

INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA DO RIO  
GRANDE DO NORTE

Romário Pereira da Silva

**Visão computacional: um estudo de caso aplicado à língua brasileira de sinais  
(LIBRAS)**

Nova Cruz/RN

2022

Romário Pereira da Silva

**Visão computacional: um estudo de caso aplicado à língua brasileira de sinais  
(LIBRAS)**

Trabalho de Conclusão de Curso apresentado ao Curso Superior de Tecnologia em Análise e Desenvolvimento de Sistemas do Instituto Federal de Educação, Ciência e Tecnologia do Rio Grande do Norte, em cumprimento às exigências legais como requisito parcial à obtenção do título de Tecnólogo em Análise e Desenvolvimento de Sistemas.

Orientador: Prof. Me. Alan Klinger Sousa Alves

Nova Cruz/RN

2022

Catálogo da publicação na fonte  
Biblioteca do Instituto Federal de Educação, Ciência e Tecnologia do RN  
*Campus Nova Cruz*

S586v SILVA, Romário Pereira da.

Visão computacional: um estudo de caso aplicado à língua brasileira de sinais (LIBRAS). / Romário Pereira da Silva. – Nova Cruz/RN, 2022.  
36f.

Orientador: Prof. Me. Alan Klinger Sousa Alves - Monografia (Monografia em ciências exatas e da terra). – Instituto Federal de Educação, Ciência e Tecnologia do Rio Grande do Norte – Nova Cruz/RN, 2022.

1. Desenvolvimento de software – Monografia. 2. Libras – Monografia. 3. Surdez - Monografia. I. ALVES, Alan Klinger Sousa.  
II. Título.

IFRN

CDU: 004.41(0813.2)

Romário Pereira da Silva

**Visão computacional: um estudo de caso aplicado à língua brasileira de sinais  
(LIBRAS)**

Trabalho de Conclusão de Curso apresentado ao Curso Superior de Tecnologia em Análise e Desenvolvimento de Sistemas do Instituto Federal de Educação, Ciência e Tecnologia do Rio Grande do Norte, em cumprimento às exigências legais como requisito parcial à obtenção do título de Tecnólogo em Análise e Desenvolvimento de Sistemas.

Trabalho aprovado. Nova Cruz/RN, 24 de Novembro de 2022.

BANCA EXAMINADORA

---

**Prof. Me. Alan Klinger Sousa Alves**  
Orientador - IFRN

---

**Prof. Me. Andre Vieira dos Santos**  
Examinador interno - IFRN

---

**Prof. Me. José de Anchieta Gomes dos Santos**  
Examinador interno - IFRN

## **AGRADECIMENTOS**

A todos os professores do Instituto Federal de Educação, Ciência Tecnologia do Rio Grande do Norte (IFRN) que colaboraram e construíram bases sólidas no meu desenvolvimento e aprendizagem para o crescimento profissional. Seus nomes são inesquecíveis e por isso, dedico-lhes minha profunda admiração e respeito.

A todos que acreditaram na realização deste trabalho e deram-me forças e estímulo para dar prosseguimento a esta pesquisa e obter sucesso. Em especial, meus familiares, meu orientador, Professor Me. Alan Klinger Sousa Alves, e aos meus colegas de turma.

A Deus criador dos céus e da terra, o que me deu a vida.

## RESUMO

Os surdos brasileiros enfrentam muitos problemas relacionados à comunicação com pessoas não surdas. Os primeiros desafios surgem no âmbito familiar, ainda na sua infância encontram dificuldades para se comunicarem com seus parentes. Quando crescem os desafios continuam quando precisam se relacionar com outras pessoas além de seus familiares, frequentar a escola, fazer faculdade ou conseguir um emprego. Além das dificuldades na comunicação com pessoas próximas, eles ainda têm que lidar com o descumprimento de leis criadas para protegê-los e garantir seus direitos, como a LEI Nº 13.146 que garante o direito ao atendimento igualitário por órgãos públicos, mas, na prática, isso raramente acontece. Tendo este cenário como campo de estudo, este trabalho apresenta uma ferramenta que utiliza o *framework MediaPipe* para capturar expressões faciais, posição e configuração das mãos em vídeos e traduzir palavras e letras em libras para o português brasileiro. Esta ferramenta conseguiu encontrar o significado dos sinais referentes às palavras Deus, direito, cunhado e muro, além das letras A, B, C, e D, identificando cada parâmetro formador do sinal individualmente e depois juntando-os para encontrar o significado.

**Palavras-chave:** Surdez; Tradutor; Libras; Português brasileiro

## ABSTRACT

Brazilian deaf people face many problems related to communication with non-deaf people. The first challenges arise in the family sphere, still in their childhood they find it difficult to communicate with their relatives. When they believe, the challenges continue when they need to relate to people other than their families, attend school, go to college or get a job. In addition to the difficulties in communicating with close people, they still have to deal with non-compliance with laws created to protect them and guarantee their rights, such as LAW N<sup>o</sup>. 13,146 which guarantees the right to equal treatment by public bodies, but in practice this rarely happens. Having this scenario as a field of study, this work presents a tool that uses the MediaPipe framework to capture facial expressions, position and configuration of hands in videos and translate words and letters in Libras into Brazilian Portuguese. This tool was able to find the meaning of the signs referring to the words God, right, brother-in-law and wall, in addition to the letters A, B, C, and D, identifying each parameter forming the sign individually and then putting them together to find the meaning.

**Key words:** Deafness, Translator, LIBRAS, Brazilian portuguese

## LISTA DE ILUSTRAÇÕES

Figura 1 – Detalhes dos <i>hand landmarks</i> . . . . .	15
Figura 2 – Detalhes dos <i>face landmarks</i> . . . . .	16
Figura 3 – Detalhes dos <i>pose landmarks</i> . . . . .	17
Figura 4 – Detalhes dos ids e posições dos <i>hand landmarks</i> . . . . .	19
Figura 5 – Captura da configuração de mão correspondente à letra “A” do alfabeto manual. . . . .	20
Figura 6 – dicionário contendo as configurações de mão capturadas. . . . .	20
Figura 7 – Detalhes do posicionamento dos <i>face landmarks</i> . . . . .	24
Figura 8 – Detalhes das áreas definidas como pontos de articulação. . . . .	25
Figura 9 – Detalhes dos <i>landmarks</i> usados para a detecção do parâmetro de direção da palma da mão. . . . .	27
Figura 10 – Tradução do sinal “cunhado”. . . . .	29
Figura 11 – Tradução do sinal “Muro”. . . . .	30
Figura 12 – Tradução do sinal “Direito”. . . . .	31
Figura 13 – Tradução do sinal “Deus”. . . . .	32



## LISTA DE TABELAS

Tabela 1 – Percentual de acertos na identificação dos sinais . . . . .	34
--	----

## SUMÁRIO

<b>1</b>	<b>INTRODUÇÃO</b>	<b>10</b>
1.1	OBJETIVO	11
1.2	OBJETIVOS ESPECÍFICOS	11
1.3	JUSTIFICATIVA	11
<b>2</b>	<b>REFERENCIAL TEÓRICO</b>	<b>13</b>
2.1	TRABALHOS RELACIONADOS	13
2.2	FORMAÇÃO DOS SINAIS NA LIBRAS	14
2.3	O <i>FRAMEWORK MEDIAPIPE</i>	14
2.4	<i>OPENCV</i>	17
2.5	PYTHON	17
<b>3</b>	<b>METODOLOGIA</b>	<b>18</b>
3.1	BIBLIOGRAFIA SOBRE OS PARÂMETROS DA LIBRAS	18
3.2	IMPLEMENTAÇÃO	18
<b>3.2.1</b>	<b>Detecção das configurações de mãos</b>	<b>19</b>
<b>3.2.2</b>	<b>Detecção dos movimentos das mãos</b>	<b>21</b>
<b>3.2.3</b>	<b>Detecção das expressões faciais</b>	<b>23</b>
<b>3.2.4</b>	<b>Detecção do ponto de articulação</b>	<b>25</b>
<b>3.2.5</b>	<b>Detecção da orientação da palma da mão</b>	<b>26</b>
<b>3.2.6</b>	<b>Tradução dos sinais em libras para o português</b>	<b>27</b>
<b>4</b>	<b>TESTES</b>	<b>33</b>
<b>5</b>	<b>CONCLUSÃO</b>	<b>35</b>
5.1	TRABALHOS FUTUROS	35
	<b>REFERÊNCIAS</b>	<b>36</b>

## 1 INTRODUÇÃO

De acordo com um estudo realizado pelo Instituto Locomotiva em parceria com a Semana da Acessibilidade Surda, a deficiência auditiva atinge cerca de 10,7 milhões de brasileiros (LOCOMOTIVA, 2019 apud FREITAS, 2021). Um dos motivos para esse resultado é o processo de envelhecimento da população pelo qual o Brasil está passando neste momento. 9% das pessoas com deficiência auditiva nasceram com essa condição e 91% adquiriram ao longo da vida, sendo que metade foi antes dos 50 anos.

