do veículo e o acionamento do botão, é realizada a captura de imagem da placa, o algoritmo de reconhecimento de caractere irá extrair todos os caracteres contidos na placa veicular.
- (B) - Após o reconhecimento dos caracteres, o algoritmo realizará uma consulta na base de dados comparando a placa detectada com as placas já cadastradas.
- (C) - A base de dados retorna o resultado da consulta de placas, se alguma placa for encontrada ocorre o registro do controle daquela placa e um retorno positivo ao sistema computacional, caso contrário, não ocorre registro de controle e o retorno é negativo.
- (D) - Se houver um retorno positivo da base de dados, o Arduino recebe um comando do Raspberry para o acionamento de um dos leds⁵.
- (E) - O Arduino após receber os comandos, aciona o respectivo led sinalizando o possível acesso ou não do veículo na instituição.
- (F) - No sistema web o usuário é capaz de realizar seu cadastro, esses dados serão necessários quando ocorrer o registro de controle de entrada e saída do veículo, para saber a qual usuário se associa a determinado veículo.
- (G) - Todos os dados do cadastro de usuário são armazenados na base de dados.
- (H) - O usuário é capaz de realizar o cadastro, edição, listagem e exclusão do seu veículo no sistema web, para assim quando necessitar entrar no local, o sistema operacional seja capaz de identificar e encontrar seu veículo na base de dados.
- (I) - Todos os dados do cadastro de veículo são armazenados na base de dados.
- (J) - O administrador do sistema web é capaz de solicitar a listagem todo controle de entrada e saída de veículos
- (K) - A base de dados retorna à consulta de todo controle de entrada e saída de veículos para a listagem.

3.2 Esquema de Ligação do Sistema Físico

O esquema de ligação ajuda a compreender a ligação física do projeto, mostrando exatamente como deve ser montado. No caso deste projeto, mostra as respectivas portas utilizadas do Arduino, os componentes físicos etc. A Figura 16 representa a ligação do sistema físico.

⁵ Os leds aqui fazem as vezes da cancela. Abrindo (led verde) ou mantendo fechada (led vermelho).

Figura 16 – Esquema de ligação do sistema físico



Fonte: Próprios autores

3.3 Requisitos funcionais

Segundo Alff (2018) “os requisitos funcionais são toda as necessidades, características ou funcionalidade esperada em um processo que pode ser atendido pelo software”.

Ainda segundo o mesmo autor:

Podemos dizer que é considerado um requisito funcional, todo cenário onde o usuário informa um dado, ou um sistema terceiro realiza uma solicitação qualquer durante uma interação com o sistema, que então, responde com determinada ação correspondente. (ALLF, 2018)

A Tabela 3 representa os requisitos funcionais desse presente sistema.

Tabela 3 - Requisitos funcionais do sistema

Identificação	Nome
RF01	Efetuar login
RF02	Realizar cadastro
RF03	Recuperar senha
RF04	Manter cadastro

RF05	Cadastrar veículo
RF06	Manter veículo
RF07	Listar controle
RF08	Captar imagem
RF09	Reconhecer placa
RF10	Consultar placa
RF11	Registrar controle
RF12	Sinalizar acesso

Fonte: Próprios autores

3.4 Requisitos não funcionais

“Diferente dos requisitos funcionais, que diz como deve fazer, os não funcionais dizem de que maneira se deve fazer, devem expressar uma premissa ou restrição do sistema.” (ALFF, 2018)

A Tabela 4 representa os requisitos não funcionais desse presente sistema.

Tabela 4 - Requisitos não funcionais do sistema

Identificação	Nome	Tipo
RNF01	Autenticação de usuários com token	Segurança
RNF02	Níveis de acesso	Segurança
RNF03	Criptografia de senhas de acesso	Segurança
RNF04	Layout web responsivo	Usabilidade
RNF05	Utilização de JavaScript para web	Padrões
RNF06	Utilização de Python para reconhecimento de placas	Padrões
RNF07	Utilização de sistema Linux no Raspberry	Compatibilidade

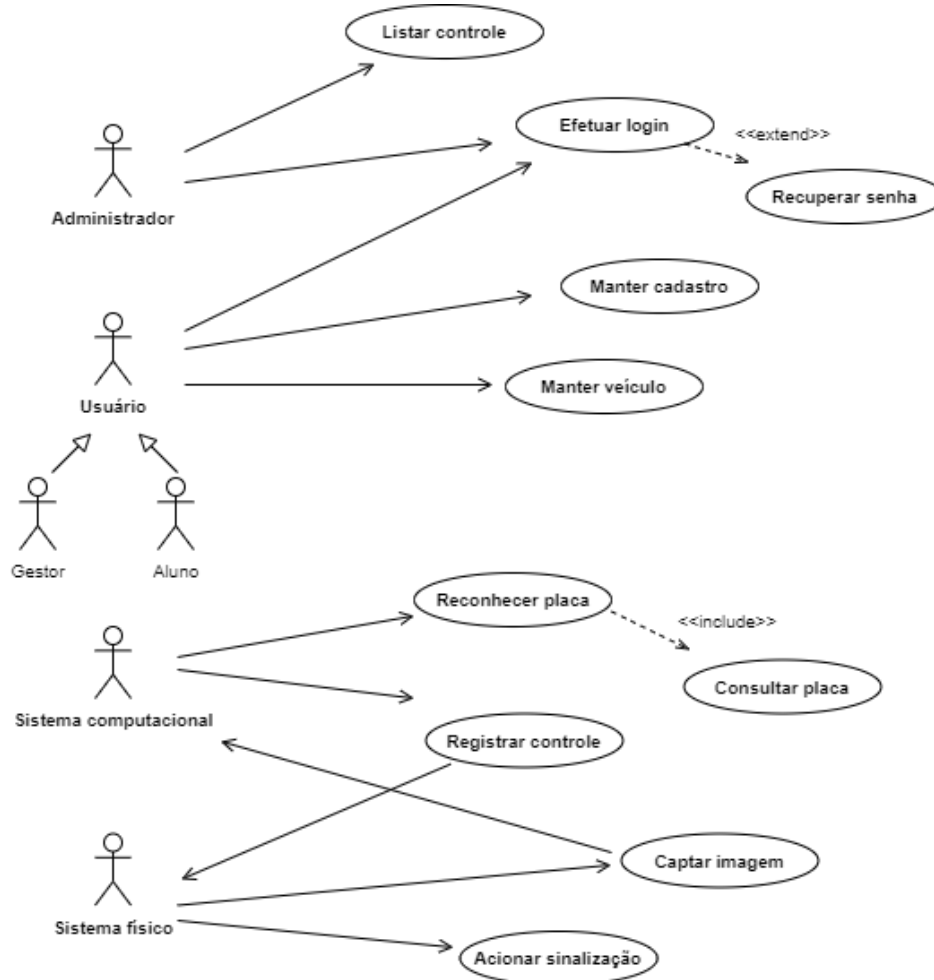
Fonte: Próprios autores

3.5 Diagrama de casos de uso alto nível

O objetivo desse diagrama é o resumo das interações dos usuários com o sistema, nele encontramos os atores e suas respectivas ações. A Figura 17 demonstra as principais

funcionalidades do sistema web de cadastro de veículos e a realização do reconhecimento de placas.

Figura 17 - Diagrama de casos de uso de alto nível



Fonte: Próprios autores

3.6 Diagrama de casos de uso baixo nível

Os casos de uso de baixo nível têm como objetivo descrever de forma detalhada como será a utilização de determinada funcionalidade do sistema. Os casos de usos de baixo nível deste projeto podem ser vistos no Apêndice A.

