



**Fundação Educacional do Município de Assis  
Instituto Municipal de Ensino Superior de Assis  
Campus "José Santilli Sobrinho"**

**MATHEUS GUSTAVO SANTOS DA SILVA**

**PROTÓTIPO DE ROBÓTICA E IOT PARA CADEIRA DE RODAS  
INTELIGENTE**

**Assis/SP  
2022**



**Fundação Educacional do Município de Assis  
Instituto Municipal de Ensino Superior de Assis  
Campus "José Santilli Sobrinho"**

**MATHEUS GUSTAVO SANTOS DA SILVA**

**PROTÓTIPO DE ROBÓTICA E IOT PARA CADEIRA DE RODAS  
INTELIGENTE**

Trabalho de conclusão de curso de Ciência da Computação, do Instituto Municipal de Ensino Superior de Assis – IMESA e da Fundação Educacional do Município de Assis – FEMA, como requisito parcial à obtenção do Certificado de Conclusão.

**Orientando(a): Matheus Gustavo Santos da Silva  
Orientador(a): Me. Douglas Sanches da Cunha**

**Assis/SP**  
**2022**

### Dados Internacionais de Catalogação na Publicação (CIP)

S586p Silva, Matheus Gustavo Santos da.

Protótipo de robótica e IOT para cadeira de rodas inteligente /  
Matheus Gustavo Santos da Silva – Assis, SP: FEMA, 2022.

54 f.

Trabalho de Conclusão de Curso (Graduação) – Fundação  
Educativa do Município de Assis – FEMA, curso de Ciência da  
Computação, Assis, 2022.

Orientador: Prof. M.<sup>e</sup> Douglas Sanches da Cunha.

1. Cadeira de Rodas. 2. IOT. 3. Inteligência Artificial. 4.  
Acessibilidade. 5. Deficiência Física. I. Título.

CDD 629.892

Biblioteca da FEMA

Ficha catalográfica elaborada de acordo com os dados fornecidos pelo(a)  
autor(a).

# PROTÓTIPO DE ROBÓTICA E IOT PARA CADEIRA DE RODAS INTELIGENTE

MATHEUS GUSTAVO SANTOS DA SILVA

Trabalho de Conclusão de Curso apresentado ao Instituto Municipal de Ensino Superior de Assis, como requisito do Curso de Graduação Ciência da Computação, avaliado pela seguinte comissão examinadora:

**Orientador:** \_\_\_\_\_  
Me. Douglas Sanches Cunha

**Examinador:** \_\_\_\_\_  
Dr. Luiz Ricardo Begosso

## **AGRADECIMENTOS**

Agradeço a Deus pela oportunidade de realizar meus projetos e sonhos, também pela sustentação que ele tem me dado e aconchego nos momentos difíceis da minha vida.

Agradeço a minha família por me apoiar, inspirar, por investir sua confiança e ter possibilitado o término deste trabalho e o desenvolvimento do meu curso. Devo meus agradecimentos principalmente para minha vó Maria de Fátima dos Santos que sempre foi minha parceira na vida e sempre me aconselhou em minhas decisões, e minha mãe Cristiane dos Santos que sempre moveu seus esforços ao meu favor e investiu o pouco que tinha em minha educação.

Agradeço meu vô José Ribeiro dos Santos Neto, a pessoa em que me espelho, que me deu exemplo moral, exemplo de perseverança e me ensinou a buscar meus sonhos de forma honesta e com alegria. Agradeço a meu vô por confiar no meu compromisso de concluir esse curso e pelos dias que passávamos juntos falando de como a natureza de coisas do dia a dia pode ser interessante, quem sabe daí vem meu interesse pela tecnologia.

Agradeço minha namorada Laura Camargo Zibordi por ter dado nova cor a meus dias e memórias, por ter me feito companhia em momentos difíceis e por ter me dado motivação para o desenvolvimento deste trabalho.

Agradeço meu amigo Lucas Gabriel de Assis Lima, pela companhia e parceria no decorrer do curso e na vida, agradeço pelo apoio e pela inspiração de sempre avançar e dar o melhor de mim.

Agradeço meus professores pela instrução e capacitação.

## RESUMO

Conforme a tecnologia avança, a relação entre o ser humano e máquinas cresce, pois, as máquinas facilitam o dia a dia realizando tarefas que não podem ser feitas por humanos ou que as elas realizam com mais eficiência. Levando em conta aplicações da tecnologia na vida do ser humano é possível ver uma relação quase que simbiótica a cada dia. O presente projeto tem como objetivo a confecção de um protótipo de cadeira de rodas inteligente, controlada pelo reconhecimento de expressões faciais utilizando de técnicas de IoT, robótica e reconhecimento facial.

**Palavras-chave:** Cadeira de Rodas; Internet das coisas; Inteligência Artificial; Acessibilidade; Deficiência Física.

## **ABSTRACT**

As technology advances, the relationship between humans and machines grows, as machines facilitate their daily lives by performing tasks that cannot be done by humans or that they perform more efficiently. Taking into account applications of technology in human life it's possible to see an almost symbiotic relationship every day. This project aims to build a prototype of an intelligent wheelchair, controlled by the recognition of facial expressions using IoT, robotic and facial recognition techniques.

**Keywords:** Wheelchair; Internet of Things; Artificial Intelligence; Accessibility; Physical Disability.



## LISTA DE ILUSTRAÇÕES

Figura 1 - Exemplo de código Dart. ....	21
Figura 2 - Declaração de variáveis. ....	22
Figura 3 - Estrutura de condições. ....	22
Figura 4 - Laços de repetição. ....	23
Figura 5 - Exemplo de função. ....	23
Figura 6 - Janela de busca aberta em imagem à procura de um objeto. ....	27
Figura 7 - Haar-Features. ....	27
Figura 8 - Haar-Feature aplicada à imagem. ....	28
Figura 9 – Cálculo do peso. ....	28
Figura 10 - Classificador binário do algoritmo de Viola-Jones. ....	29
Figura 11 - Raspberry Pi. ....	30
Figura 12 - Diagrama de fluxo do projeto. ....	31
Figura 13 – Dependencias utilizadas no presente trabalho. ....	32
Figura 14 – Estrutura padrão projeto Flutter. ....	32
Figura 15 – Conexão Processo principal – Isolate. ....	34
Figura 16 - App desligado. ....	35
Figura 17 - App com comando para direita. ....	36
Figura 18 – Organização do acervo de imagens. ....	39
Figura 19 - Acervo de imagens (Comando Ligar). ....	39
Figura 20 - Bateria Li-ion 18650 Sanyo 12v 4400mah. ....	40
Figura 21 - Regulador De Tensão Ajustável Lm2596 Dc-dc Step Down. ....	41
Figura 22 - Micro Motor Redução N20 12v 100rpm. ....	41
Figura 23 - Driver Motor Ponte H L298n. ....	42
Figura 24 - Protótipo pré-montado. ....	42

Figura 25 - Pré-montagem eletrônica. ....	43
Figura 26 - Protótipo em perspectiva. ....	44
Figura 27 - Protótipo visão frontal. ....	45
Figura 28 - Protótipo visão das conexões. ....	45

## LISTA DE TABELAS

Tabela 1 - Materiais e custos.....	40
------------------------------------	----

## **LISTA DE ABREVIATURAS E SIGLAS**

AGI – Inteligência Artificial Geral

ANI – Inteligência Artificial Limitada

ASI – Superinteligência

IOT – Internet of Things (Internet das Coisas)

## SUMÁRIO

<b>1. INTRODUÇÃO</b>	<b>14</b>
1.1 OBJETIVOS	15
1.2 JUSTIFICATIVA	15
1.3 MOTIVAÇÃO	15
1.4 PERSPECTIVA E CONTRIBUIÇÃO	15
1.5 METODOLOGIA	16
<b>2 REVISÃO BIBLIOGRÁFICA</b>	<b>17</b>
2.1 INTERNET DAS COISAS (IOT)	17
2.2 INTELIGÊNCIA ARTIFICIAL	18
<b>2.2.1 Inteligência Artificial Limita</b>	<b>19</b>
<b>2.2.2 Inteligência Artificial Geral</b>	<b>19</b>
<b>2.2.3 Superinteligência Artificial</b>	<b>19</b>
2.3 TETRAPLEGIA	20
<b>3 TECNOLOGIAS PARA O DESENVOLVIMENTO</b>	<b>21</b>
3.1 DART	21
3.2 FLUTTER	23
3.3 SOCKETS	24
<b>3.3.1 TCP/IP</b>	<b>24</b>
<b>3.3.2 Sockets</b>	<b>25</b>
3.4 WEBSOCKET	25
3.5 PYTHON	26
3.6 ALGORITMO DE VIOLA-JONES	27
3.7 RASPBERRY PI	29
<b>4 DESENVOLVIMENTO DO PROJETO</b>	<b>31</b>
4.1 DESENVOLVIMENTO DO APLICATIVO MÓVEL	31
4.2 SERVIÇO SOCKET E RECONHECIMENTO FACIAL	36
<b>4.2.1 Serviço socket</b>	<b>36</b>
<b>4.2.2 Reconhecimento facial</b>	<b>38</b>
4.3 CONFECÇÃO E PROGRAMAÇÃO DO PROTÓTIPO DE CADEIRA DE RODAS	40
<b>5 CONSIDERAÇÕES FINAIS</b>	<b>46</b>

5.1 TRABALHOS FUTUROS	46
<b>REFERÊNCIAS</b>	<b>48</b>
<b>6 APÊNDICE</b>	<b>51</b>
	<b>51</b>
	<b>53</b>

## 1. INTRODUÇÃO

Inteligências artificiais são soluções tecnológicas capazes de simular comportamentos similares a inteligência humana em máquinas, sendo capazes de tomar decisões de maneira autônoma, reconhecer padrões e até mesmo aprender sobre as escolhas realizadas. Dentre as tecnologias de IA pode-se incluir o reconhecimento facial que é um sistema desenvolvido para identificar pessoas, ou movimentos, por meio de fotos ou vídeos (TOTVS, 2019).