Freitas (2021) destaca que as pessoas surdas no Brasil encontram dificuldades para exercer direitos básicos como lazer, o convívio social, acesso à saúde, educação, informação e inserção no mercado de trabalho. somente 7% têm ensino superior completo; 15% frequentaram até o ensino médio, 46% até o fundamental e 32% não possuem grau de instrução.

O mercado de trabalho, aliás, é um grande desafio a ser vencido. Dos que se arriscam nele, 37% são autônomos. Segundo Renato Meirelles, presidente do Instituto Locomotiva (FREITAS, 2021), “essas pessoas desistiram de arrumar emprego e passaram a empreender para garantir o seu sustento”.

Muitos dos problemas enfrentados pela comunidade surda podem ser resolvidos ou amenizados melhorando a comunicação entre ela e a grande massa de ouvintes que a rodeia. Fortalecimento dos vínculos familiares e sociais, melhora na qualidade da educação recebida, acesso a locais e serviços que antes não eram adaptados para atendê-las são apenas alguns dos benefícios que poderão ser alcançados com a promoção da comunicação entre surdos e não surdos.

Algumas ações têm sido tomadas ao longo dos anos para favorecer a comunidade nos desafios do dia a dia e para garantir o acesso integral a seus direitos. Em 2002, por exemplo, a língua brasileira de sinais (Libras) foi reconhecida oficialmente como língua das comunidades surdas brasileiras (BRASIL, 2002).

Na área de tecnologias alguns projetos já têm dado bons resultados. Um deles é o Vlibras, descrito em seu site oficial como: “um conjunto de ferramentas gratuitas e de código aberto que traduz conteúdos digitais (texto, áudio e vídeo) em português para Libras, tornando computadores, celulares e plataformas *Web* mais acessíveis para as pessoas surdas (VLIBRAS, 2022).”

Apesar dos esforços para a criação de mecanismos legais que favoreçam a comunidade surda e dos bons resultados na tradução de áudio e textos para língua de sinais, ainda existe um grande desafio a ser superado: a criação de uma ferramenta que consiga traduzir diálogos em libras para o português brasileiro.

Vários estudos já foram realizados com o fim de resolver este problema, (SOUZA et al., 2021), (JUNIOR, 2016), (ANJO; PIZZOLATO; FEUERSTACK, 2012), grande parte deles envolve a aplicação de *machine learning* e visão computacional. Geralmente essas técnicas são aplicadas em vídeos de pessoas se comunicando em libras visando traduzir os diálogos ou parte deles para o português.

Sendo assim, este trabalho apresenta um protótipo de uma ferramenta para a tradução de sinais em libras para o português através da aplicação de um *framework* de visão computacional em vídeos, porém, destacando-se por propor a identificação dos cinco parâmetros formadores dos sinais em libras separadamente e depois sua combinação para então encontrar a tradução.

## 1.1 OBJETIVO

Desenvolver uma ferramenta digital capaz de traduzir sinais da libras para o português brasileiro usando a abordagem de traduzir cada parâmetro formador do sinal individualmente através da aplicação de um *framework* de visão computacional

## 1.2 OBJETIVOS ESPECÍFICOS

- Fazer levantamento bibliográfico de trabalhos relacionados.
- Estudar o *framework MediaPipe* para entender como suas soluções podem ser aplicadas ao desenvolvimento deste projeto.
- Usar o *framework MediaPipe* para detectar os cinco parâmetros formadores dos sinais na libras.
- Encontrar o sinal que está sendo realizado através da combinação destes parâmetros e traduzi-lo para o português brasileiro

## 1.3 JUSTIFICATIVA

Garantir às pessoas surdas o pleno exercício de seus direitos é fundamental para seu desenvolvimento como cidadão ativo na comunidade onde vive. E em meio a tantos desafios a dificuldade na comunicação com pessoas não surdas aparece como uma das maiores barreiras.

É difícil imaginar como uma pessoa surda assiste a um noticiário na TV que não oferece um serviço de tradução, como ela compra passagens em um aeroporto ou faz

*check-in* em um hotel onde os funcionários não são capacitados para entender a linguagem de sinais. Até mesmo atividades rotineiras do dia a dia como praticar um esporte coletivo, fazer consultas médicas e até mesmo conversar com familiares, por vezes, se tornam uma tarefa muito complicada.

Criar leis que obriguem instituições públicas e privadas a oferecer atendimento especializado a essas pessoas pode ser uma solução, mas isto não resolve tudo. Para que pessoas com deficiência auditiva tenham uma vida plena e independente, elas precisam se comunicar também fora das instituições. Precisam pedir informações a um estranho no ponto de ônibus, conversar com alguém que acabou de conhecer em uma festa ou com os amigos não surdos do seu ciclo social.

Construir uma ferramenta digital que resolva estes desafios pode ser a solução mais viável. Com os avanços tecnológicos dos últimos anos é possível colocar essa ferramenta ao alcance de qualquer pessoa que tenha um celular ou um computador a um baixo custo, ou até mesmo sem custos para o usuário se sua produção for uma iniciativa governamental ou *software* livre.

Atualmente existem vários estudos relacionados ao desenvolvimento de ferramentas que traduzam linguagem de sinais para o português (ANJO; PIZZOLATO; FEUERSTACK, 2012), (SOUZA et al., 2021). Muitas dessas ferramentas usam técnicas de *machine learning* e inteligência artificial para traduzir diálogos registrados em arquivos de vídeos de falantes de libras. Essas ferramentas geralmente ignoram que cada sinal é formado pela combinação de um conjunto de parâmetros independentes entre si.

A ferramenta proposta neste trabalho se diferencia por seguir uma abordagem em que cada parâmetro formador do sinal é identificado individualmente para depois serem combinados possibilitando a tradução do sinal.

O presente trabalho é estruturado em cinco partes. Na primeira é apresentada a introdução, onde é proposta uma visão geral sobre o projeto. A segunda parte é a do Referencial teórico que contém uma apresentação de conceitos relacionados ao desenvolvimento do projeto. A terceira parte é a da Metodologia, em que é apresentado os métodos utilizados para o seu desenvolvimento, na quarta parte são apresentados os testes e na quinta e última parte a conclusão com os resultados e as considerações finais.

## 2 REFERENCIAL TEÓRICO

Este capítulo irá abordar a teoria por trás da Língua Brasileira de Sinais, o *framework* MediaPipe utilizado no projeto, o OpenCV, Python e os trabalhos relacionados.

### 2.1 TRABALHOS RELACIONADOS

J. D. Barros Junior (JUNIOR, 2016) realizou um trabalho visando encontrar uma solução para a tradução automática de sinais da libras para a notação SignWriting. SignWriting é um sistema de escrita para línguas de sinais que representa os sinais graficamente por meio de um conjunto de símbolos. A estrutura de escrita do SW é composta pelos parâmetros de orientação da palma, configuração de mão, tipos de contatos, movimento, expressões faciais e localização corporal. Para seu estudo, Barros Júnior usou imagens estáticas ao invés de vídeos e considerou apenas o parâmetro de configuração de mãos.

O trabalho de Anjo, Pizzolato e Feuerstack (2012) apresenta um sistema que pode ser treinado para reconhecer gestos estáticos em um vídeo. Além de reconhecer os gestos, o sistema consegue reportar os resultados em tempo real para qualquer interface através de um protocolo TCP/IP. Para este trabalho foi usado o Kinect da Microsoft como ferramenta de captura dos movimentos. O *Kinect* é um sistema capaz de identificar e rastrear partes do corpo humano em um fluxo de vídeo. O alfabeto manual da língua brasileira de sinais foi escolhido como caso de uso (ANJO; PIZZOLATO; FEUERSTACK, 2012) para esta pesquisa.

Souza et al. (2021) apresentou um trabalho cujo objetivo foi desenvolver um sistema para tradução de libras para o português brasileiro, mas diferente dos demais estudos citados, este foi aplicado a um cenário específico do cotidiano: consultas médicas onde o(a) médico(a) falava português e o(a) paciente falava em libras. Para isto foi aplicada uma abordagem baseada em *Deep Learn* que conseguia encontrar e usar dados de um sinal em libras a partir de um vídeo (SOUZA et al., 2021).

Guerra et al. (2018) Apresentou em seu trabalho uma nova abordagem para a tradução da LIBRAS identificando as expressões faciais como elemento básico na formação da linguagem. Seu trabalho segue uma ideia semelhante à deste buscando traduzir a libras a partir da identificação de seus parâmetros básicos. Guerra, no entanto, teve o intuito de identificar apenas as expressões faciais enquanto este trabalho aborda os cinco parâmetros.