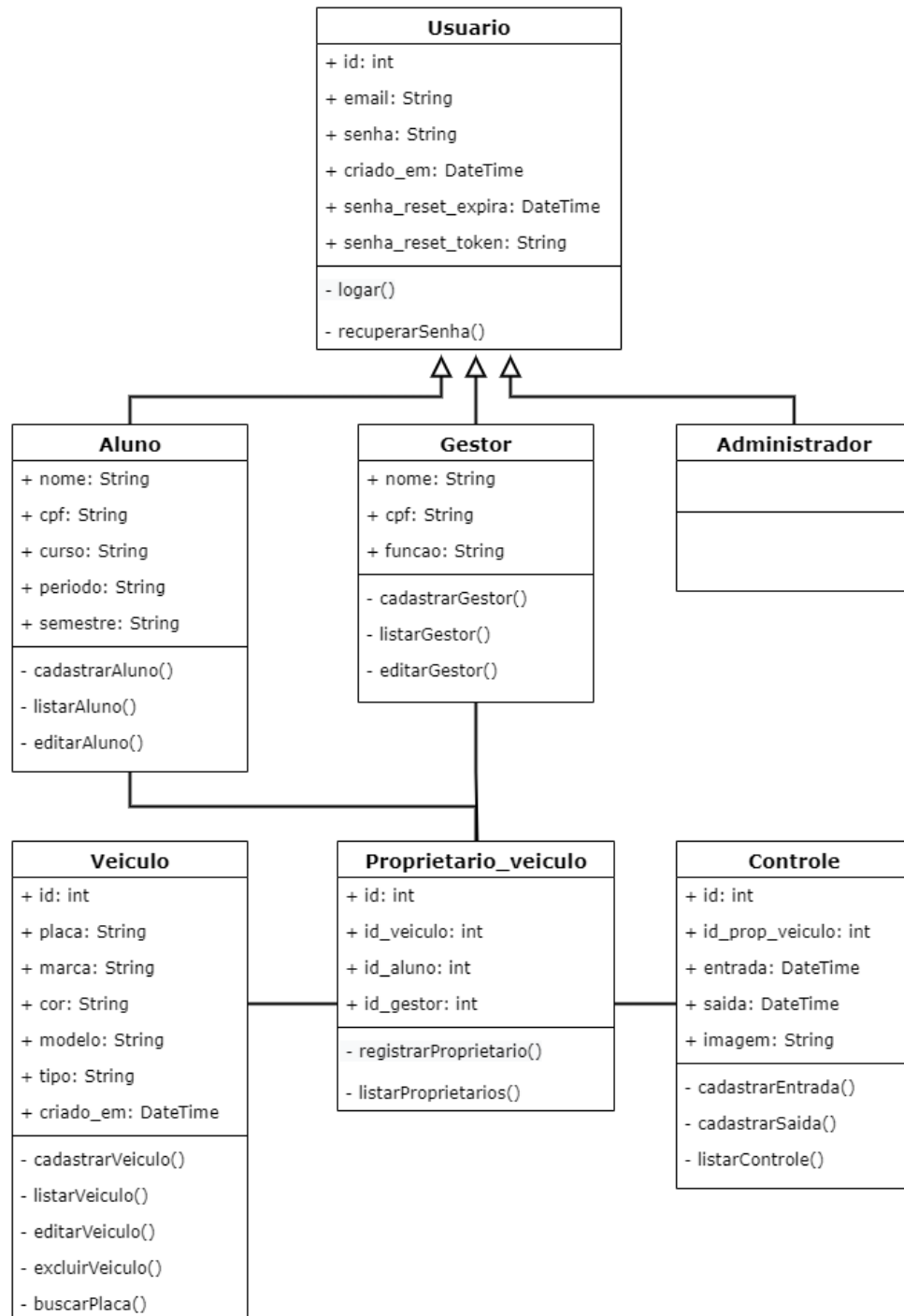
3.7 Diagrama de classes

O objetivo do diagrama de classes é demonstrar as estruturas das classes do sistema junto aos seus atributos e métodos, melhorando assim a visão geral do esquema da aplicação.

Cada classe é um objeto, os atributos são as qualidades/características da classe, já os métodos são as ações daquele determinado objeto.

A Figura 18 é uma representação do diagrama de classes deste presente projeto.

Figura 18 - Diagrama de classes do sistema

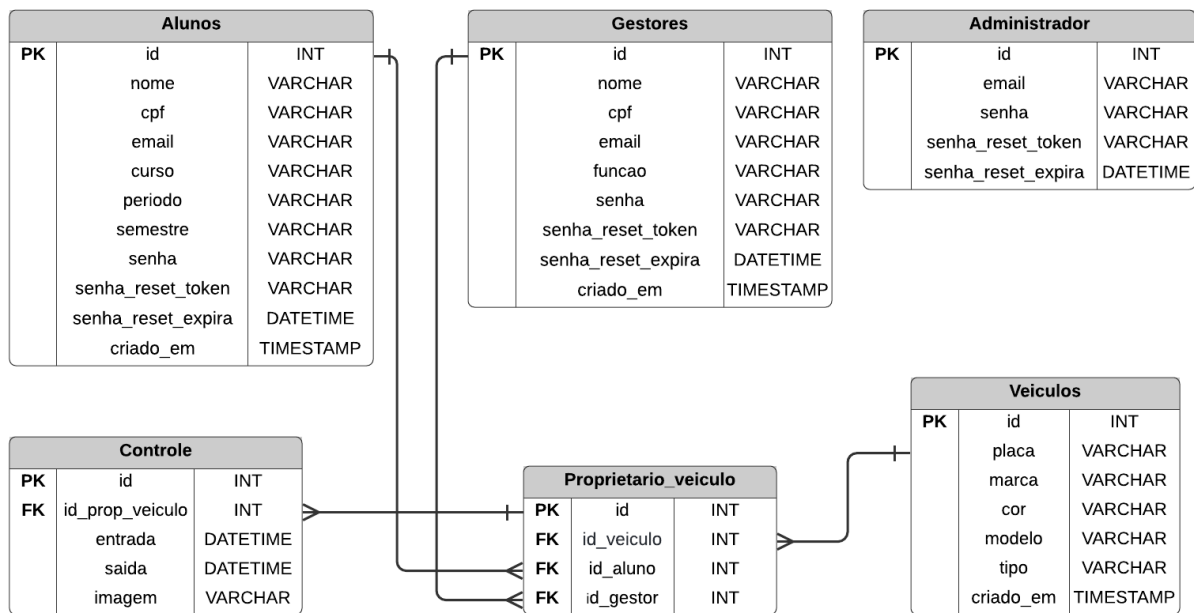


Fonte: Próprios autores

3.8 Diagrama entidade-relacionamento

O diagrama entidade-relacionamento representa os dados da aplicação que serão armazenados na base de dados com sua respectiva relação e tipo de dado ali armazenado. A Figura 19 é a representação do diagrama de banco de dados do sistema.

Figura 19 - Diagrama entidade-relacionamento do sistema



Fonte: Próprios autores

3.9 Dicionário de dados

É um conjunto de definições dos dados que são armazenados nas tabelas do banco de dados do sistema, conforme apresentado no Apêndice B.

CAPÍTULO 4 MODELAGEM DO PROTÓTIPO

Nesse capítulo será apresentado toda a modelagem de telas do sistema Web e os resultados do sistema computacional deste presente projeto.

4.1 Modelagem das telas

Nesse capítulo será apresentado toda a modelagem de telas do sistema Web, parte das modelagens de telas do sistema utilizam de *templates* (Estilizações prontas) encontrados na Internet, como as telas de login (autenticação) e o *dashboard* administrativo, já que o maior foco do nosso projeto é o reconhecimento de placas veiculares, e nosso sistema web é a construção da base de dados de consulta de placas.