A tecnologia do reconhecimento facial já existe há muito tempo, porém com o surgimento de novas tecnologias, como os smartphones, ela se tornou mais útil e acessível, pois a capacidade de processamento dos celulares atuais dá a possibilidade de o aparelho processar várias imagens, e assim atualmente existem aplicativos que não só reconhecem uma face, como também a modifica (AMAZON, 2020).

No reconhecimento facial, estratégias de classificação são utilizadas para diferenciar características, tonalidades e contraste em uma imagem. Dentre essas estratégias pode-se destacar o algoritmo de Viola-Jones que por meio de classificação consegue identificar objetos, embora ele seja muito utilizado para encontrar componentes faciais. Redes Convolucionais também são muito utilizadas no processamento de imagem, e assim como o algoritmo de Viola-Jones, elas são capazes de fazer classificação de objetos em uma imagem. O mais interessante das redes convolucionais é que elas não precisam de um pré-processamento muito extenso e são capazes de aprender a classificar objetos sozinhas.

As aplicações, tanto das redes convolucionais quanto o algoritmo de Viola-Jones, são inúmeras, e podem ser aplicadas em qualquer área da ciência, como usada por Martins et al. (2012) na identificação de ideogramas japoneses ou como Cordeiro (2019) que também usa tecnologias de classificação em textos. Metodologicamente, este trabalho visa o estudo e desenvolvimento de um protótipo de uma solução em forma de cadeira de rodas utilizando de conceitos de inteligência artificial e robótica para auxiliar a locomoção e a monitoração de pessoas portadoras da tetraplegia, o que poderia trazer a uma pessoa afetada por tetraplegia a experiência de se locomover sem a necessidade de um acompanhante, o que poderia reduzir a transição da pessoa entre cadeira de rodas normal e a cama bem como evitar ou ao menos reduzir dores no processo de troca de posição e aumentar a privacidade e a individualidade.

## 1.1 OBJETIVOS

Este projeto tem como objetivo o desenvolvimento de um robô, representando uma cadeira de rodas, controlado por reconhecimento facial, utilizando de recursos da linguagem de programação Python e do algoritmo de classificação de Viola-Jones. As imagens devem ser capturadas de um dispositivo móvel, e processadas para controlar o robô.

## 1.2 JUSTIFICATIVA

Deficiências como a paraplegia e tetraplegia são a principal motivação a realização deste trabalho para auxiliar o deslocamento de pessoas com tais problemas, visando maior independência, privacidade e um melhor estilo de vida. Também o presente trabalho tem como objetivo contribuir com pesquisas de desenvolvimento de cadeiras de rodas autônomas e fomentar mais trabalhos relacionados.

## 1.3 MOTIVAÇÃO

O presente projeto tem como motivação o lamentável estado físico e principalmente psicológico de pessoas atingidas pela tetraplegia, desse modo espera-se que o desenvolvimento de uma cadeira de rodas capaz de ser deslocada sem o auxílio de um segundo indivíduo além do utente possa contribuir com a privacidade, e saúde emocional de pessoas em condição de tetraplegia.

## 1.4 PERSPECTIVA E CONTRIBUIÇÃO

Pesquisas relacionadas a soluções em IOT para o deslocamento de pessoas e também pesquisas para auxiliar a qualidade de vida de indivíduos deficientes físicos já vem a muito tempo sendo desenvolvidas, com o aumento da capacidade de processamento de dispositivos móveis e também a existência frameworks de inteligência artificial a possibilidade de novos projetos e estudos relacionados ao deslocamento de pessoas com deficiência aumenta. O presente trabalho visa combinar robótica IOT e inteligência artificial para desenvolver uma solução de cadeira de rodas inteligente e espera-se que o desenvolvimento do protótipo possa contribuir e fomentar futuros projetos que combine as áreas da mecânica da saúde e da tecnologia.



## 1.5 METODOLOGIA

Para alcançar os objetivos estabelecidos neste projeto, inicialmente será realizada uma revisão bibliográfica objetivando explorar os assuntos abordados. Serão adotados artigos científicos, monografias, dissertações, teses, livros, resumos e artigos de periódicos e páginas Web de conteúdo confiável e seguro. Após essa revisão, serão exploradas as ferramentas descritas anteriormente, para que haja um maior entendimento em como elaborar as análises dos dados com as ferramentas e formas de visualização dos resultados obtidos. Serão estudadas a utilização da linguagem Python com suas bibliotecas de Inteligência Artificial. Por fim, os objetos relacionados a este projeto, como ferramentas e tecnologias serão relatados em forma de artigos científicos e materiais de apoio, para fornecimento de uma base fundamental.

## 2 REVISÃO BIBLIOGRÁFICA

### 2.1 INTERNET DAS COISAS (IOT)

Internet das coisas (*do inglês Internet of Things (IOT)*) teve sua ascensão com os avanços das áreas de sistemas embarcados, microeletrônica, comunicação e sensoriamento. Esse conceito (IOT) tem como objetivo utilização da internet atual conectada em dispositivos do dia a dia, da indústria, da saúde ou outros, para obter dados fornecidos por esses dispositivos ou controlá-los de forma remota (SANTOS et al., 2016).

IOT abre o caminho para as aplicações em cidades inteligentes, como o controle do tráfego de veículos, cuidados médicos inteligentes baseados no resultado da análise de dados, a produção na indústria pode ser ampliada por meio da previsão do funcionamento de máquinas e medidores inteligentes, e casas inteligentes com seus diversos dispositivos domésticos conectados (TRIPATHY; ANURADHA, 2018). Tripathy e Anuradha (2018) também comentam, “prevê-se que a rede atual de dispositivos conectados tem aproximadamente 27 bilhões de dispositivos e está aumentando”.

De fato, IOT já está fortemente presente na vida das pessoas, a coleta de dados através de smartphones faz com que empresas possam conhecer os interesses de um indivíduo e direcionar de forma precisa a divulgações de serviços e produtos de acordo com o que as pessoas têm interesse. Na agricultura, são utilizados drones e câmeras para maximizar o plantio e a saúde da safra, localizar inconsistência e até mesmo identificar pragas, também existem sistemas de irrigação que são ativados automaticamente de acordo com o clima solo e ambiente. IOT favorece a rastreabilidade de matéria prima facilitando a identificação de fatores degradantes e respectivas ações a serem tomadas, aumentando assim cada vez mais a qualidade de produtos.

Com o aumento da conectividade de dispositivos à serviços ou recursos da WEB também existe o aumento da vulnerabilidade à ataques. A alta velocidade de evolução da Internet das Coisas faz com que fabricantes de dispositivos tenham dificuldades em manter a qualidade da segurança ao adicionar funcionalidades de IP, conectividade, e credenciamento de usuários (FUKUDA, 2019). Essa problemática se dá pelas novas formas de conexão à dispositivos que efetuam a coleta e recebimento de dados (FUKUDA, 2019).

O presente trabalho tem a intenção de aplicar IOT no desenvolvimento do protótipo de cadeira de rodas inteligente. O protótipo poderá se conectar a um servidor que processará imagens obtidas de um smartphone, a função do servidor é saber qual é o comando enviado pelo usuário e direcionar ao computador da cadeira que movimentará os motores.

## 2.2 INTELIGÊNCIA ARTIFICIAL

“Proponho considerar o questionamento, ‘Podem as máquinas pensar?’”, assim Turing (1950) inicia seu artigo “Computing machinery and intelligence”.

Esse questionamento de se as máquinas podem pensar vem se desdobrando até os dias de hoje. Por milhares de anos a humanidade procurou entender como é possível pensar, como é possível perceber, compreender, prever e manipular o mundo ao nosso redor, que é muito maior e muito mais complexo que nós mesmos, mas na questão de IA não é só intrigante entender como ela pode ou deve funcionar, mas também a possibilidade de se criar entidades inteligentes (RUSSELL; NOVIRG, 2013).

Ferreira (1995), define inteligência como:

- Faculdade de aprender, apreender ou compreender; percepção, apreensão, intelecto, intelectualidade;
- Qualidade ou capacidade de compreender e adaptar-se facilmente; capacidade, penetração, agudeza, perspicácia;
- Maneira de entender ou interpretar; interpretação;
- Acordo harmonia, entendimento recíproco;
- Relações ou entendimentos secretos; conluio, maquinação, trama;
- Destreza mental; habilidade;
- Pessoa inteligente.