Bastos, Angelo e Loula (2015) Alcançaram em seu trabalho taxas de 96,77% na identificação de sinais manuais usando uma Rede Neural. Eles usaram descritores (HOG

e ZIM) para extrair informações relacionadas às bordas e formas das mãos em imagens digitais. A excelente taxa de resultados demonstra como a inteligência artificial pode contribuir para a identificação dos elementos formadores dos sinais na LIBRAS.

## 2.2 FORMAÇÃO DOS SINAIS NA LIBRAS

A Língua Brasileira de Sinais - Libras possui sua própria estrutura gramatical que se organiza a partir de parâmetros específicos que formam os níveis linguísticos (FERRAZ, 2019). Assim como nas línguas orais é possível produzir uma infinidade de mensagens a partir de um número limitado de elementos sonoros chamados fonemas, na Libras é possível produzir uma infinidade de mensagens a partir da combinação de um número limitado de elementos visuais. Tais elementos visuais são geralmente chamados de parâmetros e de acordo com Ferraz (2019) são:

- A configuração de mãos é a forma que a mão ou as mãos assumem na execução do sinal. Este parâmetro é usado na produção dos sinais considerados como datilologia (alfabeto digital/manual).
- O ponto de articulação ou locação é o local em que o sinal é articulado, dependendo do sinal o ponto de articulação pode se encontrar no corpo de quem realiza o sinal ou no espaço neutro.
- O movimento, como o nome sugere, é o movimento que a mão realiza durante a execução do sinal. Pode ser realizado por uma ou pelas duas mãos e pode ser classificado como unidirecional, bidirecional ou multidirecional.
- A orientação da palma da mão é basicamente a direção em que a palma da mão está direcionada durante a realização do sinal.
- Expressão facial e corporal é o parâmetro que auxilia a transmissão da mensagem. Através deste parâmetro é possível, por exemplo, identificar se uma mensagem expressa uma exclamação ou uma interrogação.

## 2.3 O *FRAMEWORK MEDIAPIPE*

O *framework Media Pipe* foi desenvolvido pela *Google* e aplica vários modelos de *machine learn* que trabalham juntos para identificar e rastrear mãos, rosto, o corpo humano e objetos do cotidiano em vídeos e imagens. Este *framework* oferece várias soluções que podem ser utilizadas para reconhecimento e rastreamento de partes do corpo humano e objetos em várias plataformas como web, mobile e computadores (MEDIAPIPE, 2022).

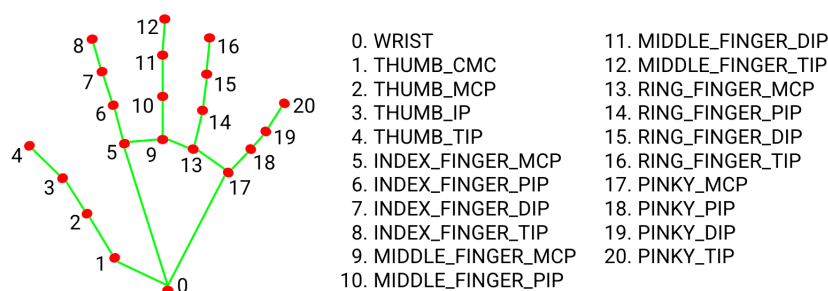
Com a documentação do *MediaPipe* foi possível entender como cada uma das soluções usadas para esta pesquisa funciona. A documentação da solução *Hands* usada nos módulos responsáveis por detectar as configurações de mãos, movimento, locação e orientação da palma da mão ajudou a identificar uma forma de capturar as posições dos *hand landmarks* (pontos que o framework projeta em locais específicos da mão em cada *frame* do vídeo). Cada *landmark* tem um identificador como mostrado na Figura 1.

A documentação da solução *Face* usada para detecção das expressões faciais não apresentava a identificação dos *landmarks* usados para fazer o rastreamento do rosto, porém ela fornece um link para um documento chamado *Real-time Facial Surface Geometry from Monocular Video on Mobile GPUs* (KARTYNNIK et al., 2019) que apresenta melhor o funcionamento do modelo de *Face 3D* usado por esta solução.

A solução *Pose* do *MediaPipe* usa aprendizado de máquina para identificar e rastrear poses do corpo humano com alta fidelidade. Ela permite estimar a posição e os movimentos dos membros do corpo humano. Esta solução foi combinada com a solução *Hands* e *Face* para que fosse possível identificar o parâmetro de locação. A seguir é apresentada uma descrição mais detalhada de cada solução usada neste projeto.

- *Hands*: A solução *Hands* emprega aprendizado de máquina (AM) para inferir 21 pontos de referência 3D nas mãos a partir de apenas um único quadro, estes pontos 3D são chamados de *hand landmarks*. Em imagens estáticas, após identificar a palma da mão e os dedos, a solução distribui 21 pontos em locais específicos das mãos e dos dedos. Em vídeos esta operação é realizada para cada *frame*. A Figura 1 demonstra os 21 *landmarks* usados por esta solução.

Figura 1 – Detalhes dos *hand landmarks*



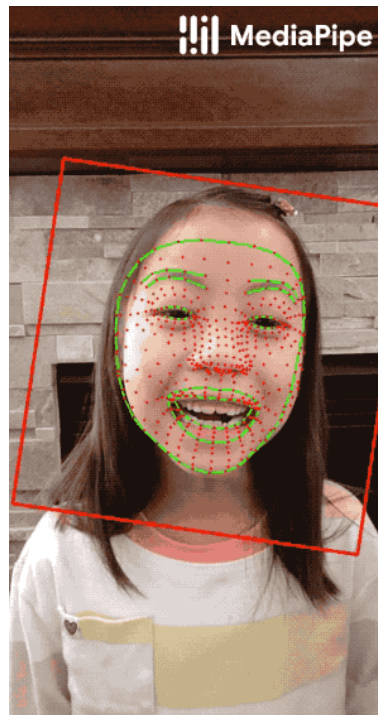
Fonte: <https://google.github.io/mediapipe/solutions/hands>

Cada um desses 21 pontos possui três eixos que representam sua posição no espaço 3D. Os eixos x e y representam os valores referentes à distância entre o ponto e a superfície superior do quadro e a distância entre o ponto e a superfície esquerda do quadro, respectivamente. O eixo z representa a profundidade que é a distância entre o ponto e a câmera (HANDS, 2022).



- *Face*: A solução *Face* aplica em tempo real 468 *face landmarks* em rostos humanos identificados em imagens ou vídeos. Esta solução consegue inferir a geometria da superfície do rosto, o que possibilita identificar a profundidade de cada *landmark*. Isto cria uma espécie de máscara 3D na superfície do rosto identificado no vídeo ou imagem como mostra a Figura 2.

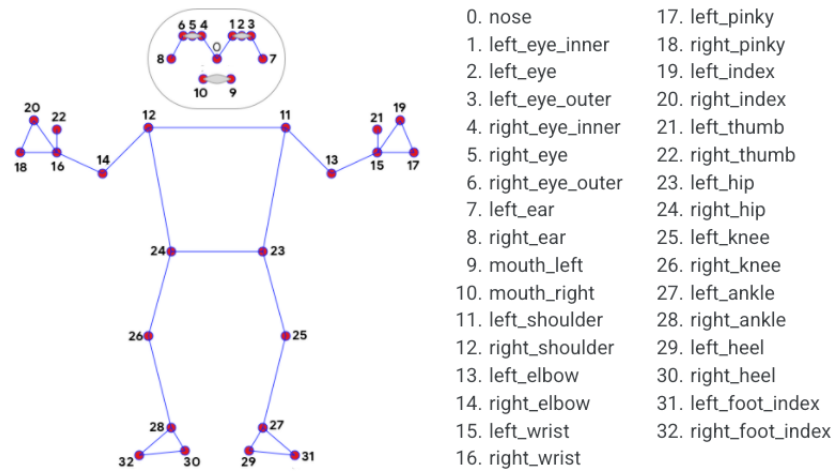
Figura 2 – Detalhes dos *face landmarks*



Fonte: [https://google.github.io/mediapipe/solutions/face\\_mesh.html](https://google.github.io/mediapipe/solutions/face_mesh.html)

Cada um desses *face landmarks* também possui três eixos usados para marcar sua posição no espaço 3D. Os eixos x, y e z marcam os valores referentes à distância entre o *face landmarks* e a superfície superior da imagem, a distância entre o *face landmarks* e a superfície esquerda da imagem e a distância entre o ponto e a câmera, respectivamente (FACE, 2022).