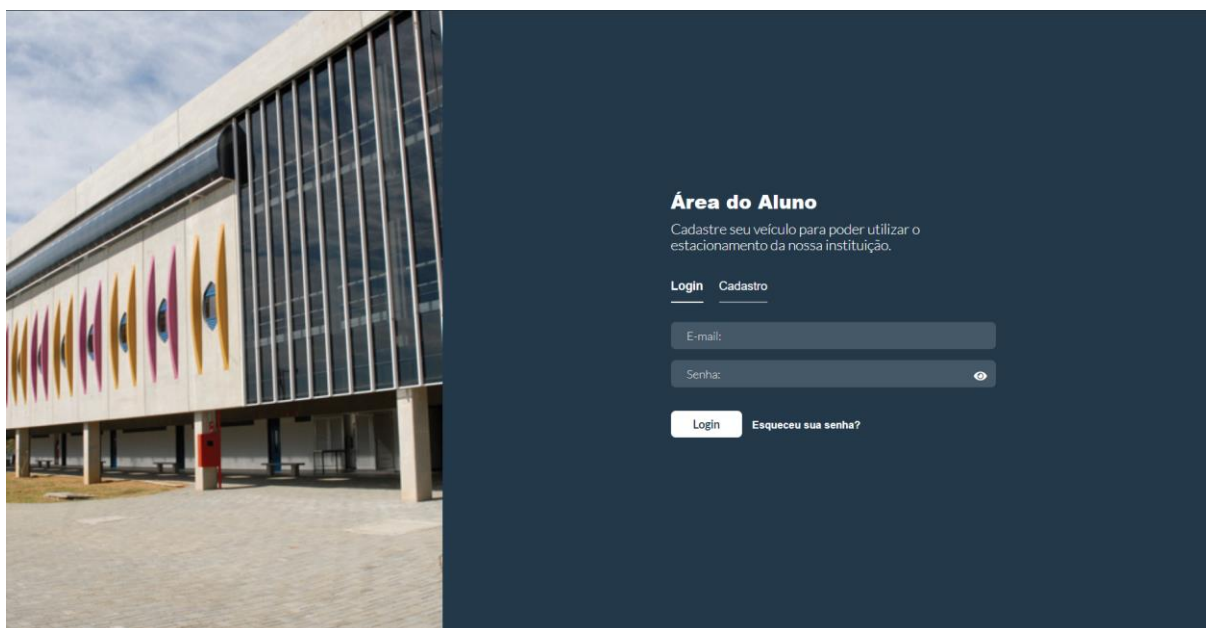
Figura 20 - Tela principal



Fonte: Próprios autores

A tela principal representada pela Figura 20, também conhecida como *landing page*, é a primeira tela que o usuário terá acesso, o intuito da página era remeter a instituição Fatec Itu, foram utilizados os logos e as cores da instituição. Essa tela é a responsável por redirecionar o usuário para sua respectiva tela de autenticação.

Figura 21 – Tela de login do aluno



Fonte: Próprios autores

A Figura 21 remete a tela de login do aluno, onde é necessário informar o e-mail e a senha do respectivo usuário, existe também a opção de “Cadastro”, onde o mesmo poderá efetuar seu cadastro caso ainda não possua um, a tela de cadastro é ilustrada pela Figura 22. A opção “esqueceu sua senha?” deve ser usada quando o usuário não lembrar a sua senha de acesso. Ao usar essa opção o usuário será redirecionado para a tela representada na Figura 23, onde poderá realizar todo o processo de recuperação de senha, somente informando seu E-mail cadastrado, e verificando sua caixa de entrada do e-mail, para assim ter acesso a tela de “restar” senha, ilustrada na Figura 24.

A diferenciação das telas de autenticação do aluno para a do gestor, é somente a cores de fundo, sendo a do gestor nas cores vermelha. Os dados de cadastro também mudam para os dois tipos de usuário, como podemos perceber na Figura 25.

Figura 22 – Tela de cadastro do aluno



Cadastro de Aluno

Cadastre seu veículo para poder utilizar o estacionamento da nossa instituição.

Login Cadastro

Nome:

E-mail:

CPF:

Selecione um curso ...

Selecione um período ...

Selecione um semestre ...

Senha:

Cadastrar

Fonte: Próprios autores

Figura 23 – Tela de recuperação de senha do aluno



Resetar Senha

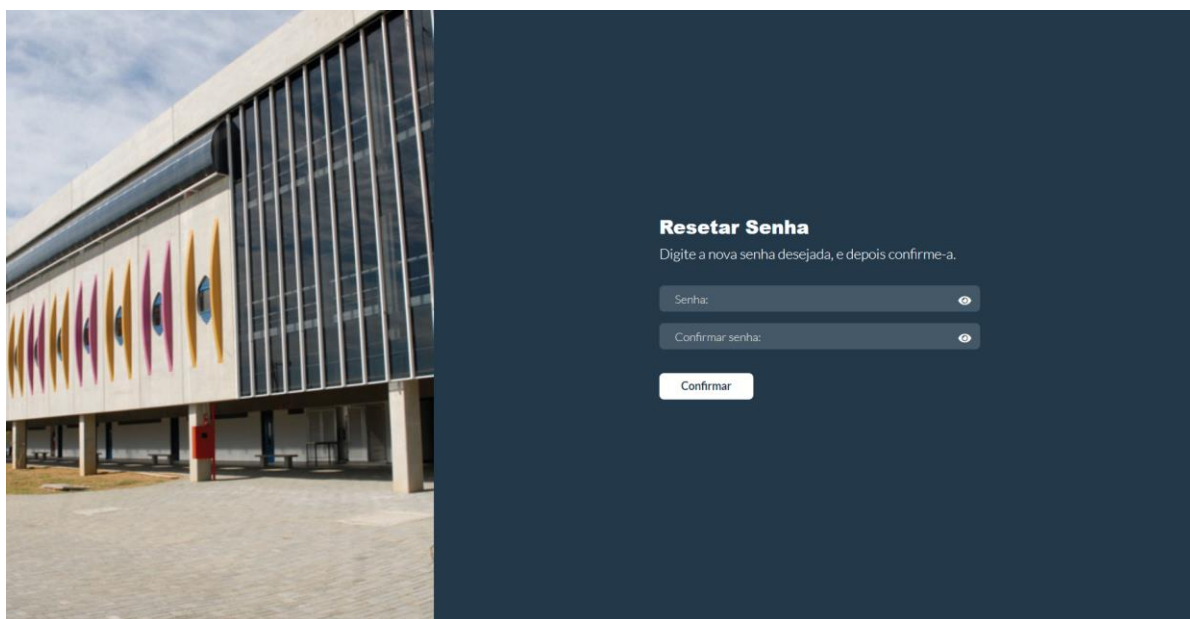
Para redefinir sua senha, digite o endereço de E-mail que você usa para fazer login.

E-mail:

Enviar link voltar

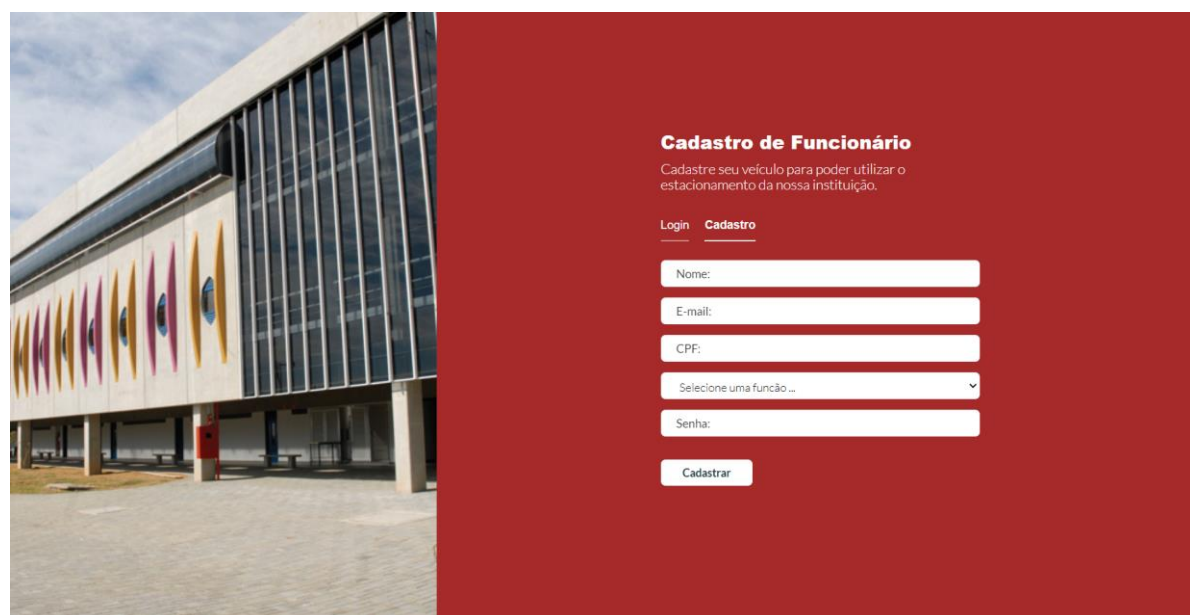
Fonte: Próprios autores

Figura 24 – Tela de reset de senha do aluno



Fonte: Próprios autores

Figura 25 – Tela de cadastro do gestor



Fonte: Próprios autores

Após a autenticação do usuário no sistema, o mesmo será redirecionado para seu respectivo dashboard, onde poderá visualizar e editar seus dados cadastrados. Logo ao lado poderá cadastrar, editar e excluir diversos veículos. Todos os veículos cadastrados ficam em um formato de lista para visualização do usuário. O dashboard do aluno é ilustrado na Figura 26.