Pode-se entender inteligência artificial como a capacidade de máquinas agirem como seres humanos em algumas situações em que há a necessidade de analisar ambientes, aprender perceber e decidir. IA se sustenta na junção de big data, computação em nuvem, modelos de dados (GABRIEL, 2019).

As pesquisas e avanços em IA são bem recentes, iniciados após a Segunda Guerra Mundial e é um campo aberto a ser explorado de diferentes formas para solucionar os mais diversos problemas. Russell e Novirg (2013) citam oito definições para inteligência artificial, uma pode ser destacada para o desenvolvimento do presente projeto, “[Automatização de]

atividades que associamos ao pensamento humano, atividades como a tomada de decisões, a resolução de problemas, o aprendizado...” (BELLMAN, 1978), uma vez que para o funcionamento do protótipo de cadeira de rodas inteligente há a necessidade de algoritmos de detecção de faces e de aprendizado de padrões a fim de classificar imagens captadas do usuário com base em um acervo de imagens previamente montado.

### **2.2.1 Inteligência Artificial Limita**

Inteligências Artificiais Limitadas (ANI) são inteligências que são programadas de forma direcionada para resolver problemas de uma única área, essas inteligências tem como característica uma grande quantidade de dados de um assunto específico e um rápido processamento. “É o tipo de inteligência artificial que consegue vencer um campeão de xadrez, mas é a única coisa que ela faz.” Diz: Gabriel (2019). Esse tipo de inteligência artificial é o tipo que se tem hoje, e aplica-se no reconhecimento facial no desenvolvimento de carros autônomos entre outras infinitas possibilidades. Esse tipo de IA que utiliza vários métodos e algoritmos como Redes Neurais, Deep Learning, Classificação etc. (GABRIEL, 2019).

### **2.2.2 Inteligência Artificial Geral**

Inteligências Artificiais Gerais (AGI) são inteligências artificiais com capacidade lógica e de aprendizado equivalente ou próxima a inteligência humana, as AGIs se diferenciam das ANIs na possibilidade de conseguirem atuar em muito mais áreas de conhecimento e atividades. Apesar de já existirem muitos computadores inteligentes capazes de fazerem múltiplas tarefas, ainda não existem projetos nessa categoria.

### **2.2.3 Superinteligência Artificial**

Superinteligências artificiais também ainda estão em fase de desenvolvimento, mas seriam entidades capazes de serem autônomas e conscientes, podendo replicar e superar o comportamento humano em suas funções intelectuais básicas, aprendizado ou tomada de decisões (GABRIEL, 2019).

## 2.3 TETRAPLEGIA

A tetraplegia é a perda dos movimentos dos braços tronco e pernas, geralmente, provocada por lesões que atingem a medula espinhal a nível da coluna cervical, devido a situações como traumatismos em acidentes, hemorragia cerebral, sérias deformidades na coluna ou doenças neurológicas.

Algumas situações que levam a tetraplegia são irreversíveis o que fazem a pessoa afetada a permanecer muito tempo em uma cama, outras podem resultar em uma fraqueza dependendo da intensidade da lesão.

A tetraplegia também pode estar acompanhada de outras complicações como:

- Alterações na sensibilidade da região comprometida;
- Alterações no tônus dos músculos dos membros afetados, podendo haver flacidez (tetraplegia flácida) ou espasticidade (tetraplegia espástica);
- Alterações no funcionamento da bexiga e do intestino;
- Dor neuropática, que é um tipo de dor provocado por lesões neurológicas.
- Disfunção sexual;
- Osteoporose;
- Úlceras por pressão;
- Outras alterações neurológicas, como surgimento de suor inexplicado ou alterações na circulação sanguínea;

(TUA SAÚDE, [entre 2007 e 2020]).

## 3 TECNOLOGIAS PARA O DESENVOLVIMENTO

### 3.1 DART

Dart é uma linguagem e programação muito bem estruturada mantida pela Google que foi criada para atuar com soluções Web. Dart é uma linguagem muito similar as linguagens de programação síncronas e tipadas que estão em mercado hoje, como JAVA, C++, ou C, é possível implementar programação multithread através de “Isolates” (BUCKETT, 2013)

A Figura 1 ilustra um código exemplo desenvolvido com Dart:

```
main() {  
    var d = "Dart";  
    String w = "World";  
    print("Hello ${d} ${w}");  
}
```

**Figura 1 - Exemplo de código Dart.**

Dart pode ser compilado em tempo de execução em seu processo Just-in-Time (JIT) e compilado para um código nativo Ahead-of-Time (AOT).

As vantagens de se utilizar AOT é que o código é compilado antes da execução fazendo que não haja interpretadores intermediários fazendo a tradução das instruções, assim as aplicações desenvolvidas com Dart tomam características nativas e uma performance muito elevada (NAPOLI, 2020).

A compilação JIT geralmente é utilizada para uma exibição de informações e atualização de telas mais rápidas através do recurso de “Hot Reload” no momento da depuração de programas e aplicativos desenvolvidos (NAPOLI, 2020).

A programas em Dart podem ser executados sendo compilados pelo Kit de Desenvolvimento da linguagem ou através da versão de desenvolvimento em Dart do Google Chrome mas também esses programas podem ser convertidos em Javascript (BUCKETT, 2013).

A sintaxe do Dart é muito parecida com a de outras linguagens de programação mais bem inseridas no dia a dia dos desenvolvedores o que facilita na hora do aprendizado e aperfeiçoamento da linguagem.

```
var name = 'Voyager I';
var year = 1977;
var antennaDiameter = 3.7;
var flybyObjects = ['Jupiter', 'Saturn', 'Uranus', 'Neptune'];
var image = {
  'tags': ['saturn'],
  'url': '//path/to/saturn.jpg'
};
```

**Figura 2 - Declaração de variáveis.**

A Figura 2 mostra como variáveis podem ser declaradas, nesse momento é possível ver similaridades com Javascript. Existe a possibilidade de dar tipos às variáveis bem como no Java (o tipo da informação antes do nome da variável), ao declarar-se uma variável que pode receber mais de um valor, essa variável passa a tomar o comportamento de uma lista.

```
if (year >= 2001) {
  print('21st century');
} else if (year >= 1901) {
  print('20th century');
}
```

**Figura 3 - Estrutura de condições.**

Na Figura 3 pode-se notar como são feitas as condições no Dart, essa sintaxe é muito comum entre outras linguagens de programação.

```

for (final object in flybyObjects) {
  print(object);
}

for (int month = 1; month <= 12; month++) {
  print(month);
}

while (year < 2016) {
  year += 1;
}

```

**Figura 4 - Laços de repetição.**

```

int fibonacci(int n) {
  if (n == 0 || n == 1) return n;
  return fibonacci(n - 1) + fibonacci(n - 2);
}

var result = fibonacci(20);

```

**Figura 5 - Exemplo de função.**

## 3.2 FLUTTER

Flutter é um framework utilizado para a criação de telas criado pela Google, e é utilizado no desenvolvimento de aplicativos nativos e reativos para iOS, Android e Web, mas também tem propostas de aplicação em desktop e dispositivos incorporados (Raspberry Pi, domésticos, automotivos etc.). Esse framework foi construído utilizando a linguagem de programação Dart que pode ser compilada para código nativo, fornecendo assim grande velocidade de execução (NAPOLI, 2020).

Flutter trabalha com widgets e ferramentas que auxiliam o desenvolvimento mobile, resultando em aplicações mais profissionais, complexas e bonitas. Widgets são componentes que estruturam os aplicativos, como se fossem blocos que devem ser posicionados e estilizados no desenvolvimento dos aplicativos. Por exemplo, através de um widget é possível criar um quadrado na tela e personalizá-lo como desejado. Os widgets são estruturados em árvores, e podem ser alinhados ou se sobrepor.



Para a renderização de seus componentes na tela dos dispositivos o Flutter utiliza a biblioteca Skia 2D escrita em C++ (NAPOLI, 2020).

### 3.3 SOCKETS

#### 3.3.1 TCP/IP

Uma rede é composta por dois ou mais dispositivos conectados, esses dispositivos podem se comunicar através de regras que padronizam a forma de enviar ou receber mensagens, essas regras são os protocolos. A arquitetura TCP/IP é uma regra geral que estrutura as redes atualmente, não é a única forma de se conectar dispositivos, porém, a arquitetura TCP/IP é um padrão adotado e faz com que os aparelhos de fornecedores diferentes possam enviar e receber mensagens.

A arquitetura TCP/IP é composta por uma camada de aplicação, transporte, rede, enlace e física.

A camada de aplicação é a camada mais próxima do usuário, onde são programados os sistemas e programas que vão conversar entre si através da rede.

A camada de transporte fornece suporte para a transferência de dados em uma rede. A partir da camada de aplicação é possível acessar funções e recursos da camada de transporte que por sua vez define como os pacotes serão enviados, recebidos e o controle de fluxo. Dentro da camada de transporte pode-se destacar os protocolos TCP e UDP onde, o protocolo TCP garante mais confiabilidade das informações a serem enviadas, por ter um controle de conectividade, e o protocolo UDP que não controla conectividade, isso faz com que ele seja mais performático, mas não garante que todos os pacotes sejam enviados e entregues.