- *Pose*: A *Pose* é uma solução que usa aprendizado de máquina para rastreamento do corpo com alta fidelidade em imagens e vídeos. A solução infere 33 pontos de referência 3D (*pose landmarks*) em todo o corpo sendo usada para, como o nome da solução sugere, possibilitar a identificação de poses realizadas por uma pessoa. A Figura 3 apresenta os 32 *pose landmarks* inseridos pela solução *Pose*.

Figura 3 – Detalhes dos *pose landmarks*

Fonte: <https://google.github.io/mediapipe/solutions/pose.html>

Os pontos 3D da solução Pose seguem o mesmo modelo dos pontos de referência 3D da solução *Hands* e *Face* com três valores, x, y e z, representando a distância entre o *landmark* e a superfície superior da imagem, a distância entre o *landmark* e a superfície esquerda da imagem e a distância entre o *landmark* e a câmera respectivamente (POSE, 2022).

## 2.4 OPENCV

OpenCV (*Open Source Computer Vision Library*) é uma biblioteca de *software* de visão computacional e aprendizado de máquina de código aberto (OPENCV, 2022). Esta biblioteca possibilita a captura e o processamento de vídeos em tempo real. O *Framework MediaPipe* usa a OpenCv nos tutoriais que aparecem em sua página oficial, sendo assim ela também foi escolhida como a ferramenta de captura de vídeos para este projeto.

## 2.5 PYTHON

Uma linguagem de programação de propósito geral, alto nível e tipagem dinâmica que, apesar de ser orientada a objetos, suporta vários outros paradigmas de programação (PYTHON, 2022). Python é uma das linguagens mais usadas para trabalhos que envolvem *machine learn*. Por ser uma linguagem de alto nível e ter uma sintaxe simples baseada na língua inglesa ela permite que o desenvolvedor se dedique mais aos objetivos do projeto e menos às questões técnicas da linguagem.

### 3 METODOLOGIA

Além das pesquisas sobre trabalhos já realizados e tecnologias disponíveis citadas em seções anteriores, foram realizadas pesquisas sobre os aspectos linguísticos da libras.

A metodologia deste trabalho pode ser dividida em três partes principais:

- Pesquisa bibliográfica sobre os parâmetros formadores dos sinais na libras.
- Implementação
- Testes

#### 3.1 BIBLIOGRAFIA SOBRE OS PARÂMETROS DA LIBRAS

Em pesquisas na plataforma *Google scholar* surgiu como um dos resultados o material didático produzido para o Instituto Federal Sul-Rio-Grandense pelo professor Diogo Souza Madeira (MADEIRA, 2012). Este material faz uma introdução à língua libras e apresenta entre seus tópicos os cinco parâmetros formadores dos sinais em libras. A partir deste material foi possível dividir os sinais da libras em unidades menores para que pudessem ser analisados separadamente.

No livro Dicionário de configurações de Mãos (FERRAZ, 2019) o autor faz uma breve introdução aos parâmetros da libras para depois dar foco ao parâmetro de configurações de mãos. O autor apresenta 64 configurações de mãos. Cada configuração é apresentada com uma foto de boa qualidade, explicações e exemplos de palavras que usam a configuração. Este material contribuiu principalmente para a implementação do módulo responsável por identificar as configurações de mãos nos vídeos.

#### 3.2 IMPLEMENTAÇÃO

No processo de implementação os conceitos aprendidos nas pesquisas relacionadas aos aspectos linguísticos da Libras foram aplicados junto ao conhecimento das tecnologias escolhidas para que fosse possível desenvolver a ferramenta. A seguir são apresentadas descrições detalhadas de como cada fase da implementação foi executada.

### 3.2.1 Detecção das configurações de mãos

A solução *Hands* faz o rastreamento de mão e dedo empregando aprendizado de máquina (ML) para inferir 21 marcos 3D de uma mão a partir de apenas um único quadro. Isto significa que a cada quadro (*frame* do vídeo) o algoritmo identifica e aplica 21 pontos 3D sobre as mãos presentes no vídeo.

Durante o desenvolvimento percebeu-se que a captura de vídeo com baixa qualidade pode prejudicar a precisão das soluções. Na solução *Hands*, por exemplo, a diminuição da iluminação no ambiente de desenvolvimento, que era o mesmo ambiente em que os vídeos para os testes iniciais estavam sendo capturados, fazia com que os *hand landmarks* apresentassem maior instabilidade em seu posicionamento de um quadro para outro. Quando a iluminação era ajustada, a instabilidade diminuía e os resultados de captura de configuração de mãos eram mais precisos.

Cada *landmarks* tem um id que vai de 0 a 20 e os eixos x, y e z que guardam as posições x, y e z dos pontos no espaço 3D. O conjunto de todos os pontos de cada quadro é um contêiner contendo estes valores. A Figura 4 apresenta detalhes dos ids dos *landmarks* e os valores dos eixos x, y e z referentes à sua posição no espaço 3d.

Figura 4 – Detalhes dos ids e posições dos *hand landmarks*

```

0
x: 0.04652288928627968
y: 0.5754896402359009
z: -2.0136671082582325e-05

1
x: 0.07297967374324799
y: 0.6033364534378052
z: 0.014863954856991768

2
x: 0.08247300982475281
y: 0.635286271572113
z: 0.019429001957178116

3
x: 0.08244496583938599
y: 0.6666641235351562
z: 0.026150863617658615

4
x: 0.07958291471004486

```

Fonte: Elaborado pelo autor, 2021

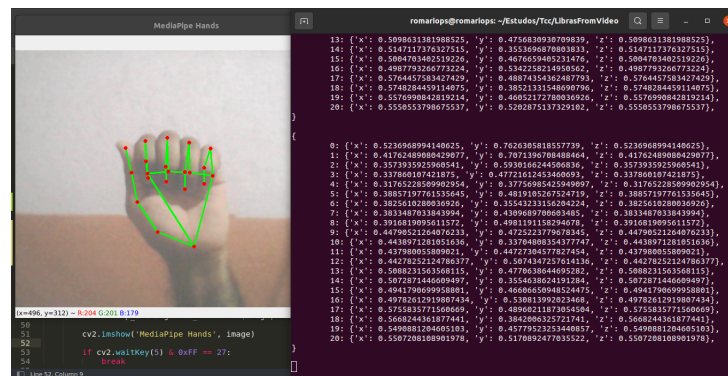
Para facilitar o tratamento destes valores foi criado um dicionário, estrutura de dados onde cada item é composto por uma chave, identificador único para cada valor, e um valor que pode ser de qualquer tipo, inclusive outro dicionário.

Neste dicionário, os ids de cada *hand landmark* são as chaves e seus valores são outros dicionários contendo os eixos x, y e z como chaves e os valores destes eixos como os

valores referenciados.

Usando o livro Dicionário de Configurações das Mãos em Libras (FERRAZ, 2019) como referência, uma configuração de mão era capturada pela webcam do *smartphone*. A cada *frame* capturado o algoritmo imprimia no terminal os dicionários citados anteriormente com os ids e os eixos x, y e z de cada *landmark*. Na Figura 5 é possível ver detalhes da configuração de mão referente à letra “A”.

Figura 5 – Captura da configuração de mão correspondente à letra “A” do alfabeto manual.



Fonte: Elaborado pelo autor, 2021

De forma manual, o dicionário foi copiado do terminal e colado em um módulo separado para ser usado posteriormente. Cada dicionário foi identificado e corresponde a uma configuração de mão. Todas as configurações de mão capturadas foram unidas em outro dicionário. Para facilitar o entendimento, o conjunto destas configurações de mãos capturadas será chamado de **configurações de mãos pré-definidas** daqui em diante. A Figura 6 demonstra os id's dos *landmarks* com os valores dos eixos x, y e z armazenados em uma estrutura de dados que foi chamada de *HANDS\_CONFIGURATION*.

Figura 6 – dicionário contendo as configurações de mão capturadas.

```

1
2 from hands_configuration.hands_config import hand_configuration_A
3 from hands_configuration.hands_config import hand_configuration_B
4 from hands_configuration.hands_config import hand_configuration_C
5 from hands_configuration.hands_config import hand_configuration_D
6
7
8 class HandsConfigurationManager:
9     def __init__(self):
10
11         self.HANDS_CONFIGURATION = {'A': hand_configuration_A.HAND_CONFIG_A,
12                                     'B': hand_configuration_B.HAND_CONFIG_B,
13                                     'C': hand_configuration_C.HAND_CONFIG_C,
14                                     'D': hand_configuration_D.HAND_CONFIG_D
15                                     }
16

```

Fonte: Elaborado pelo autor, 2021

A identificação das configurações de mãos buscou identificar se a configuração de mão capturada no vídeo era uma das configurações de mãos pré-definidas. O processo de identificação foi dividido em três etapas:

**Etapa 1:** Na primeira etapa foram medidas as distâncias entre os *hand landmarks* de cada uma das configurações de mãos pré-definidas. O algoritmo percorreu os dicionários em que os *landmarks* das configurações de mão foram armazenados e mediu a distância entre cada *landmark* e todos os outros *landmarks*. Os valores retornados também foram armazenados em um dicionário para serem usados posteriormente.