Figura 26 – Dashboard do aluno

The screenshot displays the Fatec student dashboard. On the left, a profile card for Bianca Alves Barbosa is visible, containing fields for Name, E-mail, CPF, Course, Semester, and Period. On the right, a modal titled 'Adicionar novo veículo' (Add new vehicle) is open, showing fields for License Plate, Brand, Color, Type, and Model, with a 'Salvar' (Save) button at the bottom.

Fonte: Próprios autores

Para o administrador do sistema Web, existem telas distintas e uma autenticação própria. A tela de autenticação não tem acesso direto pela *Landing Page* como as demais, ela deve ser acessada pela própria URL. A Figura 27 representa a Tela de autenticação do Administrador, ela também possui recuperação de senha com a mesma funcionalidade das demais telas de autenticação do sistema.

Figura 27 – Tela de login do administrador

The screenshot shows the administrator login page. It features the Fatec logo and the text 'Login Bem vindo a área do administrador!'. Below this, there is a login form with fields for 'E-mail' and 'Senha' (Password), an 'Entrar' (Enter) button, and a link for 'Esqueceu sua senha?' (Forgot your password?).

Fonte: Próprios autores

Com o administrador logado, o mesmo é redirecionado para o Dashboard Administrativo, onde ele pode visualizar todos os veículos e seus respectivos proprietários cadastrados na Base de Dados, ou visualizar todos os acessos feitos na instituição.

A Figura 28 ilustra a listagem de veículos, onde é possível visualizar todos os dados do veículo, e clicando no ícone de visualização se pode visualizar os dados do respectivo proprietário. A listagem também conta com um campo para buscar dados pela placa do veículo ou pelo proprietário.

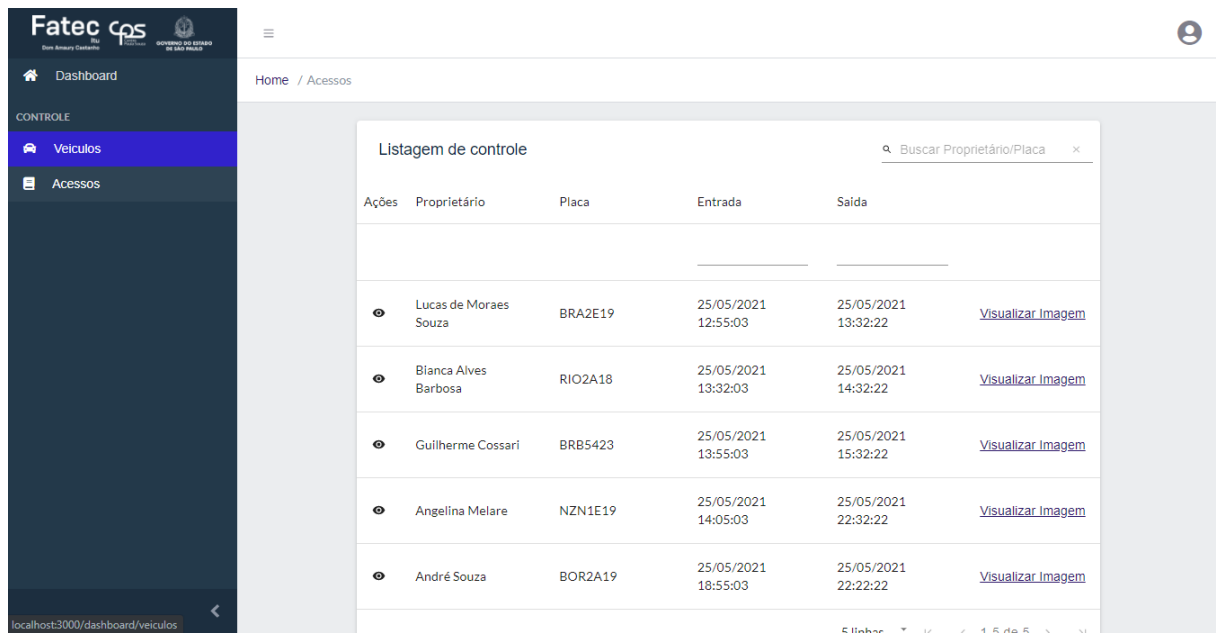
Figura 28 – Listagem dos veículos no Dashboard Administrativo

Ações	Placa	Marca	Modelo	Cor	Proprietário
	BRA2E19	Volkswagen	Fox 2014	Cinza	Lucas de Moraes Souza
	RIO2A18	Volkswagen	Corsa	Preto	Bianca Alves Barbosa
	BRB5423	Hyundai	HB20	Branco	Guilherme Cossari
	NZN1E19	Hyundai	Santa fé	preta	Angelina Melare

Fonte: Próprios autores

A Figura 29 ilustra a listagem de acesso dos veículos, onde é possível visualizar o veículo e a sua hora de entrada e saída da instituição. É possível visualizar clicando no ícone de visualização os dados de forma completa como apresenta a Figura 30, e clicando na opção “Visualizar Imagem” o navegador abre uma nova aba com a imagem armazenada no Cloudinary, do respectivo veículo capturada pela Raspicam. A listagem também possui busca por placa do veículo ou proprietário, e filtragem por data de entrada ou saída.

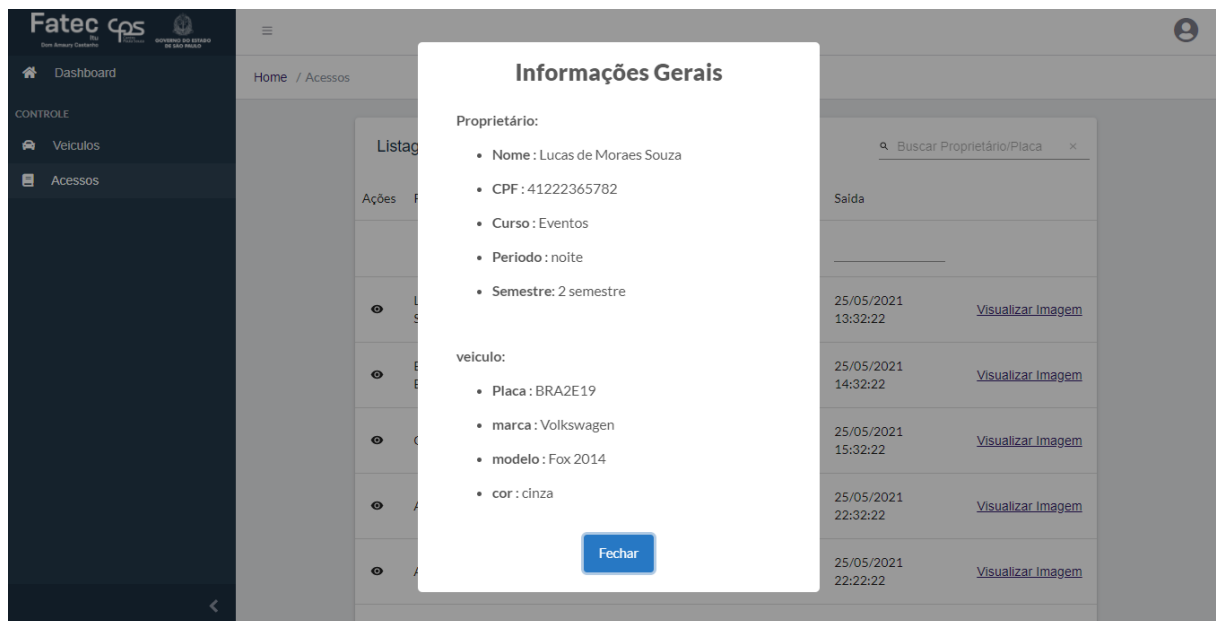
Figura 29 - Listagem dos acessos no Dashboard Administrativo