A camada de rede é a camada responsável pelo endereçamento e roteamento dos dispositivos. A partir dela é possível definir qual dispositivo deverá receber a informação através do seu IP, definir qual rota da rede a informação deve percorrer, qual porta acessar e estabelecer a conexão entre os dispositivos.

A camada de enlace é a camada abaixo da camada de rede, ela é a porta de entrada dos dados trafegados na camada Física. A camada de enlace também é responsável por corrigir possíveis erros ocorridos na camada Física.

A camada física é composta pelo hardware e pela infraestrutura de conexão, como as placas de rede e cabeamentos.

### 3.3.2 Sockets

Sockets são canais de comunicação estabelecidos entre duas aplicações, esses canais estão localizados entre a camada de aplicação e a camadas de transporte, onde a camada de transporte fornece funções que possibilita a criação de sockets. A conexão é estabelecida na rede em que as aplicações estão disponíveis e através de um "ip:porta", os sockets de cada aplicativo podem ser acessados por mais de uma aplicação e um aplicativo pode abrir múltiplos sockets desde que todos estejam disponibilizados em portas diferentes. Sockets podem ser criados utilizando o protocolo TCP e o protocolo UDP da camada de transporte. Para a comunicação ser efetivada uma aplicação grava a mensagem a ser enviada e a aplicação destinatária deve recuperar essa mensagem do socket.

O presente trabalho utiliza das abstrações de sockets para o envio de imagens entre aplicativos em uma rede.

## 3.4 WEBSOCKET

WebSocket é um canal de comunicação assíncrona bidirecional orientado a eventos, e pode fornecer dados em tempo real. Nessa arquitetura é utilizado o protocolo HTTP para efetuar uma conexão inicial entre o cliente e o servidor, porém uma vez a conexão estabelecida as entidades conectadas podem manter a comunicação de forma assíncrona (LOMBARDI, 2015).

A facilidade de implementação e os ganhos em velocidade entre a comunicação de serviços, faz com que seja muito viável a utilização de WebSockets em aplicações que necessitam de informações em tempo real, como entidades financeiras, empresas de investimentos, ferramentas de chat, jogos online entre outras aplicações.

As conexões com WebSockets são orientadas a eventos, ou seja, sempre que uma condição for ocorrer, é disparado um evento que pode ser capturado pelas partes conectadas (LOMBARDI, 2015). Principais eventos:

- Open: Ocorre quando a conexão entre cliente servidor é aberta.
- Message: Ocorre quando uma das partes recebe uma mensagem, assim é possível recuperar as informações da mensagem assim que ela chegar.
- Error: Ocorre quando acontece uma falha na conexão ou no envio da mensagem.
- PING/POG: O servidor envia um Ping e espera uma resposta Pong do cliente para se certificar se ainda existe a conexão.
- Close: Ocorre quando a conexão cliente servidor é fechada.

As possibilidades de desenvolvimento utilizando WebSockets são imensas, foi realizado testes envolvendo essa tecnologia para o envio de imagens, porém, por utilizar o protocolo HTTP existe uma certa limitação de performance no envio de uma grande massa de informações, também a falta de bibliotecas que suportam essa tecnologia dentro do desenvolvimento Flutter fez com que WebSockets não fosse escolhido para compor o desenvolvimento do presente trabalho.

### 3.5 PYTHON

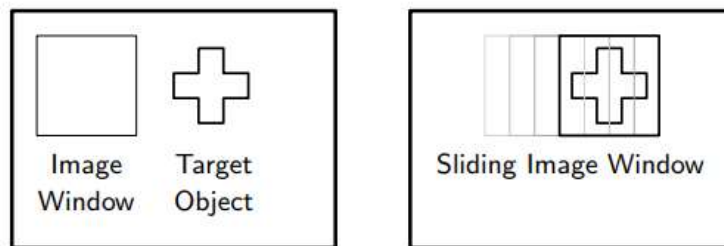
Python é uma linguagem de programação criada por Guido van Rossum em 1991. É uma linguagem interpretada, ou seja, não existe a necessidade de ser compilada previamente e por isso é muito utilizada como script, implementações, web, idioma, robótica, IA etc. (KUHLMAN, 2013). De acordo com PET, o Python foi desenvolvido para ter características simples de fácil programação e compreensão, mas tem a capacidade de suportar grandes aplicações.

PET também acrescenta que a ausência de marcações que delimitam o código, faz com que a leitura de programas escritos em Python seja muito mais fácil, e por isso é considerada uma linguagem de alto nível, abstraindo ao máximo características de máquina, essa simplicidade e sua alta capacidade faz com que Python seja muito utilizado no desenvolvimento de Inteligências Artificiais e programas relacionados à ciência de dados.

Outra vantagem de se utilizar Python é seu grande acervo de bibliotecas, ferramentas e uma comunidade muito ativa, esses fatores fazem com que o desenvolvimento em Python seja muito produtivo e o apoio da comunidade auxilia na detecção de problemas e o aprendizado.

### 3.6 ALGORITMO DE VIOLA-JONES

O objetivo do algoritmo de Viola-Jones é detectar objetos em imagens, e esse algoritmo varre uma imagem em pequenas posições, que podem ser chamadas de janelas, em busca de objetos com base em uma referência previamente determinada como ilustra a figura a seguir (LO, 2012).



**Figura 6 - Janela de busca aberta em imagem à procura de um objeto.**

Sempre que a janela se move, o conteúdo dentro dela é avaliado por classificadores binários, e caso o objeto desejado for encontrado os classificadores indicam uma saída positiva, e caso não, o classificador indica uma saída negativa. Essa técnica de detecção de objetos gera uma taxa de acerto para objetos encontrados e falsos positivos (LO, 2012).

Haar-Features são os classificadores aplicados sobre a janela a fim de encontrar traços característicos.



**Figura 7 - Haar-Features.**

Os pesos resultantes da aplicação de Haar-Features são gerados a partir da média de todos os pixels da parte branca subtraídos da média de todos os pixels da parte preta, esses valores são mais próximos de 1 quando o classificador encontra uma linha.

0.4	0.7	0.9	0.7	0.4	0.5	1.0	0.3
0.3	1.0	0.5	0.8	0.1	0.4	0.1	0.4
0.9	0.4	0.1	0.2	0.3	0.9	0.2	0.9
0.3	0.8	0.8	1.0	0.3	0.1	0.9	0.3
0.2	0.9	0.1	0.5	0.1	0.4	0.9	0.4
0.5	0.1	0.3	0.1	0.9	0.9	1.0	0.2
0.9	0.4	1.0	0.2	0.7	0.3	0.1	0.4
0.4	0.9	0.9	0.9	0.2	1.0	0.5	0.9

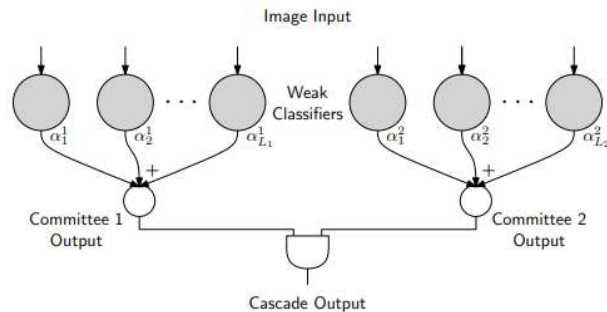
Figura 8 - Haar-Feature aplicada à imagem.

$$\begin{aligned}
 & (0.7 + 0.4 + 0.1 + 0.5 + 0.8 + 0.2 + 0.3 + 0.7 + 0.5 + \\
 & 0.1 + 0.4 + 0.8 + 0.9 + 0.6 + 1.0 + 0.7 + 0.3 + 0.1) / 18 \\
 & - \\
 & (1.0 + 0.5 + 0.8 + 0.4 + 0.1 + 0.2 + 0.6 + 0.8 + 1.0 + \\
 & 0.9 + 0.1 + 0.5 + 0.1 + 0.3 + 0.7 + 0.4 + 1.0 + 0.2) / 18 \\
 & \mathbf{0.51 - 0.53 = -0.02}
 \end{aligned}$$

Figura 9 – Cálculo do peso.

O classificador binário do algoritmo de Viola-Jones passa por duas etapas. A primeira etapa é onde vários “classificadores fracos” tentam identificar características de um objeto na área delimitada da janela, esses classificadores são muito ágeis para darem um resultado, porém tem uma taxa de acerto muito baixa, cada classificador gera uma saída positiva ou negativa, onde seu valor determina um peso que é adicionado à saída de todo o conjunto de classificadores onde é calculada uma média ponderada. A segunda etapa é responsável por recuperar a saída do conjunto de “classificadores fracos” e verificar de acordo com o resultado se o objeto está presente ou não, caso o valor resultante for positivo, a janela de verificação é passada para uma nova camada de classificadores, assim só é possível dizer

que o objeto desejado está na imagem quando todas as saídas dos classificadores da camada atual forem positivas (LO, 2012).