**Etapa 2:** Na segunda etapa, com o vídeo sendo capturado em tempo real, a solução *Hand* identificava e rastreava cada uma das mãos presentes no vídeo inferindo seus respectivos *landmarks*. A solução *hand* permite que seja definida a quantidade máxima de mãos que serão rastreadas, para este projeto foi definido que duas mãos seriam rastreadas.

Assim como na etapa anterior, as distâncias entre os *landmarks* das mãos rastreadas nos vídeos também foram medidas. Desta vez porém, os valores não foram armazenados, pois seriam usados e descartados na próxima etapa.

**Etapa 3:** Nesta última fase do processamento os valores das distâncias entre os *landmarks* capturados em tempo real no vídeo eram comparados aos valores das distâncias entre os *landmarks* das configurações de mãos pré-definidas. Por exemplo, a distância entre o *landmark* de id 0 e o *landmark* de id 1 da configuração de mão capturada no vídeo era comparada com a distância entre o *landmark* de id 0 e o *landmark* de id 1 de cada uma das configurações de mãos pré-definidas. Isto acontecia para todos os 21 *landmarks* que eram capturados no vídeo.

Percebeu-se que mesmo com pouca instabilidade havia uma pequena diferença nas posições dos *landmarks* de um frame para outro e isso fazia com que as distâncias entre os *landmarks* variassem mesmo quando a mão capturada no vídeo permanecia em uma mesma configuração. Para amenizar este problema foi definida uma variável com um valor de margem de erro para que caso a distância entre os *landmarks* estivesse dentro desta margem ela fosse considerada válida.

Também foi criado um variável para servir de contador para registrar quantas vezes as distâncias entre os *landmarks* eram equivalentes. Percebeu-se que algumas distâncias não ficavam dentro da margem de erro e, por tanto, não eram consideradas válidas. Desta forma, estabeleceu-se um valor arbitrário de 350 que correspondia à quantidade mínima de distâncias equivalentes para que a configuração de mão capturada no vídeo fosse considerada igual à configuração de mão pré-definida.

### 3.2.2 Detecção dos movimentos das mãos

A solução *Hands* também foi usada para a implementação do módulo responsável por detectar os movimentos das mãos. Este módulo detecta se a mão está se movimentando em uma das oito direções pré-definidas ou se a mão está parada. As direções pré-definidas

são:

- *Up*: movimento em linha reta direcionado para cima;
- *Down*: movimento em linha reta direcionado para baixo;
- *Left*: movimento em linha reta direcionado para esquerda;
- *Right*: movimento em linha reta direcionado para direita;
- *Left-Up*: movimento diagonal direcionado para cima e para esquerda;
- *Right-Up*: movimento diagonal direcionado para cima e para direita;
- *Left-Down*: movimento diagonal direcionado para baixo e para esquerda;
- *Right-Down*: movimento diagonal direcionado para baixo e para direita;
- *Stopped*: a mão está parada.

Como foi visto no tópico 2.2.1 a solução *Hands* identifica e rastreia mãos em vídeos e imagens e infere 21 pontos 3D (*hand landmarks*) em locais específicos da mão rastreada. Os eixos x e y de cada ponto 3D representam os valores referentes à distância entre o ponto e a superfície superior do *frame* e a distância entre o *landmark* e a superfície esquerda do *frame* respectivamente. Rastreando os valores destes dois eixos é possível saber se a mão realizou algum movimento em uma das direções mencionadas.

Para isso, foi necessário capturar as posições da mão duas vezes dentro de um certo período de tempo e depois verificar se houve alteração nos valores dos eixos x e y. se houve alteração nos valores desses eixos então a mão realizou algum movimento e a direção do movimento é retornada, se não houve alteração então a mão permanece no mesmo lugar sendo retornado o *status stopped*, parada.

Depois de alguns testes o período de tempo escolhido foi de meio segundo, pois percebeu-se que se o período de tempo fosse longo a mão poderia realizar um pequeno movimento e voltar para o mesmo lugar dando a impressão de que não havia se movido. Percebeu-se também que capturar as posições x e y de todos os 21 *hand landmarks* poderia gerar problemas, pois, por exemplo, se um dedo estivesse se movendo enquanto a palma da mão estava parada, as alterações nas posições dos *hand landmarks* do dedo davam a entender que toda a mão estava se movendo.

Sendo assim, apenas os valores dos eixos x e y do *landmark* com id 0 foram usados para detectar os movimentos da mão. Este *landmark* foi escolhido porque fica localizado próximo ao pulso, como mostrado na Figura 1. Dessa forma o movimento só é registrado se toda a mão mudar de lugar.

A cada meio segundo as posições dos eixos x e y do *hand landmark* 0 são capturadas e armazenadas duas vezes. Para verificar em qual direção a mão está se movendo é realizada uma sequência de comparações que seguem o seguinte exemplo: caso o valor de x na segunda captura seja maior que o valor de x da primeira captura e o valor de y permaneça o mesmo, a mão realizou um movimento para a direita sendo retornado o *status Right*, tendo em vista que o valor de x representa a distância do *landmark* para o limite esquerdo do frame (linha horizontal) e o valor de y representa a distância do *landmark* para o limite superior do frame (linha vertical).

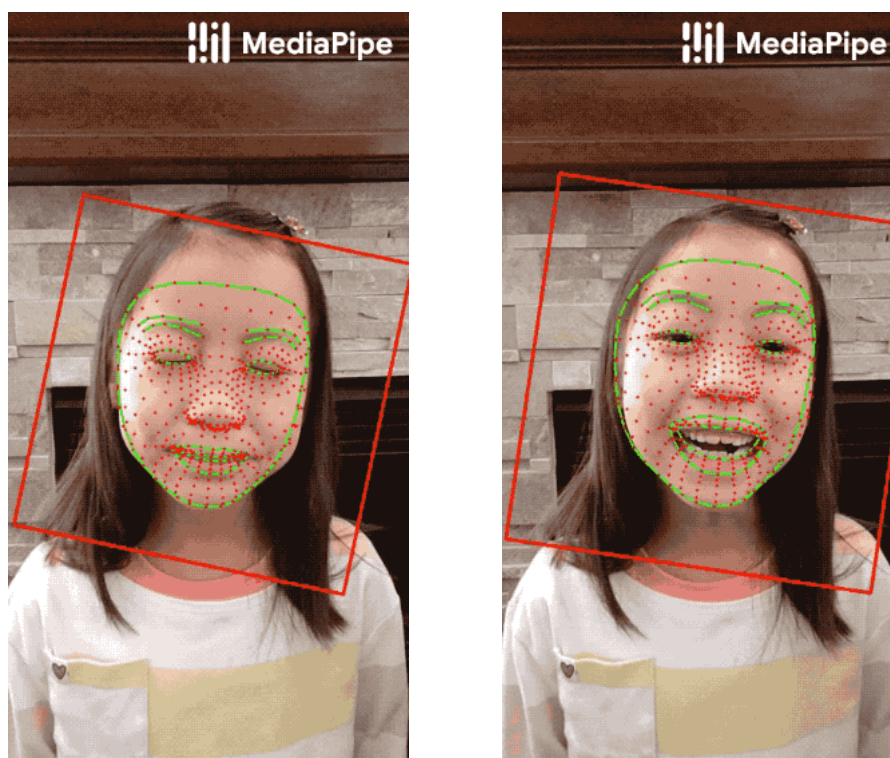
### 3.2.3 Detecção das expressões faciais

Para a detecção das expressões faciais foi usada a solução *Face* que rastreia e infere 468 *face Landmarks* em rostos humanos detectados em imagens e vídeos. cada um desses *landmarks* fica localizado em um local específico do rosto e, no caso dos vídeos, quando a expressão facial muda, o *landmark* muda os valores dos eixos x, y e z acompanhando sua localização original, como demonstrados nas Figuras 7a e 7b. A Figura 7a demonstra que os *landmarks* localizados na parte inferior e na parte superior da boca ficam próximos quando a boca está fechada. De forma semelhante a Figura 7b demonstra que os *landmarks* localizados na parte inferior e na parte superior da boca ficam distantes quando a boca está aberta.