Ações	Proprietário	Placa	Entrada	Saida
	Lucas de Moraes Souza	BRA2E19	25/05/2021 12:55:03	25/05/2021 13:32:22 Visualizar Imagem
	Blanca Alves Barbosa	RIO2A18	25/05/2021 13:32:03	25/05/2021 14:32:22 Visualizar Imagem
	Guilherme Cossari	BRB5423	25/05/2021 13:55:03	25/05/2021 15:32:22 Visualizar Imagem
	Angelina Melare	NZN1E19	25/05/2021 14:05:03	25/05/2021 22:32:22 Visualizar Imagem
	André Souza	BOR2A19	25/05/2021 18:55:03	25/05/2021 22:22:22 Visualizar Imagem

Fonte: Próprios autores

Figura 30 – Informações Gerais do respectivo acesso



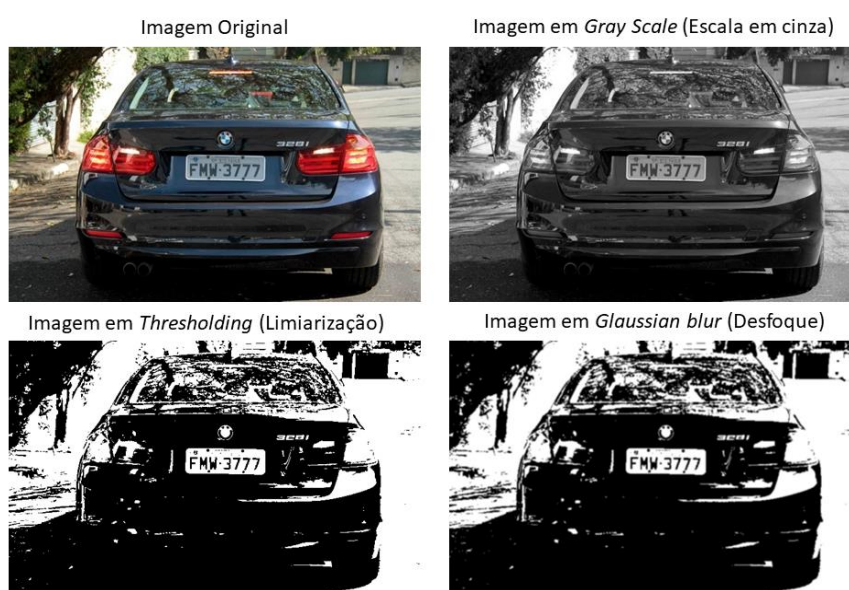
Informações Gerais	
Proprietário:	
• Nome :	Lucas de Moraes Souza
• CPF :	41222365782
• Curso :	Eventos
• Período :	noite
• Semestre :	2 semestre
veículo:	
• Placa :	BRA2E19
• marca :	Volkswagen
• modelo :	Fox 2014
• cor :	cinza
Fechar	

Fonte: Próprios autores

4.2 Resultados obtidos

Para essa etapa foram realizados testes com imagens de diversas placas veiculares, para visualizar o percentual de acertos que o sistema de reconhecimento possui. Todas essas imagens passaram pelas 4 etapas: Pré-processamento, identificação da placa, extração dos caracteres e reconhecimento desses caracteres. A Figura 31 representa o resultado obtido na etapa de pré-processamento.

Figura 31 – Resultado da etapa de pré-processamento do projeto



Fonte: Próprios autores

A Figura 32 representa os contornos obtidos na etapa de identificação da placa, onde foi utilizada o método de segmentação por contornos, todos os contornos encontrados ficam com linhas azuis. A Figura 33 representa a placa identificada pelo algoritmo.

Figura 32 – Contornos encontrados na etapa de identificação da placa



Fonte: Próprios autores

Figura 33 – Placa identificada na etapa de identificação da placa

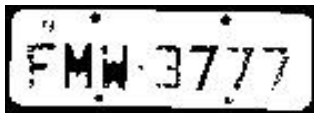


Fonte: Próprios autores

O Quadro 2 apresenta todos os testes realizados com diversas placas veiculares e os respectivos resultados obtidos pelo sistema computacional. A grande maioria das imagens apresentadas foram capturadas com a Raspicam.

Quadro 2 – Resultados obtidos a partir do Sistema Computacional

Índice	Imagem Capturada	Placa identificada	Caracteres obtidos
01			FXR9915
02			BRA3R52
03			(DZF1FU

04			DZF1F64
05			BRA2E19
06			BRA5IL
07			ENW9720
08			FMW3777
09			ATF7354

Fonte: Próprios autores

Pode-se concluir dos testes realizados que 9 das imagens testadas, 6 obtiveram 100% dos caracteres obtidos, algumas tiveram um dos 7 caracteres que compõem a placa trocados, como no caso da imagem de índice 04 do Quadro 2, onde o caractere “0” cortado foi confundido com o caractere “6”.

As placas Mercosul apresentaram maior índice de erros, sendo 5 das placas utilizadas para teste, somente 2 obtiveram 100% dos caracteres. Um dos fatores que influenciaram na extração dos caracteres, foram as informações que contém a placa, como a bandeira, a escrita “BR” no canto inferior esquerdo e a escrita “Brasil” centralizada no topo da placa, trazendo as vezes caracteres como “(” ou até mesmo “BRA5IL”. Para melhorar o reconhecimento em

placas Mercosul, foi utilizado de funções *replaces*⁶ para eliminar caracteres especiais como (, . / + ‘ : °) , isso aumentou a taxa de sucesso.

Para o teste de placas antigas no território brasileiro, todas as imagens obtiveram 100% dos caracteres.

Outros fatores que influenciam na identificação da placa é a iluminação e posicionamento da placa, imagens bem iluminadas obtiveram resultados melhores que placas escuras, assim como placas bem centralizadas.

⁶ *Replace* – Função na programação que substitui um caractere ou palavra por outra.

CONSIDERAÇÕES FINAIS

O desenvolvimento deste presente projeto possibilitou a compreensão da necessidade de uma segurança melhorada para a instituição e seus visitantes, gerando a ideia de um software, e uma respectiva arquitetura, para o acompanhamento de entrada e saída de veículos gerando maior confiança, agilidade e proteção aos envolvidos.

Como mostrado ao longo dos capítulos, foi realizada a modelagem de um sistema computacional e físico que pudesse realizar o processo de reconhecimento de placas e controle de acesso na instituição de ensino de maneira automatizada. Para realizar a modelagem de forma correta foi realizado vários estudos nas áreas de visão computacional e inteligência artificial, incluindo suas ferramentas e métodos.

Com relação aos testes realizados, observou-se uma acurácia superior a 66% de acerto. Analisando os erros, nota-se que pequenas melhorias podem fazer com que esse índice aumente, tais como: melhoria na iluminação, aumento da resolução da imagem etc.

Diante disso, e resgatando os objetivos gerais e específicos inicialmente estipulados, pode-se inferir que foram integralmente realizados. Mesmo assim, muito mais há que se pesquisar sobre a profundidade dos métodos que foram utilizados ou novos métodos para aperfeiçoamento do sistema.

Ficam, então, como sugestões para trabalhos futuros a implementação de melhorias no presente projeto, tais como o aperfeiçoamento das técnicas utilizadas para reconhecimento em placas Mercosul e pesquisa de métodos que tragam melhores resultados, tais como a aplicação de Redes Neurais Convolucionais. E a ampliação para o controle da entrada e saída de pedestres utilizando reconhecimento facial.

REFERÊNCIAS BIBLIOGRÁFICAS

ALFF, Chico. **O que são Requisitos Funcionais e Não Funcionais?** 2018. Disponível em: <https://analisederequisitos.com.br/requisitos-funcionais-e-nao-funcionais/>. Acesso em: 10 nov. 2020.

ALMEIDA, Luiz Eduardo Pita Mercês. **Estudo de caso em Reconhecimento Automático de Placas Veiculares usando linguagem de programação Python.** 2018. 90 f. TCC (Graduação) - Curso de Engenharia Elétrica, Departamento de Engenharia Elétrica, Universidade Federal da Paraíba, João Pessoa, 2018. Disponível em: https://drive.google.com/file/d/1K25LmViw_Mg-k-E4_Um6GuJhJoesrSeE/view. Acesso em: 20 out. 2020.