**Figura 10 - Classificador binário do algoritmo de Viola-Jones.**

Esse contexto em cascata para a detecção de objetos na imagem que o algoritmo de Viola-Jones aplica é um método chamado AdaBoost e serve para treinar os classificadores, as saídas negativas de “classificadores fracos” podem ser ajustadas de forma dinâmica com base no resultado de classificadores anteriores. De acordo com Chaves (2012) o algoritmo que trabalha junto ao AdaBoost (nesse caso Viola-Jones) é convocado de forma repetitiva e em cada repetição AdaBoost fornece ao algoritmo de Viola-Jones novos pesos referentes às execuções anteriores.

### 3.7 RASPBERRY PI

O Raspberry Pi (ou RPi) é um microcomputador completo, com seus componentes em uma única placa lógica. Há o processador, a memória RAM e a placa de vídeo impressos, e entradas USB, HDMI, áudio e vídeo composto, para câmera e telas LCD e uma GPIO, com pinos I/O de múltiplo propósito. A alimentação é feita através de uma porta microUSB, que permite usar fontes de energia de telefones celulares. O RPi pode ser utilizado como um computador normal, com teclado, mouse, monitor (TVs de tubo inclusas), fonte e um cartão microSD com o sistema e programas (GOGONI, [entre 2010 e 2020]).



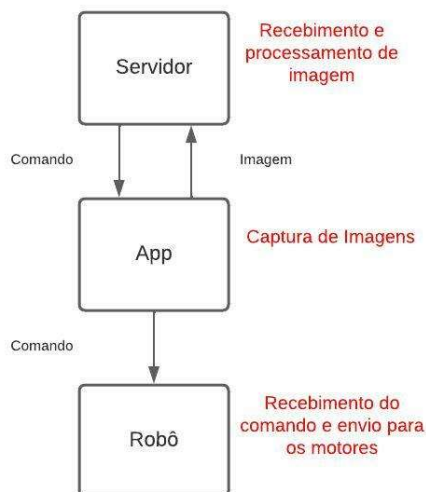
**Figura 11 - Raspberry Pi.**

## 4 DESENVOLVIMENTO DO PROJETO

O protótipo de robótica e IOT para cadeira de rodas inteligente tem por principal função o desenvolvimento de um robô que simula uma cadeira de rodas controlado por reconhecimento facial, capaz de captar comandos enviados pelo usuário.

O desenvolvimento do presente projeto foi separado em três etapas ilustradas na Figura 12:

- Desenvolvimento de um aplicativo móvel para captação de imagens;
- Desenvolvimento de serviço socket e inteligência artificial capaz de classificar imagens por reconhecimento facial;
- Confeção e programação do protótipo da cadeira de rodas.



**Figura 12 - Diagrama de fluxo do projeto.**

### 4.1 DESENVOLVIMENTO DO APLICATIVO MÓVEL

O Aplicativo foi desenvolvido utilizando a tecnologia Flutter e tem por objetivo acessar a câmera do celular e enviar fotos em tempo real para que o servidor possa detectar os comandos expressados na face do usuário.

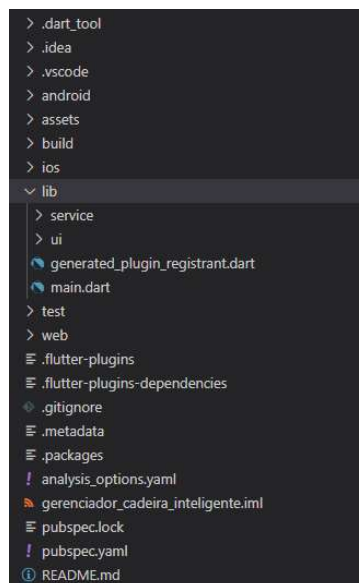


O arquivo que gerencia as dependências é o “pubspec.yaml” encontrado na pasta raiz do projeto. Para o desenvolvimento foi necessário adicionar às dependências os pacotes que auxiliam o gerenciamento da câmera, manipulação de imagens, banco de dados Sqlite e as demais dependências ilustradas na Figura 13.

```
dependencies:  
  flutter:  
    sdk: flutter  
  
  cupertino_icons: ^1.0.2  
  camera: ^0.9.7+1  
  image: ^3.1.1  
  ffi: ^1.1.2  
  sqflite:  
  path:  
  uuid: 3.0.6
```

Figura 13 – Dependências utilizadas no presente trabalho.

Na estrutura padrão dos aplicativos desenvolvidos em Flutter a pasta “lib” é onde se encontra a raiz do projeto, isto é, o arquivo “main.dart” onde todo o ciclo de vida do aplicativo é iniciado (Figura 14).



```
> .dart_tool  
> .idea  
> .vscode  
> android  
> assets  
> build  
> ios  
▼ lib  
  > service  
  > ui  
  generated_plugin_registrant.dart  
  main.dart  
  > test  
  > web  
  .flutter-plugins  
  .flutter-plugins-dependencies  
  .gitignore  
  .metadata  
  .packages  
  ! analysis_options.yaml  
  gerenciador_cadeira_inteligente.iml  
  pubspec.lock  
  ! pubspec.yaml  
  @ README.md
```

Figura 14 – Estrutura padrão projeto Flutter.

O primeiro passo a se realizar dentro do arquivo “main.dart” é capturar as câmeras disponíveis através do método “availableCameras” que retorna uma lista com todas as câmeras disponíveis, essa ação deve ser inserida dentro do método “main” para capturar os recursos de câmera no momento do início da execução do aplicativo.

```
List<CameraDescription> cameras = [];

Future<void> main() async {
  WidgetsFlutterBinding.ensureInitialized();
  cameras = await availableCameras();
  runApp(const MyApp());
}
```

Para a exibição das imagens recebidas na tela é necessário utilizar o Widget CameraPreview que recebe em seu construtor um objeto da classe CameraController que pode ser instanciado também a partir de seu construtor recebendo a câmera desejada, a resolução e o formato da imagem como parâmetros.

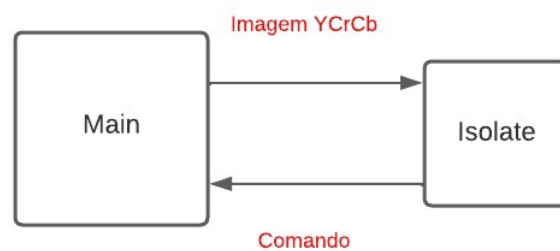
O Flutter disponibiliza imagens no formato “BGRA8888” para dispositivos IOS, e “YCbCr” para dispositivos Android. O presente trabalho foi desenvolvido com ênfase em dispositivos Android pela sua acessibilidade, contudo o tipo de imagem a ser abordado é o YCbCr que consiste no componente luminância (Y) e duas componentes de crominância (Cb e Cr). (TRANSCORTEC, 2022). De acordo com Hisour (2022), YCbCr é uma família de espaços de cores usada como parte do pipeline de imagens coloridas em vídeo e fotografia digital sistemas. Espaços de cores YCbCr são definidos por uma transformação de coordenadas matemáticas de um espaço de cores RGB associado.

A partir da classe CameraController é possível iniciar uma stream de imagem chamando o método “startImageStream”, com isso é possível tratar cada imagem de forma independente.

```
CameraController controller = CameraController(widget.cameras[1],
  ResolutionPreset.low,
  imageFormatGroup: ImageFormatGroup.yuv420);

controller.startImageStream((image) async {
  ...
}
```

Uma vez com as imagens sendo disponibilizadas a cada instante há a necessidade de enviá-las para o servidor python onde há uma inteligência artificial que processa cada imagem de forma individual, para isso foi criado um Isolate, que no Flutter tem função similar a threads, porém a memória entre Isolates não é compartilhada, e através de eventos entre o processo principal do aplicativo e o Isolate é possível realizar o envio das imagens para o servidor de forma assíncrona (Figura 15).



**Figura 15 – Conexão Processo principal – Isolate.**

Para Inicializar o Isolate deve-se utilizar o método estático “Isolate.spawn()” que recebe como parâmetro o método que será executado em paralelo, e quais os parâmetros que devem ser enviados para a nova área de memória alocada.

```

Future<Isolate> startStreaming(SendPort linkPort, SendPort commandPort) async {
  final dbRepository = DataBaseConfigRepository(context: null);
  ConfiguracaoModel? config = await dbRepository.findByKey('servidor');
  Isolate isolate = await Isolate.spawn(
    enviaImagem,
    SocketParams(
      linkPort: linkPort,
      commandPort: commandPort,
      servidor: config?.valor));
  return isolate;
}
  
```

A variável “linkPort” é enviada para o Isolate por é a partir dela que os eventos serão emitidos, logo para o processo principal da aplicação enviar as imagens ao Isolate é

necessário executar a função “linkport.send(...)” que emite um evento que pode ser capturado em outro processo, lembrando que isso é necessário pois Isolates não compartilham memória entre si.

```
void mudaControle() async {
  setState(() {
    _controle = !_controle;
  });
  if (_controle) {
    controller.startImageStream((image) async {
      if (_deveEnviar == true) {
        _deveEnviar = false;
        sendImage.send(image);
      }
    });
  } else {
    controller.stopImageStream();
  }
}
```

Após a conversa entre os processos de Isolate estar estabelecida, é possível realizar o envio das imagens para o socket. As imagens são transportadas criptografadas em base64 para o servidor através de uma conexão socket utilizando o protocolo TCP/IP. As Figuras 16 e 17 ilustram o resultado final do aplicativo.