Figura 7 – Detalhes do posicionamento dos *face landmarks*

(a) Posicionamento dos *face landmarks* (b) posicionamento dos *face landmarks*



Fonte: [https://google.github.io/mediapipe/solutions/face\\_mesh.html](https://google.github.io/mediapipe/solutions/face_mesh.html)

A estratégia para realizar a identificação das expressões faciais é parecida com a estratégia usada para detectar as configurações de mãos. Assim como os *hand landmarks*, os *face landmarks* também possuem os valores  $x$ ,  $y$  e  $z$  referentes aos valores das coordenadas  $x$ ,  $y$  e  $z$  do *landmark* no espaço 3D. Rastreando esses valores é possível saber as posições dos *face landmarks* nos *frames* e identificar o padrão em que se encontram. Sabendo qual padrão corresponde a uma determinada expressão facial é possível descobrir quando esta expressão facial aparece no vídeo.

Durante o desenvolvimento deste módulo percebeu-se que não era necessário (pelo menos não para a proposta deste projeto) processar os valores das coordenadas  $x$ ,  $y$  e  $z$  de todos os *face landmarks*, então foram escolhidos *landmarks* que pareciam sofrer maiores alterações em sua posição quando a expressão facial mudava e os que sofriam poucas alterações foram descartados. Por exemplo: *landmarks* localizados ao redor da boca mudam visivelmente de posição quando uma pessoa está séria e passa a sorrir ou boceja, porém os *landmarks* localizados no nariz, praticamente não alteraram sua posição.

No entanto, esta parte do desenvolvimento enfrentou alguns impedimentos. Parte deste projeto teve que ser desenvolvido nos laboratórios do instituto que foram parcialmente abertos durante a pandemia de COVID-19 com autorização das autoridades de saúde e seguindo rigorosos critérios de segurança sanitária. Para usar o laboratório os alunos

deveriam cumprir várias medidas de segurança sanitária, entre elas permanecer sempre de máscara, o que impediu a captura das expressões faciais em um primeiro momento.

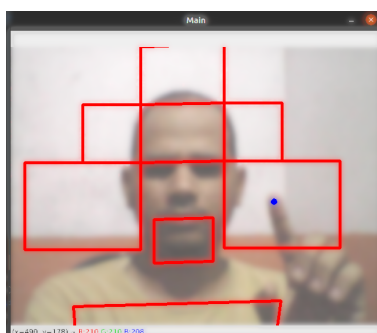
A seguir são apresentadas as expressões faciais que foram pré-definidas arbitrariamente para este projeto e as formas como o autor as descrevia.

- *Normal*: O rosto no estado normal, sem expressão aparente;
- *Angry*: Expressão de raiva expressada aqui como sobrancelhas empurradas uma em direção a outra e para baixo, lábios franzidos e fechados em forma de O e bochechas cheias de ar;
- *Surprise*: Expressão de surpresa. sobrancelhas curvadas para cima, olhos arregalados e boca semi-aberta;
- *Happy*: Expressão de felicidade. Boca sorrindo com os dentes superiores um pouco à mostra e sobrancelhas um pouco curvadas para cima;

### 3.2.4 Detecção do ponto de articulação

O ponto de articulação ou locação é o parâmetro referente ao local no corpo ou no espaço neutro onde o sinal é articulado. Para a detecção deste parâmetro foram usadas três soluções do *MediaPipe*: *Hands*, *Face* e *Pose*. As soluções *Face* e *Pose* foram usadas para desenhar retângulos no *frame* usando seus *landmarks* como referência e a solução *Hands* foi usada para identificar se a mão estava dentro dos limites de algum destes retângulos. A figura 8 apresenta as áreas definidas para a captura da detecção do parâmetro de locação.

Figura 8 – Detalhes das áreas definidas como pontos de articulação.



Fonte: Elaborado pelo autor, 2021

A figura 8 mostra os locais onde os sinais podem ser articulados, eles foram representados por áreas retangulares que cobrem certas partes do rosto e do corpo da pessoa que aparece no vídeo. Na prática, os sinais em libras são articulados no espaço

neutro ou em pontos específicos do corpo do falante, em alguns casos o falante de libras precisa tocar partes do corpo para realizar o sinal.

Porém, o algoritmo implementado neste projeto não pretendeu ser tão preciso a ponto de diferenciar se a mão está tocando o corpo ou está apenas à frente dele. Caso a mão entre em alguma das áreas pré-definidas como pontos de articulação, esta área é retornada. Foram definidas oito áreas como pontos de articulação, sete localizadas próximas ao rosto e uma à frente do tórax. São elas:

Rosto:

- *ABOVE THE HEAD*: Acima da cabeça;
- *FOREHEAD*: Testa;
- *LEFT OF FOREHEAD*: Lado esquerdo da testa;
- *RIGHT OF FOREHEAD*: Lado direito da testa;
- *LEFT OF FACE*: Lado esquerdo do rosto;
- *RIGHT OF FACE*: Lado direito do rosto;
- *CHIN*: Queixo.

Corpo:

- *CHEST*: Região ao tórax.

### 3.2.5 Detecção da orientação da palma da mão

O quinto e último parâmetro estudado neste projeto é o parâmetro Orientação da Palma da Mão que se refere à direção em que a palma da mão está apontando. Esta parte do desenvolvimento foi uma das mais desafiadoras, pois a mudança na orientação da palma da mão pode ser muito sutil e difícil de identificar.

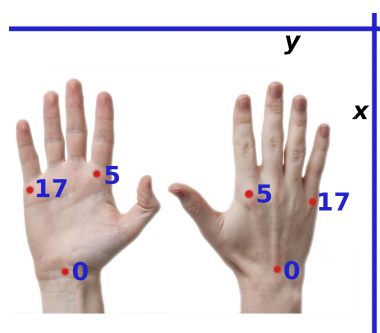
A solução *Hand* do *Media Pipe* foi usada para a identificação deste parâmetro. No entanto, diferente da identificação das configurações de mãos, que localiza e processa as posições de todos os *hand landmarks*, para identificar a orientação da palma da mão foram considerados apenas os *landmarks* de ID's 0, 5, e 17. Estes *landmarks* foram escolhidos porque ficam localizados nas extremidades da palma da mão como pode ser visto na Figura 1.

Estes *hand landmarks* foram escolhidos porque ficam localizados na ponta superior esquerda, ponta superior direita e na parte inferior do modelo de mão 3D usado pelo

*framework*. A estratégia para identificar a orientação da palma da mão e comparar as distâncias entres esses *landmarks* e os limites dos *frames*.

Na Figura 9 vemos um exemplo: se o *landmark* 5 está mais próximo do eixo x que o *landmark* 17 a palma da mão está virada para frente, caso contrário a palma da mão está virada para trás. O *landmark* 0, neste caso, serve para identificar se os dedos estão apontados para cima ou para baixo.

Figura 9 – Detalhes dos *landmarks* usados para a detecção do parâmetro de direção da palma da mão.



Fonte: Imagem editada pelo autor, 2021

Ao todo seis possíveis direções das palmas das mãos direções foram pré-definidas:

- *Front*: palma da mão virada para frente;
- *Back*: palma da mão virada para trás;
- *Top*: palma da mão virada para cima;
- *Down*: palma da mão virada para baixo;
- *Right*: palma da mão virada para direita;
- *Left*: palma da mão virada para esquerda.

### 3.2.6 Tradução dos sinais em libras para o português

O módulo de tradução dos sinais em libras tem uma lista de palavras que funciona como um banco de dados. Nesta lista, que é na verdade uma estrutura de dados, cada dado corresponde a uma palavra armazenada em português escrito e os parâmetros que formam o seu sinal correspondente em libras.

O processo de tradução dos sinais em libras para o português funciona da seguinte forma: uma pessoa realiza sinais da libras diante de uma câmera. A cada certo período de tempo pré-determinado o algoritmo identifica os parâmetros que estão sendo executados e

armazena-os em uma estrutura de dados. Nos testes realizados foi determinado um período de meio segundo.

Logo após, o algoritmo busca entre as palavras registradas a palavra que possui a lista de parâmetros idêntica à passada, caso não encontre o algoritmo refaz o processo. Caso a palavra seja encontrada ela é escrita no terminal e o algoritmo repete o processo.