ALVES FILHO, Sebastião Emidio. **Arquitetura de um cluster computacional de baixo consumo e com proporcionalidade energética.** 2017. 85 f. Tese (Doutorado) - Curso de Engenharia de Computação, Universidade Federal do Rio Grande do Norte, Natal, 2018. Disponível em: https://repositorio.ufrn.br/jspui/bitstream/123456789/24935/1/SebastiaoEmidioAlvesFilho_TESE.pdf. Acesso em: 31 out. 2020.

BARBOSA, Alexandre Henrique C.. **Processamento digital de imagens para o reconhecimento de placas de veículos.** 2017. 25 f. TCC (Graduação) - Curso de Análise de Sistemas, Faculdade de Balsas, Balsas, 2017. Disponível em: <http://www.unibalsas.edu.br/wp-content/uploads/2017/01/Alexandre.pdf>. Acesso em: 07 nov. 2020.

BAÚ DA ELETRÔNICA (org.). **Aprenda a Controlar Motor DC com Módulo Ponte H e NodeMCU.** 2017. Disponível em: <http://blog.baudaeletronica.com.br/driver-motor-ponte-h-l298n-e-nodemcu/>. Acesso em: 01 out. 2020.

BISHOP, Christopher M.. **Pattern Recognition and Machine Learning.** New York: Springer Science+Business Media, Llc, 2006. 736 p. Disponível em: <http://users.isr.ist.utl.pt/~wurmd/Livros/school/Bishop%20Pattern%20Recognition%20And%20Machine%20Learning%20-%20Springer%202006.pdf>. Acesso em: 18 out. 2020.

CAMPOS, Tatiane Jesus de. **Reconhecimento de Caracteres Alfanuméricos de Placas em Imagens de Veículos.** 2001. 120 f. Dissertação (Mestrado) - Curso de Ciência da Computação, Universidade Federal do Rio Grande do Sul, Porto Alegre, 2001. Disponível em: <https://lume.ufrgs.br/handle/10183/2329>. Acesso em: 26 out. 2020.

CHAGAS, Elza Marisa Paiva de Figueiredo; RODRIGUES, Denilson Laudares. **MÉTODO DE SEGMENTAÇÃO DE OBJETOS EM IMAGENS BASEADO EM CONTORNOS ACTIVOS E ALGORITMO GENÉTICO.** Disponível em: <https://web.fe.up.pt/~tavares/downloads/publications/artigos/METNUMp545.pdf>. Acesso em: 28 maio 2021.

CLOUDINARY. **Documentação.** Disponível em: https://cloudinary.com/documentation/solution_overview. Acesso em: 07 jun. 2021.

DIAS, Eng. Fábio Gaiotto. **Melhorias Para Sistemas De Reconhecimento Da Placa De Licenciamento Veicular.** (Dissertação De Mestrado Da Faculdade De Engenharia Elétrica E De Computação). Unicamp. Campinas – Sp, 2005.

ELIAS, Paulo Sá. **Algoritmo, Inteligência Artificial e o Direito**. Disponível em: <https://www.conjur.com.br/dl/algoritmos-inteligencia-artificial.pdf>. Acesso em: 18 out. 2020.

FACON, Jacques. **Processamento e Análise de Imagens**. 2002. 128 f. Dissertação (Mestrado) - Curso de Informática Aplicada, Pontifícia Universidade Católica do Paraná, Curitiba, 2002. Disponível em: <https://webserver2.tecgraf.puc-rio.br/~scuri/inf1378/pub/CursoProcImagem.pdf>. Acesso em: 09 nov. 2020.

GONÇALVES, Flávio Roberto de Freitas. **Visão Computacional aplicada a automação de colhedora multifuncional de hortícolas-Alface**. 2019. 94 f. Tese (Doutorado) - Curso de Engenharia de Sistemas Agrícolas, Universidade Federal do Ceará, Fortaleza, 2019. Disponível em: http://www.repositorio.ufc.br/bitstream/riufc/49255/13/2019_tese_frdfgoncalves.pdf. Acesso em: 07 nov. 2020.

GREENEMEIER, Larry. **Visionary Research. Teaching Computers To see Like a Human**. 2008. Disponível em: <https://www.scientificamerican.com/article/visionary-research/>. Acesso em: 20 out. 2020.

HAYKIN, Simon. **Redes Neurais: princípios e prática**. 2. ed. Porto Alegre: Bookman, 2001. 902 p. Tradução de: Paulo Martins Engel.

HEEKS, Richard; ROBINSON, Andrew. Ultra-low-cost computing and developing countries. **Communications Of The Acm**. Manchester, p. 22-24. 26 fev. 2014.

HURWITZ, Judith; KIRSCH, Daniel. **Machine Learning**. New Jersey: John Wiley & Sons, Inc., 2018. 75 p. Organização: IBM. Disponível em: <https://www.ibm.com/downloads/cas/GB8ZMQZ3>. Acesso em: 20 out. 2020.

INSTITUTO ATLÂNTICO. **Visão Computacional: como essa tecnologia replica a visão humana?** Disponível em: <https://blog.atlantico.com.br/visao-computacional-como-essa-tecnologia-replica-a-visao-humana/>. Acesso em: 07 nov. 2020.

JSON Web Token. Disponível em: <https://jwt.io/>. Acesso em: 11 nov. 2020.

LEE, Matthias (org.). **Pytesseract**. 2020. Disponível em: <https://pypi.org/project/pytesseract/>. Acesso em: 27 out. 2020.

LENON. **Node.js. O que é, como funciona e quais as vantagens**. 2018. Disponível em: <https://www.opus-software.com.br/node-js/>. Acesso em: 13 nov. 2020.

LORENA, A. C.; CARVALHO, A. C. de. **Uma introdução às support vector machines**. Revista de Informática Teórica e Aplicada, v. 14, n. 2, p. 43–67, 2007.

MDN, Colaboradores da. **HTML**. 2019. Disponível em: <https://developer.mozilla.org/pt-BR/docs/Web/HTML>. Acesso em: 10 nov. 2020.

MDN, Colaboradores da. **HTTP**. 2020. Disponível em: <https://developer.mozilla.org/pt-BR/docs/Web/HTTP>. Acesso em: 10 nov. 2020.

MDN, Colaboradores da. **JavaScript**. 2019. Disponível em: https://developer.mozilla.org/pt-BR/docs/Web/JavaScript/About_JavaScript. Acesso em: 10 nov. 2020.

MILANO, Danilo de; HONORATO, Luciano Barrozo. **Visão Computacional**. 2010. Universidade Estadual de Campinas. Disponível em: <https://docplayer.com.br/3058305-Visao-computacional->

danilo-de-milano-luciano-barrozo-honorato-unicamp-universidade-estadual-de-campinas-ft-faculdade-de-tecnologia.html. Acesso em: 19 out. 2020.

MORAIS, Emerson Cordeiro. **Reconhecimento de padrões e redes neurais artificiais em predição de estruturas secundárias de proteínas**. 2010. 148 f. Tese (Doutorado) - Curso de Ciência em Engenharia de Sistemas e Computação, Universidade Federal do Rio de Janeiro, Rio de Janeiro, 2010. Disponível em: <https://www.cos.ufrj.br/uploadfile/1277729485.pdf>. Acesso em: 25 out. 2020.

NAKABAYASHI, Luciana Akemi. **A Contribuição Da Inteligência Artificial (IA) Na Filosofia Da Mente**. 2009. 109 F. Dissertação (Mestrado) - Curso De Tecnologias Da Inteligência E Design Digital, Pontifícia Universidade Católica De São Paulo, São Paulo, 2009. Disponível Em: <Http://Livros01.Livrosgratis.Com.Br/Cp096609.Pdf>. Acesso Em: 18 Out. 2020.

NARDI, Eduardo Amadeo de Carli *et al.* **Estudo de caso em Reconhecimento Automático de Placas Veiculares usando linguagem de programação Python**. 2018. 6 f. Tese (Doutorado) - Curso de Engenharia Elétrica, Universidade Federal da Paraíba, João Pessoa, 2018.

NPM. **Bcrypt**. Disponível em: <https://www.npmjs.com/package/bcrypt>. Acesso em: 10 nov. 2020.