**Figura 16 - App desligado**



**Figura 17 - App com comando para direita**

## 4.2 SERVIÇO SOCKET E RECONHECIMENTO FACIAL

### 4.2.1 Serviço socket

O serviço responsável por receber as imagens foi desenvolvido utilizando a linguagem de programação python. Para realizar a conexão entre o serviço e o aplicativo móvel foi

```
def monta_servidor():
    try:
        HOST=''
        PORT=8485

        sock=socket.socket(socket.AF_INET, socket.SOCK_STREAM)
        print('Socket criado!')

        sock.bind((HOST,PORT))
        sock.listen(2)
        print('Servidor iniciado')

        return sock
    except:
        print('Falha ao iniciar servidor')
```

utilizado Sockets TCP. No momento de instanciar o socket são passados os parâmetros “socket.AF\_INET” que define que o protocolo de endereço padrão utilizado foi o IPV4 e “socket.SOCK\_STREAM” simboliza o protocolo de conexão TCP.

Cada cliente conectado ao socket é tratado de forma individual através de threads, ou seja, a cada nova conexão ao socket, um novo thread também é aberto e só será terminado quando o cliente for desconectado.

```
def main():
    sock = monta_servidor()
    while True:
        conn,addr = sock.accept()

        for client,id,_ in conexoes:
            if addr[0] == client:
                conexoes.remove((client, id,_))

        identificador = str(uuid.uuid4())
        conexoes.append((addr[0], identificador, conn))

        thread = threading.Thread(target=recebe_mensagem_cliente, args=[conn,
        addr, identificador, conexoes])
        thread.start()
```

O método “recebe\_mensagem\_cliente” é responsável por iniciar as validações de recebimento de imagens, a partir dele é possível descobrir se a mensagem recebida veio do aplicativo móvel e se caso sim a mensagem já é direcionada para o método “processamento\_controle” que captura a mensagem completa e chama o método da

```
def recebe_mensagem_cliente(conexao, addr, uuid, conexoes):
    while True:
        try:
            header = conexao.recv(20)
            headerDecoded = header.decode()
            if headerDecoded[0] == '0':
                processamento_controle(header, conexao, addr, conexoes)
            else:
                processa_iot()
        except:
            print('Cliente desconectado')
            break
```

inteligência artificial que define o comando exibido na imagem. Uma vez com o comando definido é realizado um broadcast entre todos os clientes conectados em todos os threads.

#### 4.2.2 Reconhecimento facial

A implementação do reconhecimento facial é implementada em duas partes, a primeira é a detecção da face exibida na imagem e a segunda é o próprio reconhecimento do comando.

O algoritmo de Viola-Jones é utilizado para realizar a detecção da face, primeiramente há a necessidade de abrir o arquivo com o padrão do objeto a ser detectado, essa operação é realizada através do método “cv2.CascadeClassifier()”.

```
face_cascade= cv2.CascadeClassifier('C:/../haarcascade_frontalface_default.xml')
```

Em seguida deve-se realizar a chamada do método “detectMultiScale()” que realiza a detecção do objeto que corresponde ao arquivo carregado anteriormente, nesse caso, o arquivo capaz de mapear as faces.

```
faces = face_cascade.detectMultiScale(imagem, 1.2, 8)
```

Este método recebe como parâmetro a imagem em que se deseja realizar a detecção, o segundo parâmetro é um valor que define quantos vizinhos cada candidato retângulo deve ter para ser identificado, e o terceiro parâmetro diz respeito a quantidade máxima de faces a serem encontradas. Por fim a função “detectMultiScale()” retorna uma lista com as coordenadas de todas as faces encontradas.

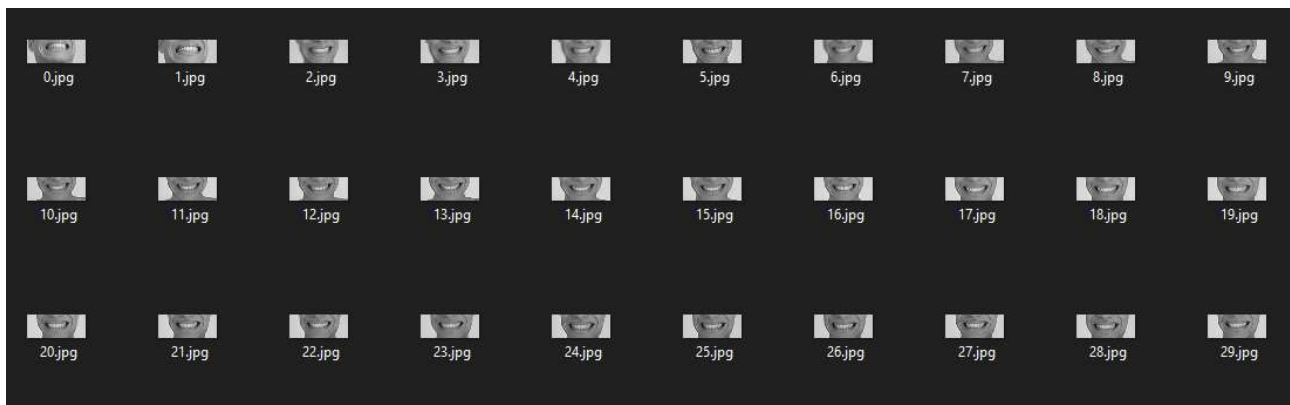
Por fim deve ser utilizado o algoritmo LBPH disponibilizado também pela biblioteca OpenCv, basicamente esse algoritmo utiliza de modelos matemáticos para reconhecer padrões, não necessariamente sendo considerado um algoritmo de aprendizado de máquina.

```
recognizer = cv2.face.LBPHFaceRecognizer_create()
```

A partir de um banco de imagens é possível treinar o algoritmo LBPH. No caso do presente trabalho o banco de imagens foi segmentado em várias pastas como ilustra a Figura 18 e a Figura 19, cada uma com uma série de imagens com as expressões que representarão os comandos. Para treinar basta carregar as imagens do banco de imagens executar o método “train” da instancia LBPH.

Nome	Data de modificação	Tipo	Tamanho
direita	11/07/2022 14:21	Pasta de arquivos	
esquerda	11/07/2022 14:21	Pasta de arquivos	
ligar	11/07/2022 14:21	Pasta de arquivos	
normal	11/07/2022 14:38	Pasta de arquivos	

**Figura 18 – Organização do acervo de imagens.**



**Figura 19 - Acervo de imagens (Comando Ligar).**

Por fim para reconhecer o comando exibido na imagem basta executar o método predict do algoritmo LBPH que retorna o identificador do comando correto mapeado pelo banco de imagens.

```
idf, conf = reconhecer.predict(roi)
comomando = commands[idf]
```



### 4.3 CONFECÇÃO E PROGRAMAÇÃO DO PROTÓTIPO DE CADEIRA DE RODAS

O protótipo da cadeira de rodas foi desenvolvido para receber mensagens através de wifi e assim processar o comando embutido na mensagem. A ideia é o robô confeccionado ser independente do aparelho que captura as imagens, com isso, qualquer smartphone android pode ser utilizado para controlar a cadeira. Para a confecção da cadeira de rodas foram necessários os materiais descritos na Tabela 2.

**Tabela 1 - Materiais e custos.**

<b>Material</b>	<b>Preço</b>
Regulador De Tensão Ajustável Lm2596 Dc-dc Step Down (Figura 21)	R\$ 30,00
Kit Raspberry Pi4 4gb De Ram C/ Case+cartão 64gb +hdmi+fonte	R\$ 800,00
Bateria Li-ion 18650 Sanyo 12v 4400mah (Figura 22)	R\$ 126,98
Driver Motor Ponte H L298n L298 (Figura 24)	R\$ 32,80
2 X Micro Motor Redução N20 12v 100rpm Suporte + Roda (Figura 23)	R\$ 101,06
Cabos para eletrônica	R\$ 20,00
<b>TOTAL: R\$ 1.110,84</b>	



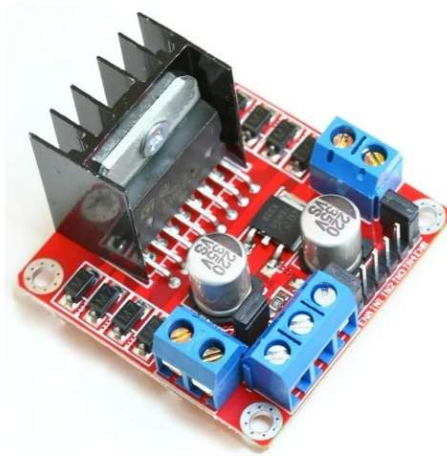
**Figura 20 - Bateria Li-ion 18650 Sanyo 12v 4400mah.**