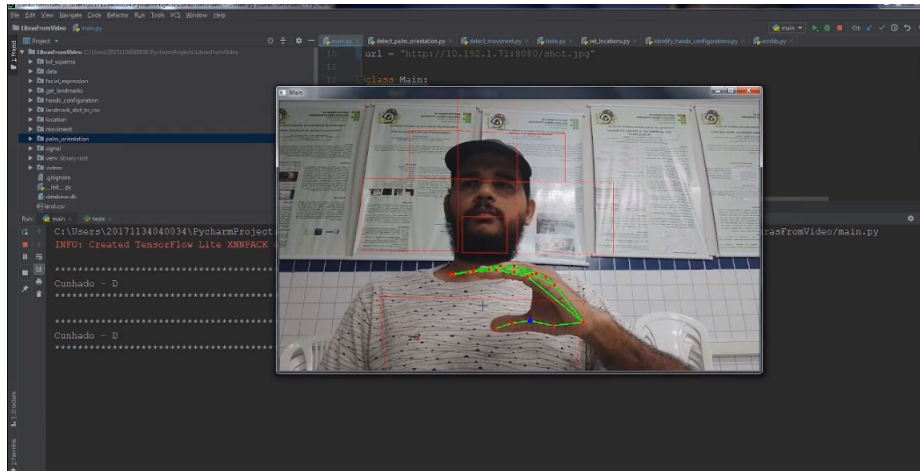
Nos testes foram definidas quatro palavras e cinco letras, foram as palavras cunhado, Deus, direito e muro e as letras A, B, C e D. Na libras os sinais que correspondem a uma palavra podem sofrer variações. Isso significa que pode haver mais de um sinal em libras para uma mesma palavra. As palavras usadas para os testes seguiram as seguintes formas de execução:

Cunhado:

- Mão Direita:
  - *Configuração da mão*: Configuração de mão em “c”.
  - *Locação*: Altura do peito.
  - *Movimento*: Da esquerda para a direita.
  - *Orientação da palma da mão*: Para esquerda.
  - *Expressão facial*: normal.
  
- Mão Esquerda:
  - *Configuração da mão*: Ignorado
  - *Locação*: Ignorado
  - *Movimento*: Ignorado
  - *Orientação da palma da mão*: Ignorado
  - *Expressão facial*: Ignorado.

A figura 10 apresenta detalhes da tradução do sinal em libras referente à palavra cunhado.

Figura 10 – Tradução do sinal “cunhado”.



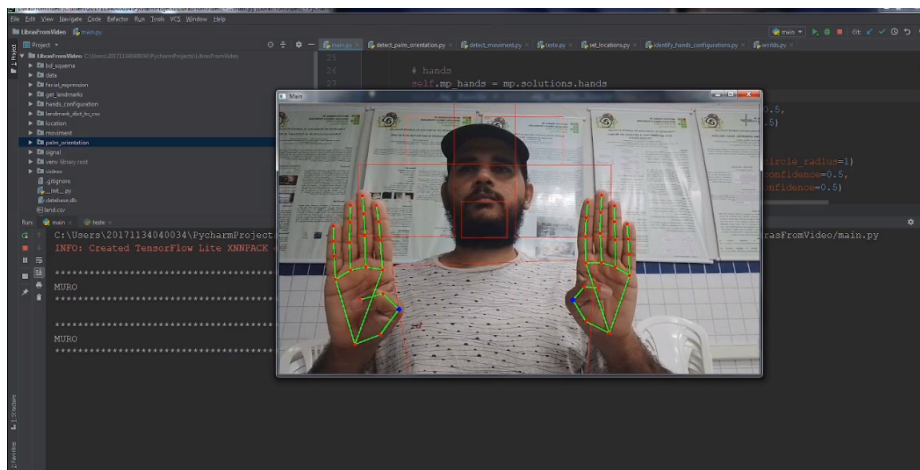
Fonte: Elaborado pelo autor, 2021

Muro:

- Mão Direita:
  - *Configuração da mão*: Configuração de mão em “B”.
  - *Locação*: Altura do peito.
  - *Movimento*: Da esquerda para a direita.
  - *Orientação da palma da mão*: Para frente.
  - *Expressão facial*: normal.
- Mão Esquerda:
  - *Configuração da mão*: Configuração de mão em “B”.
  - *Locação*: Altura do peito.
  - *Movimento*: Da direita para a esquerda.
  - *Orientação da palma da mão*: Para frente.
  - *Expressão facial*: normal.

A figura 11 apresenta detalhes da tradução do sinal em libras referente à palavra muro.

Figura 11 – Tradução do sinal “Muro”.



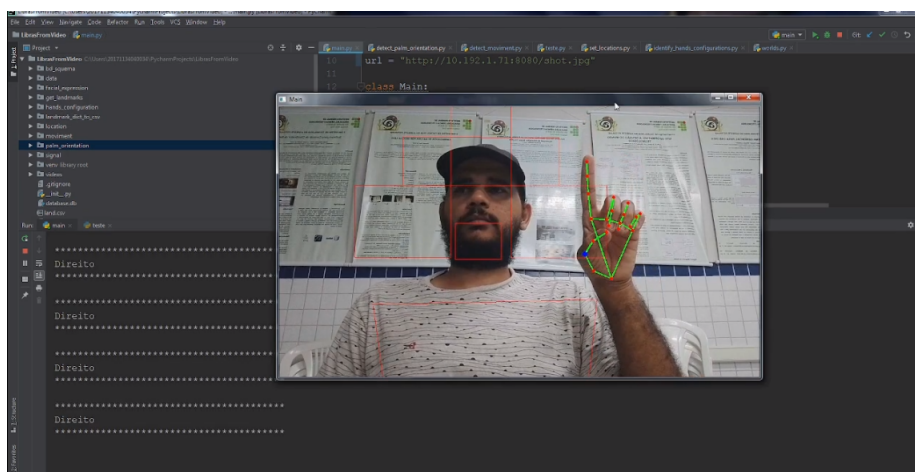
Fonte: Elaborado pelo autor, 2021

## DIREITO:

- Mão Direita:
  - *Configuração da mão*: Configuração de mão em “D”.
  - *Locação*: Lado direito do rosto.
  - *Movimento*: Para baixo.
  - *Orientação da palma da mão*: Para frente.
  - *Expressão facial*: Normal.
- Mão Esquerda:
  - *Configuração da mão*: Ignorado.
  - *Locação*: Ignorado.
  - *Movimento*: Ignorado.
  - *Orientação da palma da mão*: Ignorado.
  - *Expressão facial*: Ignorado.

A figura 12 apresenta detalhes da tradução do sinal em libras referente à palavra direito.

Figura 12 – Tradução do sinal “Direito”.



Fonte: Elaborado pelo autor, 2021

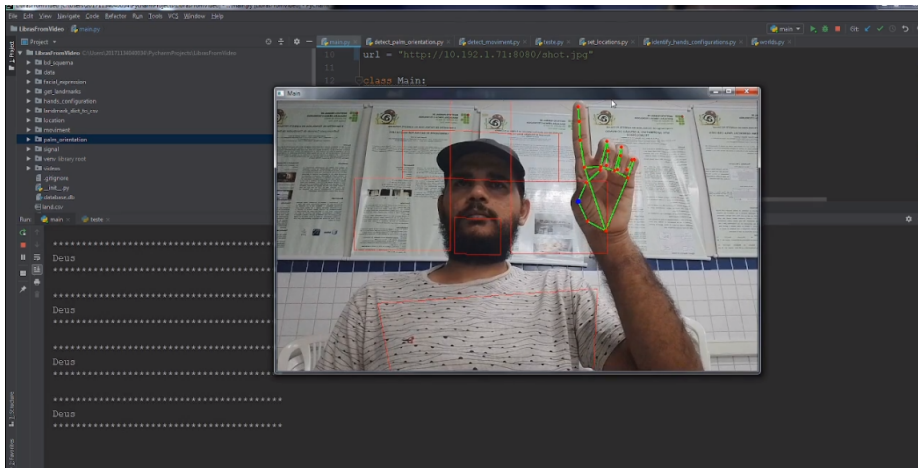


DEUS:

- *Mão Direita:*
  - *Configuração da mão:* Configuração de mão em “D”.
  - *Locação:* Lado direito do rosto.
  - *Movimento:* Para cima.
  - *Orientação da palma da mão:* Para frente.
  - *Expressão facial:* Normal.
- *Mão Esquerda:*
  - *Configuração da mão:* Ignorado.
  - *Locação:* Ignorado.
  - *Movimento:* Ignorado.
  - *Orientação da palma da mão:* Ignorado.
  - *Expressão facial:* Ignorado.

A figura 13 apresenta detalhes da tradução do sinal em libras referente à palavra Deus.

Figura 13 – Tradução do sinal “Deus”.



Fonte: Elaborado pelo autor, 2021

## 4 TESTES

Para realizar os testes foram usados três voluntários. Esses não tinham conhecimento de desenvolvimento de software e não falavam em libras, portanto, não sabiam quais os significados dos sinais que estavam executando, mas foram informadas que estes gestos faziam parte da linguagem de sinais usada por pessoas surdas. Eles foram orientados sobre como e onde deveriam posicionar as mãos, qual configuração suas mãos deveriam assumir e sobre qual movimento deveriam realizar. Cada voluntário realizou o sinal três vezes para cada palavra enquanto era filmado.