OLIVEIRA, Leonardo Augusto; GONZAGA, Adilson. Localização, Segmentação e Reconhecimento de caracteres em Placas de Automóveis. **Avanços em Visão Computacional**, [s.l.], p.283-302, 18 dez. 2012. Omnipax Editora. DOI: 10.7436/2012.avc.15.

OPENCV. Disponível em: Acesso em: <https://opencv.org/about/> 20 de out. 2020

PAES, Alexandre Santos Lima. **RECONHECIMENTO AUTOMÁTICO DE PLACAS AUTOMOBILÍSTICAS**. 2017. 47 f. TCC (Graduação) - Curso de Engenharia Eletrônica e de Computação da Escola Politécnica, Universidade Federal do Rio de Janeiro, Rio de Janeiro, 2017. Disponível em: <http://monografias.poli.ufrj.br/monografias/monopoli10023622.pdf>. Acesso em: 20 out. 2020.

PEREIRA DE FIGUEIREDO, Felipe. **Introdução à Inteligência Artificial e Machine Learning**, 2020.

PYTHON. Disponível em: Acesso em: <https://www.python.org/> 20 de out. 2020

REACT. **React**. 2020. Disponível em: <https://pt-br.reactjs.org/>. Acesso em: 10 nov. 2020.

REIS, Fábio dos. **Como funciona uma Ponte H: controle direcional de motores dc**. Controle direcional de motores DC. 2017. Disponível em: <http://www.bosontreinamentos.com.br/eletronica/curso-de-eletronica/como-funciona-uma-ponte-h-controle-direcional-de-motores-dc/>. Acesso em: 31 out. 2020.

RODRIGUES, Natielli. **O que é um Arduíno? Para que serve e como funciona?** Disponível em: <http://meuprofessordefisica.com/2018/03/27/o-que-e-um-arduino-para-que-serve-e-como-funciona/>. Acesso em: 26 maio 2021.

RUSSELL, Stuart J.; NORVIG, Peter. **Artificial Intelligence: a modern approach**. New Jersey: Prentice-Hall, 1995. 946 p. Disponível em: <https://www.cin.ufpe.br/~tfl2/artificial-intelligence-modern-approach.9780131038059.25368.pdf>. Acesso em: 07 nov. 2020

SILVA JÚNIOR, Waldemir Sabino da. **Reconhecimento de padrões utilizando filtros de correlação com análise de componentes principais**. 2010. 224 f. Tese (Doutorado) - Curso de Engenharia Elétrica, Universidade Federal do Rio de Janeiro, Rio de Janeiro, 2010. Disponível em: <http://www02.smt.ufrj.br/~eduardo/teses/waldir-junior-doutorado.pdf>. Acesso em: 25 out. 2020.

SOBRAL, João Luís. **Segmentação de Imagem**. Disponível em: <http://gec.di.uminho.pt/lesi/vpc0304/Aula07Segmenta%C3%A7%C3%A3o.pdf>. Acesso em: 28 maio 2021.

SOLDAFRIA. **O QUE É UM ARDUINO, PARA QUE SERVE, COMO FUNCIONA, ONDE COMPRAR?** 2019. Disponível em: https://www.soldafria.com.br/blog/o-que-e-um-arduino-para-que-serve-como-funciona-onde-comprar?gclid=CjwKCAjwqcKFBhAhEiwAfEr7zaKOvUyCPeo1Q9uozr0qdK3iJDvJXL8HOCdnWQ0LAb7J5tNYRofBoCi04QAvD_BwE. Acesso em: 26 maio 2021.

SONAWANE, Vaishnavi. **ROI Segmentation, Contour Detection and Image Thresholding Using OpenCV**. Disponível em: <https://medium.com/swlh/roi-segmentation-contour-detection-and-image-thresholding-using-opencv-c0d2ea47b787>. Acesso em: 28 maio 2021.

SOUZA, Fábio. **Arduino UNO**. 2013. Disponível em: <https://www.embarcados.com.br/arduino-uno/>. Acesso em: 27 maio 2021.

TESSERACT. 2014. WOODS, Richard E.; GONZALEZ, Rafael C. **Digital Image Processing**. 3. ed. Florida: Pearson, 2007. 976 p

THEODORIDIS, Sergios; KOUTROUMBAS, Konstantinos. **Pattern Recognition**. United States Of America: Elsevier Academic Press, 2003. 711 p.

TRENTINI, Vinicius Bergoli *et al.* **Reconhecimento Automático de Placas de Veículos**. 2010. 6 f. Monografia (Especialização) - Curso de Ciência da Computação, Departamento de Computação, Universidade Estadual Paulista - Unesp (Campus de Bauru), Presidente Prudente, 2010. Disponível em: http://iris.sel.eesc.usp.br/wvc/anais_WVC2010/artigos/poster/72819.pdf. Acesso em: 26 out. 2020.

VALLE FILHO, Adhemar Maria do. **Um modelo para implementação de consciência em Robôs Móveis**. 2003. 128 f. Tese (Doutorado) - Curso de Engenharia de Produção, Universidade Federal de Santa Catarina, Florianópolis, 2003. Cap. 4. Disponível em: <https://repositorio.ufsc.br/bitstream/handle/123456789/85265/197729.pdf?sequence=1>. Acesso em: 18 out. 2020.

VON ATZINGEN, Gustavo Voltani. **Simulação, controle e automação de um forno tipo túnel utilizando tecnologia embarcada**. 2017. 140 f. Tese (Doutorado) - Curso de Engenharia de Alimentos, Faculdade de Zootecnia e Engenharia de Alimentos, Pirassununga, 2017. Disponível em: <https://www.teses.usp.br/teses/disponiveis/74/74132/tde-14062017-105834/publico/DO3268241COR.pdf>. Acesso em: 28 out. 2020.

ZHANG, Z., F, *et al.* **A framework for quantifying the impacts of sub-pixel reflectance variance and covariance on cloud optical thickness and effective radius retrievals based on the bi-spectral method** 2017. Auckland, New Zealand, AIP Conference Proceedings, vol. 1810

APÊNDICES

APÊNDICE A – Sequência de diagramas de casos de usos baixo nível

Tabela 5 - Diagrama de casos de uso baixo nível do requisito efetuar login

Caso de Uso	Efetuar login	
Ator Principal	Usuário e Administrador	
Ator Secundário		
Pré-Condição	Conter cadastrado no sistema.	
Pós-Condição	O usuário e/ou administrador será redimensionado para seu dashboard de acordo com seu nível de acesso.	
Ação do Ator		Responsabilidade do Sistema
1- Acessar a sua respectiva tela de login.		
2- Preencher e-mail e senha.		
		3- Verificar se o e-mail e senha constam no banco de dados do tipo de usuário informado.
		4- Se constar o usuário é redirecionado para seu dashboard, se não, exibe mensagem de erro.

Tabela 6 - Diagrama de casos de uso baixo nível do requisito realizar cadastro

Caso de Uso	Manter cadastro	
Ator Principal	Usuário	
Ator Secundário		
Pré-Condição		
Pós-Condição	O usuário será redimensionado para seu dashboard de acordo com seu nível de acesso ou seus dados serão atualizados na base de dados.	
Ação do Ator		Responsabilidade do Sistema
1- Acessar sua respectiva tela de cadastro.		
2- Preencher as informações a serem cadastradas.		
		3- Verificar se o e-mail já existe.
		4- Se o e-mail existir exibir mensagem de erro, se não, inserir os dados na base de dados e redirecionar o

	usuário para seu dashboard.
5- Para atualizar o perfil basta entrar no seu respectivo dashboard e clicar sobre o botão de edição para liberação dos campos.	
	6- Após a confirmação de atualização os dados são atualizados na base de dados

Tabela 7 - Diagrama de casos de uso baixo nível do requisito recuperar senha

Caso de Uso	Recuperar senha	
Ator Principal	Usuário e Administrador.	
Ator Secundário		
Pré-Condição	Conter cadastrado no sistema.	
Pós-Condição	Envio de e-mail para recuperação de senha.	
Ação do Ator		Responsabilidade do Sistema
1- Acessar sua respectiva tela de esqueci minha senha.		
2- Informar e-mail cadastrado.		
		3- Verificar se o e-mail consta no banco de dados do tipo de usuário informado.
		4- Se constar é enviado um e-mail para o usuário contendo um link de redirecionamento a tela de resetar senha, se não, exibe mensagem de erro.
5- Acessa o link de redirecionamento e informa a nova senha desejada.		
		6- A informação de senha do usuário é atualizada na base de dados.