**Figura 21 - Regulador De Tensão Ajustável  
Lm2596 Dc-dc Step Down.**

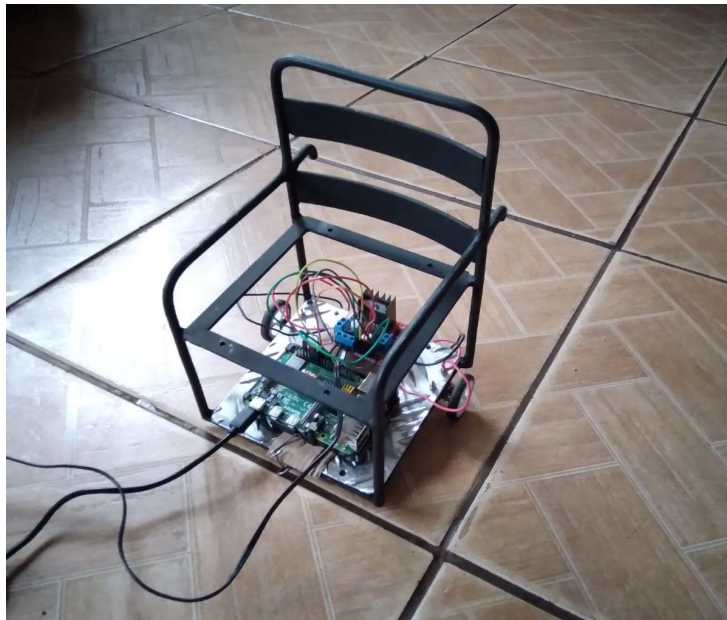


**Figura 22 - Micro Motor Redução N20 12v  
100rpm.**

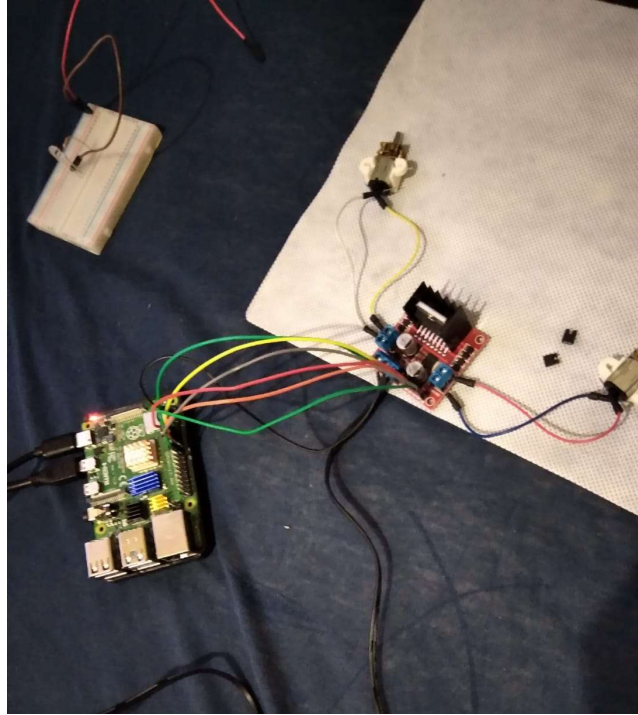


**Figura 23 - Driver Motor Ponte H  
L298n.**

Houve a necessidade de providenciar uma estrutura para o suporte das peças e simular o formato de uma cadeira de rodas convencional. Essa estrutura foi adquirida através de doação.



**Figura 24 - Protótipo pré-montado.**



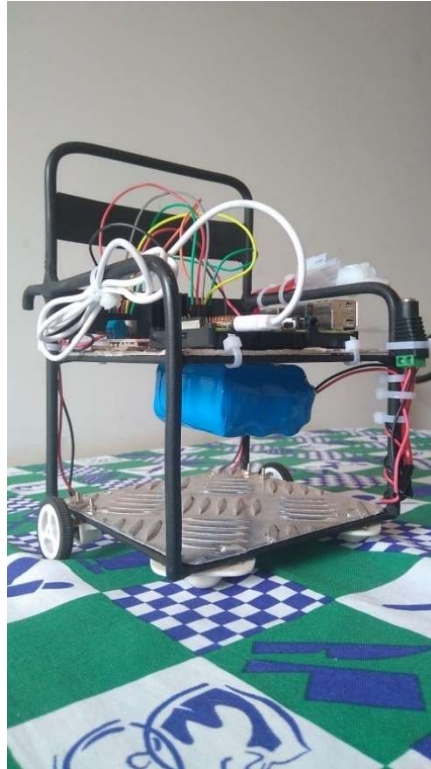
**Figura 25 - Pré-montagem eletrônica.**

A Figura 25 ilustra como as placas foram dispostas na estrutura da cadeira, a Figura 26 mostra como foi feita a conexão entre o Raspberry Pi a Ponte H e os motores.

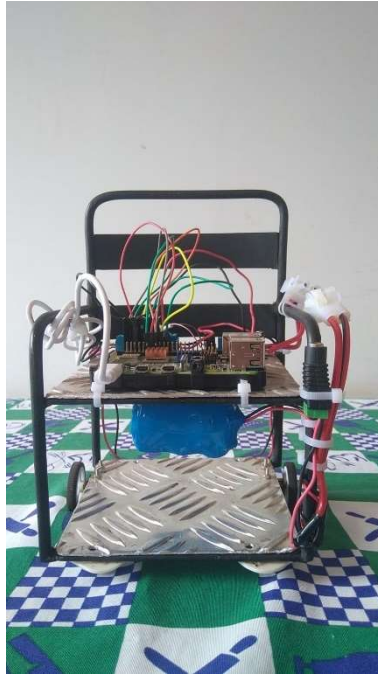
Inicialmente foi necessário energizar todos os componentes de forma individual, uma vez que os motores são de 12V (Voltz) e o Raspberry Pi só pode ser energizado em até 5V, para isso foram utilizadas fontes separadas com as respectivas voltagens a fim de realizar os testes. Ao ser incluído o regulador de tensão ajustável foi possível alimentar os motores e o Raspberry Pi a partir da mesma fonte de energia, a bateria de lítio.

O programa que roda no protótipo da cadeira também foi escrito em Python, utilizando a biblioteca GPIO, que possibilita acessar os pinos de prototipação Raspberry Pi, com isso é possível controlar os dois motores de forma independente, ou seja, para a cadeira andar para frente, basta energizar ambos os motores com a mesma velocidade, para virar para a direita é necessário desacelerar o motor da direita e acelerar o motor da esquerda, e vice-versa.

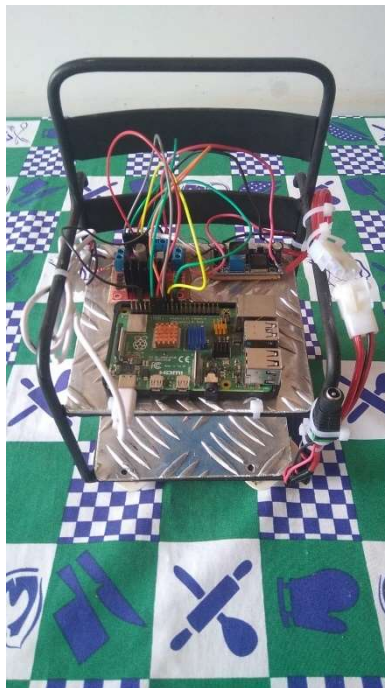
Para a recepção dos comandos do serviço que suporta a inteligência artificial, foi desenvolvido um cliente Socket que se conecta ao servidor e passa a escutar os comandos recebidos, e de acordo com o comando é executado um método que acelera os motores como descrito anteriormente. O resultado final do protótipo é ilustrado nas imagens, Figura 26, Figura 27 e Figura 28.



**Figura 26 - Protótipo em perspectiva.**



**Figura 27 - Protótipo visão frontal.**



**Figura 28 - Protótipo visão das conexões.**

## 5 CONSIDERAÇÕES FINAIS

O desenvolvimento do presente trabalho foi inspirado em fomentar a pesquisa e a implementação de soluções para as mais variadas limitações físicas em pessoas, sejam essas limitações ocasionadas por acidentes, problemas relacionados ao cérebro ou deficiências correntes desde o nascimento, como síndromes ou má formações.

Durante toda a pesquisa e desenvolvimento apresentados no presente trabalho, diversos desafios foram batidos e novas metas foram impostas. Também vários formatos e modelos que estruturaram o projeto foram surgindo ou sendo substituídos, até o modelo final apresentado.

O resultado esperado da implementação do protótipo foi atingido, o robô foi construído e funciona como desejado. O robô recebe mensagens de um servidor que processa imagens capturadas de um dispositivo móvel.

Um dos maiores desafios a ser superado foi o envio de imagens do aplicativo móvel para o servidor em tempo hábil para fazer manobras em tempo real com a cadeirinha. Uma vez que várias imagens são enviadas sucessivamente problemas de performance e tempo de resposta surgiram, diversas tecnologias foram testadas como é o caso de WebSockets, porém o resultado teve melhor desempenho quando o envio de imagens foi desenvolvido implementado por sockets.

O tempo de resposta das imagens enviadas do aplicativo móvel até a movimentação da cadeira foi melhor do que o esperado inicialmente, porém, em alguns casos dada uma certa iluminação ou variações muito bruscas do ambiente, há uma pequena falha de integridade no comando processado pela inteligência artificial, esse problema pode ser facilmente solucionado aumentando o acervo de imagens utilizadas para treinar a IA variando ambientes e iluminações.