No momento de submeter os vídeos ao sistema surgiram alguns desafios. Como foi citado na seção 3.2.1, fatores como a qualidade do vídeo e a iluminação do ambiente onde o vídeo é capturado podem causar instabilidade no posicionamento dos *landmarks* dificultando a identificação dos parâmetros. Analisando-se os vídeos dos voluntários e pesquisando vídeos de falantes de libras na internet, percebeu-se que pessoas diferentes realizam os sinais de formas levemente diferentes e isto também pode dificultar a identificação dos parâmetros.

Para contornar este último obstáculo foram alterados os valores das variáveis de margem de erro e o contador criado na sessão 3.2.1. A primeira variável define a tolerância na variação do posicionamento dos *landmarks* de um *frame* para outro e a segunda variável mede a quantidade de resultados válidos entre as comparações das distâncias dos *landmarks* registrados e as comparações das distâncias dos *landmarks* capturados em tempo real. Como cada vídeo foi registrado em horários e locais com iluminação diferentes, considerou-se que não seria interessante apresentar os valores exatos de cada variável, pois não será possível mensurar a iluminação ou a forma exata que cada pessoa realizou o sinal.

O maior desafio dessa fase foi encontrar limites para os valores da variável de margem de erro e do contador que fossem baixos o suficiente para possibilitar a identificação dos parâmetros e que, ao mesmo tempo, não fossem tão baixos a ponto de, por exemplo, uma configuração de mão ser confundida com outra. Outro fato interessante é que, em alguns vídeos, os parâmetros eram identificados em apenas alguns *frames* o que fazia com que, por exemplo, se a execução de um sinal durasse um segundo rendendo um vídeo com 30 *frames*, em apenas dois ou três *frames* todos os parâmetros eram identificados corretamente possibilitando a tradução do sinal.

A Tabela 1 apresenta os resultados dos testes em percentual. Como cada voluntário executou três vezes cada sinal, cada acerto ficou equivalente a 33,3%. Pelos resultados é possível perceber que cada sinal foi traduzido pelo menos uma vez corretamente para cada voluntário e que o sinal equivalente à palavra muro foi o que teve mais acertos.

Tabela 1 – Percentual de acertos na identificação dos sinais

Voluntários	Sinais			
	Cunhado	Muro	Direito	Deus
Voluntário 1	66,6%	66,6%	33,3%	66,6%
Voluntário 2	33,3%	33,3%	66,6%	33,3%
Voluntário 3	33,3%	66,6%	33,3%	33,3%
Total	44,4%	55,5%	44,4%	44,4%

Fonte: Elaborado pelo autor

É importante observar que os vídeos usados nos testes não foram gravados em ambiente com iluminação controlada e isto pode ter afetado a capacidade do sistema detectar os parâmetros. Como foi citado anteriormente, os *landmarks* sofrem variações em sua posição original de um *frame* para outro e seu posicionamento pode se tornar muito instável dependendo da qualidade da iluminação. Provavelmente Testes com vídeos gravados em um ambiente com iluminação controlada poderiam exibir resultados mais satisfatórios.

## 5 CONCLUSÃO

As tecnologias da informação já fazem parte do cotidiano da maioria dos brasileiros. A velocidade com que elas se atualizam dificulta acompanhar seus avanços e proporcionam o desenvolvimento de uma gama de ferramentas para resolver os mais diversos tipos de problemas que vão desde chamar um táxi a fazer uma consulta médica online.

Apesar disso, criar uma ferramenta que possibilite a comunicação entre surdos e não surdos ainda é um desafio a ser vencido e o problema provavelmente não está em limitações tecnológicas. A humanidade conseguiu colocar um drone em Marte que envia fotos em alta resolução da superfície do planeta vermelho para a terra, mas ainda não desenvolveu uma ferramenta que traduza as línguas de sinais com eficiência.

Este trabalho teve como propósito apresentar o protótipo de um tradutor de língua de sinais que se destaca por seguir um modelo arquitetural que divide suas etapas de identificação e tradução do sinal em módulos individuais. As principais vantagens em dividir o processo de tradução em processos menores é que isto permite ao desenvolvedor explorar as tecnologias com o melhor desempenho em cada fase do processo de tradução.

Este projeto propôs a tradução de sinais em libras para o português usando a abordagem de identificar cada parâmetro do sinal separadamente e depois juntá-los para encontrar a tradução. Durante seu desenvolvimento surgiram vários desafios como a realização do sinal que sofre alterações de pessoa para pessoa dificultando o mapeamento dos parâmetros e a instabilidade dos *landmarks* que variava de acordo com fatores como a iluminação do ambiente onde o vídeo estava sendo capturado.

Apesar disso, o trabalho apresentou resultados que demonstram que os objetivos foram alcançados e que a ideia proposta funcionou possibilitando a tradução de quatro palavras da libras para o português, a saber: cunhado, Deus, direito e muro, além das letras A, B, C e D.

### 5.1 TRABALHOS FUTUROS

- Desenvolver uma versão que possibilite a tradução de diálogos simples.
- Desenvolver uma versão para plataformas mobile.
- Combinar o tradutor de libras para português com um tradutor de português para libras, como o VLibras, por exemplo, para criar um tradutor libras-português-libras.

## REFERÊNCIAS

- ANJO, M. d. S.; PIZZOLATO, E. B.; FEUERSTACK, S. A real-time system to recognize static gestures of brazilian sign language (libras) alphabet using kinect. In: **IHC**. [S.l.: s.n.], 2012. p. 259–268. Citado 3 vezes nas páginas 11, 12 e 13.
- BASTOS, I. L. O.; ANGELO, M. F.; LOULA, A. C. Recognition of static gestures applied to brazilian sign language (libras). 2015. Citado na página 13.
- BRASIL. Brasília, DF, 2002. Disponível em: [http://www.planalto.gov.br/ccivil\\_03/leis/2002/110436.htm](http://www.planalto.gov.br/ccivil_03/leis/2002/110436.htm). Acesso em: 23 set. 2022. Citado na página 10.
- FACE, M. [S.l.], 2022. Disponível em: [https://google.github.io/mediapipe/solutions/face\\_mesh.html](https://google.github.io/mediapipe/solutions/face_mesh.html). Acesso em: 23 set. 2022. Citado na página 16.
- FERRAZ, C. L. M. **Dicionário de Configuração de Mãos em Libras**. [S.l.]: Editora UFRB, 2019. Citado 3 vezes nas páginas 14, 18 e 20.
- FREITAS, K. Dia internacional da linguagem de sinais procura promover a inclusão de pessoas surdas. 2021. Disponível em: <https://www.al.sp.gov.br/noticia/?23/09/2021/dia-internacional-da-linguagem-de-sinais-procura-promover-a-inclusao-de-pessoas-surdas->. Acesso em: 23 set. 2022. Citado na página 10.
- GUERRA, R. R. et al. Facial expression analysis in brazilian sign language for sign recognition. 2018. Citado na página 13.
- HANDS, M. [S.l.], 2022. Disponível em: <https://google.github.io/mediapipe/solutions/hands.html>. Acesso em: 23 set. 2022. Citado na página 15.
- JUNIOR, J. D. B. Tradução automática de línguas de sinais: do sinal para a escrita. Universidade Federal do Pampa, 2016. Citado 2 vezes nas páginas 11 e 13.
- KARTYNNIK, Y. et al. Real-time facial surface geometry from monocular video on mobile gpu. 2019. Citado na página 15.
- LOCOMOTIVA, I. Agência brasil: País tem 10,7 milhões de pessoas com deficiência auditiva, diz estudo. 2019. Disponível em: <https://ilocomotiva.com.br/clipping/agencia-brasil-pais-tem-107-milhoes-de-pessoas-com-deficiencia-auditiva-diz-estudo>. Acesso em: 23 set. 2022. Citado na página 10.
- MADEIRA, D. S. **Língua Brasileira de Sinais**. [S.l.]: Universidade Aberta do Brasil, 2012. Citado na página 18.
- MEDIAPIPE. [S.l.], 2022. Disponível em: <https://google.github.io/mediapipe/>. Acesso em: 23 set. 2022. Citado na página 14.
- OPENCV. [S.l.], 2022. Disponível em: <https://opencv.org/about/>. Acesso em: 23 set. 2022. Citado na página 17.
- POSE, M. [S.l.], 2022. Disponível em: <https://google.github.io/mediapipe/solutions/pose.html#mediapipe-pose>. Acesso em: 23 set. 2022. Citado na página 17.

PYTHON. [S.l.], 2022. Disponível em: <https://www.python.org/about/>. Acesso em: 23 set. 2022. Citado na página 17.

SOUZA, R. S. de et al. Automatic recognition of continuous signing of brazilian sign language for medical interview. 2021. Citado 3 vezes nas páginas 11, 12 e 13.

VLIBRAS. [S.l.], 2022. Disponível em: <https://www.gov.br/governodigital/pt-br/vlibras>. Acesso em: 23 set. 2022. Citado na página 10.