Tabela 8 - Diagrama de casos de uso baixo nível do requisito cadastrar veículo

Caso de Uso	Manter veículo
Ator Principal	Usuário
Ator Secundário	
Pré-Condição	Conter cadastro no sistema e estar autenticado.
Pós-Condição	Armazenamento de um novo registro referente ao veículo.

Ação do Ator	Responsabilidade do Sistema
1- Preencher as informações a serem cadastradas.	
	2- Verifica se a placa informada já existe.
	3- Se a placa existir exibir mensagem de erro, se não, inserir os dados na base de dados.
4- Para edição dos dados do veículo, basta clicar no botão de edição do respectivo veículo listado no dashboard.	
	5- Após a confirmação de atualização os dados são atualizados na base de dados
6- Para a exclusão do veículo, basta clicar no botão de exclusão do respectivo veículo listado no dashboard.	
	7- Após a confirmação de exclusão os dados do veículo serão excluídos da base de dados.

Tabela 9 - Diagrama de casos de uso baixo nível do requisito listar controle

Caso de Uso	Listar controle	
Ator Principal	Administrador	
Ator Secundário		
Pré-Condição	Estar autenticado no sistema.	
Pós-Condição	Acesso aos dados do controle.	
Ação do Ator	Responsabilidade do Sistema	
1- Escolher a opção de listagem de controle.		
	2-	Realizar a listagem do controle dos veículos.
3- Realizar filtragens por data, mês e ano e ordenar os dados.		

Tabela 10 - Diagrama de casos de uso baixo nível do requisito captar vídeo

Caso de Uso	Captar imagem	
Ator Principal	Sistema físico.	
Ator Secundário		
Pré-Condição	Receber a ação do clique do botão.	
Pós-Condição	Reconhecimento da placa.	
Ação do Ator		Responsabilidade do Sistema
1- O botão quando clicado efetua a captura de imagem do veículo.		
		2- A imagem é captada pela Raspicam.
3- Toda a imagem capturada pela Raspicam é enviada para o Sistema computacional		

Tabela 11 - Diagrama de casos de uso baixo nível do requisito reconhecer placa

Caso de Uso	Reconhecer placa	
Ator Principal	Sistema computacional	
Ator Secundário		
Pré-Condição	Entrada da imagem veicular.	
Pós-Condição	Consultar placa.	
Ação do Ator		Responsabilidade do Sistema
1- Utiliza-se a entrada da imagem da Raspicam.		
2- Realiza-se o pré-processamento da imagem aplicando todos os filtros com a biblioteca OpenCV.		
3- Realiza a identificação da placa para separá-la do restante da imagem.		
4- Realiza-se a extração dos caracteres que contém a placa.		
5- Realiza-se o reconhecimento desses caracteres extraídos com a biblioteca Tesseract.		
6- Com o reconhecimento sucedido realiza-se a consulta da placa.		

Tabela 12 - Diagrama de casos de uso baixo nível do requisito consultar placa

Caso de Uso	Consultar placa	
Ator Principal	Sistema computacional	
Ator Secundário		
Pré-Condição	Entrada da <i>string</i> de caracteres reconhecidos da placa	
Pós-Condição	Registro de entrada ou Registro de saída	
Ação do Ator		Responsabilidade do Sistema
1- Utiliza-se a entrada de uma <i>string</i> de caracteres		
2- Realiza-se a consulta na base de dados dos veículos. Se houver a placa cadastrada realiza o registro de entrada ou saída e o envio do dado de liberação para o Arduino. Se não, tem que haver a liberação manual e o envio do dado de negação para o Arduino		

Tabela 13 - Diagrama de casos de uso baixo nível do requisito registrar controle

Caso de Uso	Registrar controle	
Ator Principal	Sistema computacional	
Ator Secundário		
Pré-Condição	Realização da consulta da placa.	
Pós-Condição	Liberação da cancela.	
Ação do Ator		Responsabilidade do Sistema
1- Entrada dos dados do veículo e do usuário encontrados pela consulta da placa.		
2- Se o botão pressionado pelo veículo for de entrada, se registra a entrada do veículo junto aos dados acima encontrados e a imagem captada do veículo.		
3- Se o botão pressionado pelo veículo for de saída, se registra a saída do veículo junto aos dados do 1 passo encontrados.		
		4- Inserir os dados na base de dados.

Tabela 14 - Diagrama de casos de uso baixo nível do requisito acionar cancela

Caso de Uso	Acionar sinalização	
Ator Principal	Sistema físico	
Ator Secundário		
Pré-Condição	Realização do registro de controle.	
Pós-Condição		
Ação do Ator		Responsabilidade do Sistema
1- Registro do controle realizado com sucesso.		
2- O Arduino recebe um caractere da Raspberry especificando a permissão ou negação de acesso do veículo.		
3- De acordo com o caractere recebido é acendido um led diferente. Verde para permissão e vermelho negação.		

APÊNDICE B – DICIONÁRIO DE DADOS

Tabela 15 - Dicionário de dados do diagrama entidade relacionamento

Nome	Tipo de dados	Descrição
id	INT	Valor inteiro que representa o código único de um registro na tabela. Todos os ids encontrados nesse diagrama emitem mesma função.
nome	VARCHAR	Campo de texto que representa o nome do usuário, sendo ele aluno ou gestor.
cpf	VARCHAR	Campo texto que representa o código do documento do usuário, sendo ele aluno, gestor ou visitante.
email	VARCHAR	Campo de texto que representa um e-mail valido de acesso do usuário, sendo ele aluno, gestor ou administrador.
curso	VARCHAR	Campo de texto que representa o curso no qual o usuário aluno cursa na instituição. Ex.: Gestão Empresarial, Eventos etc.

periodo	VARCHAR	Campo de texto que representa o período que o usuário aluno cursa na instituição. Ex.: manhã, tarde, noite.
semestre	VARCHAR	Campo de texto que representa o semestre que o usuário aluno cursa na instituição. Ex.: 1 semestre, 2 semestre etc.
senha	VARCHAR	Campo de texto que representa a senha de acesso do usuário, sendo ele aluno, gestor ou administrador.
funcao	VARCHAR	Campo de texto que representa a função que o usuário gestor tem dentro da instituição. Ex.: professor, coordenador etc.
senha_reset_token	VARCHAR	Campo de texto que representa o token para realizar a mudança de senha do usuário, sendo ele aluno, gestor ou administrador.
senha_reset_expira	DATETIME	Campo de data e hora que representa até o dia e hora que o usuário, sendo ele aluno, gestor ou administrador, pode fazer a mudança de senha.
criado_em	DATETIME	Campo de data e hora que representa o instante único que o registro foi inserido na base de dados.
placa	VARCHAR	Campo de texto que representa os caracteres da placa veicular que identifica o veículo.
marca	VARCHAR	Campo de texto que representa a marca do veículo. Ex.: Ford, Fiat etc.
cor	VARCHAR	Campo de texto que representa a cor do veículo. Ex.: preto, prata etc.
modelo	VARCHAR	Campo de texto que representa o modelo do veículo. Ex.: Palio, Celta etc.
tipo	VARCHAR	Campo de texto que representa qual o tipo de veículo. Ex.: carro, moto etc.
id_veiculo	INT	Valor inteiro que representa o código único de um veículo nos registros, a qual pertence determinado usuário.
id_aluno	INT	Valor inteiro que representa o código único de um

		aluno nos registros, a qual pertence determinado veículo.
id_gestor	INT	Valor inteiro que representa o código único de um gestor nos registros, a qual pertence determinado veículo.
id_prop_veiculo	INT	Valor inteiro que representa o código único de um registro que aponta o veículo a seu respectivo proprietário.
entrada	DATETIME	Campo de data e hora que representa a o momento que o veículo entrou na instituição.
saida	DATETIME	Campo de data e hora que representa a o momento que o veículo saiu da instituição.
imagem	VARCHAR	Campo de texto que representa o caminho relativo da imagem do veículo capturado que se encontra salvo no servidor.