### 5.1 TRABALHOS FUTUROS

O desenvolvimento do protótipo de robótica e IoT para cadeira de rodas inteligente levanta a possibilidade de a criação de um protótipo em tamanho real. Este trabalho pode ser levado

a diante e resultar em um produto acessível que realmente soluciona o problema de pessoas portadoras da tetraplegia ou paralisia.

Uma melhoria possível seria a combinação do Algoritmo de Viola-Jones com redes neurais convolucionais, essa mudança traria mais performance e integridade para os comandos processados.



## REFERÊNCIAS

AMAZON. **Os fatos sobre a tecnologia de reconhecimento facial com inteligência artificial.** Disponível em: <https://aws.amazon.com/pt/rekognition/the-facts-on-facial-recognition-with-artificial-intelligence/>. Acesso em: 28 nov. 2020.

ARAUJO, Thamara. **Quais são os principais tipos de inteligência artificial?** [S. l.], 21 dez. 2020. Disponível em: <https://people.com.ai/tipos-de-inteligencia-artificial/>. Acesso em: 12 mar. 2022.

BELLMAN, R. E. (1978). *An Introduction to Artificial Intelligence: Can Computers Think?* Boyd & Fraser Publishing Company.

CHAVES, Bruno B. **ESTUDO DO ALGORITMO ADABOOST DE APRENDIZAGEM DE MÁQUINA APLICADO A SENSORES E SISTEMAS EMBARCADOS.** 2012. Dissertação (Mestrado em Engenharia Mecânica) - Escola Politécnica da Universidade de São Paulo, [S. l.], 2012.

BUCKETT, Chris. **Dart in Action.** [S. l.: s. n.], 2013.

CORDEIRO, Bernardo, **BERT E WORD2VEC: UMA ANÁLISE INFERENCIAL E COMPUTACIONAL NA CLASSIFICAÇÃO DE TEXTOS COM REDES NEURAIS CONVOLUCIONAIS.** 2019. 51f. Monografia – Universidade federal do Rio de Janeiro, Rio de Janeiro, 2019.

FERREIRA, A. B. H. *Novo Dicionário Aurélio Básico da Língua Portuguesa.* Nova Fronteira, RJ, 1995.

FUKUDA, LEONARDO MASSAMI. **SEGURANÇA DA INFORMAÇÃO EM IOT.** 2019. MONOGRAFIA (Especialização em Gestão da Tecnologia da Informação e Comunicação) - Universidade Tecnológica Federal do Paraná, [S. l.], 2019.

GABRIEL, MARTHA. **TIPOS DE INTELIGÊNCIA ARTIFICIAL**. [S. l.], 30 jan. 2019. Disponível em: [https://gennegociosegestao.com.br/tipos-de-inteligencia-artificial/#:~:text=Intelig%C3%A2ncia%20Artificial%20Limita%20\(ANI\)&text=Se%20voc%C3%A2%20pedir%20para%20ela,jogar%20damas%2C%20ela%20n%C3%A3o%20saber%C3%A1](https://gennegociosegestao.com.br/tipos-de-inteligencia-artificial/#:~:text=Intelig%C3%A2ncia%20Artificial%20Limita%20(ANI)&text=Se%20voc%C3%A2%20pedir%20para%20ela,jogar%20damas%2C%20ela%20n%C3%A3o%20saber%C3%A1). Acesso em: 12 mar. 2022.

HISOUR. **Espaços de cores YCbCr**. Disponível em: <https://www.hisour.com/pt/ycbcr-color-spaces-26075/>. Acesso em: 12 mar. 2022.

KUHLMAN, Dave. **A Python Book: Beginning Python, Advanced Python, and Python Exercises**. [S. l.: s. n.], 2013.

LO, Charles. **A High-Performance Architecture for Training Viola-Jones Object Detectors**. 2012. Tese (Mestrado em Ciências Aplicadas) - Department of Electrical and Computer Engineering, University of Toronto, [S. l.], 2012.

LOMBARDI, Andrew. **WebSocket**. [S. l.: s. n.], 2015.

MARTINS, Lucimara. MURAMATSU, Mário. SERAPIÃO, Adriane. **Image classification of Kuzushiji ideograms with Convolutional Neural Networks**. 2019, Departamento de Estatística, Matemática Aplicada e Computação, Rio Claro, 12f. Universidade Estadual Paulista, 2019.

NAPOLI, Marco L. **Flutter: A HANDS ON GUIDE TO APP DEVELOPMENT**. [S. l.: s. n.], 2020.

P. Viola and M. J. Jones, **Robust real-time face detection** International Journal of Computer Vision, vol. 57, no. 2, pp. 137–154, May 2004.

PET. **INTRODUÇÃO À PROGRAMAÇÃO COM PYTHON**. IFSP - Câmpus São Carlos. (Apostila.)

RUSSELL, Stuart; NOVIRG, Peter. **Artificial Intelligence**. [S. l.: s. n.], 2013.

SANTOS, Bruno P; SILVA, Lucas A. M.; CELES, Clayson S. F. S.; NETO, João B.; PERES, Bruna S.; VIEIRA, Marcos Augusto M.; VIEIRA, Luiz Filipe M.; GOUSSEVSKAIA, Olga N.; LOUREIRO, Antonio A. F. **INTERNET DAS COISAS: DA TEORIA À PRÁTICA. Minicursos SBRC-Simpósio Brasileiro de Redes de Computadores e Sistemas Distribuídos**, Salvador, BA, v. XXXIV, p. 1-50, 19 set. 2016.

S. LAWRENCE, C. L. Giles, Ah Chung Tsoi and A. D. Back, **Face recognition: a convolutional neural-network approach** in *IEEE Transactions on Neural Networks*, vol. 8, no. 1, pp. 98-113, Jan. 1997, doi: 10.1109/72.554195.

TOTVS. **Inteligência artificial: porque a tecnologia tem se tornado cada vez mais estratégica?** Disponível em: <https://www.totvs.com/blog/inovacoes/o-que-e-inteligencia-artificial/>. Acesso em: 28 nov. 2020.

TRANSCORTEC. **Interface HDMI.** Disponível em: <https://www.transcortec.com.br/interface-hdmi.php>. Acesso em: 12 mar. 2022.

TRIPATHY, B.K.; ANURADHA, J. **INTERNET OF THINGS (IoT):** Technologies, Applications, Challenges, and Solutions. [S. l.: s. n.], 2018.

TUA SAUDE. **O que é tetraplegia e como identificar.** Disponível em: <https://www.tuasaude.com/o-que-e-tetraplegia>. Acesso em: 28 nov. 2020.

TURING, A. M. Computing Machinery and Intelligence. **Mind**, Oxford University Press, v. 59, n. 236, p. 433-460, 01 out. 1950.

## 6 APÊNDICE

```
@override
Widget build(BuildContext context) {
  if (!controller.value.isInitialized) {
    return Container();
  }
  return Column(mainAxisAlignment: MainAxisAlignment.start, children: [
    Container(...),
    Center(child: CameraPreview(controller)),
    Expanded(child: _controlesCaptura())
  ]);
}
```

```
late CameraController controller;

void _iniciaCamera() {
  super.initState();
  controller = CameraController(widget.cameras[1], ResolutionPreset.low,
    imageFormatGroup: ImageFormatGroup.yuv420);
  controller.initialize().then((_) {
    if (!mounted) {
      return;
    }
    setState(() {});
  });
}
```

```

void enviaImagen(SocketParams params) async {
  ReceivePort receiveImage = ReceivePort();
  Stopwatch stopwatch = Stopwatch();
  params.linkPort.send(receiveImage.sendPort);

  Socket socket = await _abreSocket(params.commandPort, stopwatch,
  params.servidor);

  receiveImage.listen((message) {
    stopwatch.start();
    CameraImage img = message;
    String header = "0:data:image;base64,";
    String base64Image = header + base64Encode(img.planes[0].bytes) + ',';
    socket.encoding = utf8;
    socket.write(base64Image);
  });
}

```

```

def trainData():
  global recognizer
  global trained
  global commands
  commands = os.listdir('train')
  trained = True

  ids = []
  faces = []

  for i,p in enumerate(commands):
    for f in os.listdir(f'train/{p}'):
      img = cv2.imread(f'train/{p}/{f}', 0)
      faces.append(img)
      ids.append(i)
  recognizer.train(faces, np.array(ids))

```

```
import cv2
import numpy as np
import os

boolsaveimg = False
recognizer = cv2.face.LBPHFaceRecognizer_create()
trained = False
commands = []
face_cascade = cv2.CascadeClassifier('D:/gerenciador-cadeira-inteligente/server-ia/haarcascade_frontalface_default.xml')

def recognize(imagem):
    faces = face_cascade.detectMultiScale(imagem, 1.2, 8)

    if(len(faces)>0):
        # RUN ALL FACES IN FRAME
        for (x,y,w,h) in faces:

            roi = imagem[y:y+h, x:x+w]
            roi = cv2.resize(roi,(50,50))[30:50]

            cv2.rectangle(imagem,(x,y),(x+w,y+h),(255,0,0),2)

            if trained:
                idf, conf = recognizer.predict(roi)
                com = commands[idf]
                cv2.putText(imagem,com, (x, y), 3, 1, (0,0,0), 2, cv2.LINE_AA)
                return imagem, com
    return imagem, '-'
```