

Centro Universitário Dom Bosco do Rio de Janeiro – UNIDOMBOSCO-RJ



UniDomBosco
Centro Universitário
Dom Bosco do Rio de Janeiro

Carlos Henrique Barros Silva Campos

Pedro Henrique Correa Rocha

Rafael Henrique Nogueira Alves

Victor Hugo Pacheco de Lima

T.A.L.I.A - Tradutor Automático de Libras com Inteligência Artificial

RESENDE

2023

Carlos Henrique Barros Silva Campos
Pedro Henrique Correa Rocha
Rafael Henrique Nogueira Alves
Victor Hugo Pacheco de Lima

T.A.L.I.A - Tradutor Automático de Libras com Inteligência Artificial

Trabalho de Conclusão de Curso apresentado ao Centro Universitário Dom Bosco do Rio de Janeiro Curso de Sistemas de Informação, como requisito parcial para obtenção do grau de Bacharel em Sistemas de Informação.

ORIENTADOR: BRUNO SALDANHA SAMPAIO

RESENDE
2023

Catálogo na fonte
Biblioteca Central da Associação Educacional Dom Bosco – Resende-RJ

C198 Campos, Carlos Henrique Barros Silva
T.A.L.I.A.: Tradutor Automático de Libras com Inteligência Artificial / Carlos Henrique Barros Silva Campos; Pedro Henrique Correa Rocha; Rafael Henrique Nogueira Alves; Victor Hugo Pacheco de Lima - 2023. 72f.

Orientador: Bruno Saldanha Sampaio
Trabalho de conclusão de curso apresentado como requisito parcial à finalização do curso de Sistemas de Informação do Centro Universitário Dom Bosco do Rio de Janeiro, da Associação Educacional Dom Bosco.

1. Informática. 2. Software. 3. Aplicativo. 4. Inteligência artificial. 5. Inclusão social. 6. Língua Brasileira de Sinais. 7. Libras. I. Rocha, Pedro Henrique Correa. II. Alves, Rafael Henrique Nogueira. III. Lima, Victor Hugo Pacheco de. IV. Sampaio, Bruno Saldanha. V. Centro Universitário Dom Bosco do Rio de Janeiro. VI. Associação Educacional Dom Bosco. VII. Título.

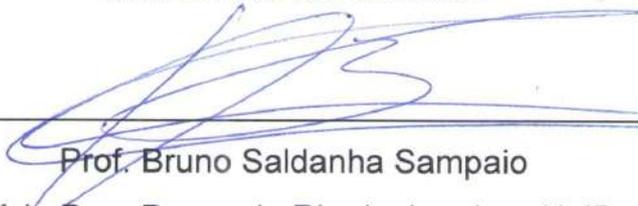
CDU 004.42(043)

Carlos Henrique Barros Silva Campos
Pedro Henrique Correa Rocha
Rafael Henrique Nogueira Alves
Victor Hugo Pacheco de Lima

T.A.L.I.A - Tradutor Automático de Libras com Inteligência Artificial

Trabalho de Conclusão de Curso apresentado ao Centro Universitário Dom Bosco do Rio de Janeiro, Curso de Sistemas de Informação, como requisito parcial para obtenção do grau de bacharel em Sistemas de Informação.

BANCA EXAMINADORA



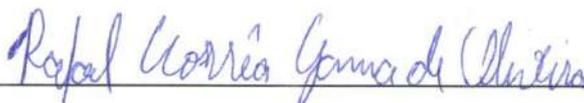
Prof. Bruno Saldanha Sampaio

Centro Universitário Dom Bosco do Rio de Janeiro - UniDomBosco - RJ



Prof. Maria Cristina Tavares de Moraes Danelon

Centro Universitário Dom Bosco do Rio de Janeiro - UniDomBosco - RJ



Prof. Rafael Corrêa Gama de Oliveira

Centro Universitário Dom Bosco do Rio de Janeiro - UniDomBosco - RJ

Resende, 02 de Dezembro de 2023.

A todos os surdos que já passaram pela dificuldade de ser compreendido.

AGRADECIMENTOS

Ao nosso orientador, professor Bruno Saldanha Sampaio, agradecemos pela dedicação, paciência, e apoio incondicional durante todo o processo de elaboração do nosso TCC. Sua orientação foi fundamental para o nosso desenvolvimento acadêmico e profissional. Agradecemos também às nossas famílias, que nos apoiaram e incentivaram durante toda a nossa trajetória acadêmica. Sem o amor e o apoio de vocês, não seria possível chegar até aqui.

"Sem linguagem não somos seres humanos completos e, por isso, é preciso aceitar a natureza e não ir contra ela. Obrigados a falar, algo que não lhes é natural, os surdos não são expostos suficientemente à linguagem e estão condenados ao isolamento e à incapacidade de formar sua identidade cultural."

Vendo Vozes: Uma Viagem pelo Mundo dos Surdos
(Sacks, Oliver ,2010, pag.1)

RESUMO

A surdez é um problema que afeta milhões de pessoas em todo o mundo e pode ser um obstáculo significativo na vida cotidiana. A comunicação é essencial para o ser humano, mas para aqueles que têm dificuldade em ouvir, a comunicação pode ser uma tarefa difícil. Isso pode levar a problemas de isolamento social, dificuldade em acessar informações importantes e falta de oportunidades educacionais e profissionais. Felizmente, a tecnologia está avançando rapidamente e, com isso, novas soluções estão sendo desenvolvidas para ajudar as pessoas que sofrem com a surdez. A inteligência artificial é a chave para o projeto que será apresentado neste artigo, pois graças ao uso de redes neurais artificiais, a TALIA é capaz de aprender a reconhecer sinais de Libras e traduzi-los em tempo real para aqueles que não entendem essa linguagem de sinais. Com isso, nosso projeto acabará se tornando um exemplo de como a tecnologia pode ser usada para melhorar a vida de pessoas surdas, essa tecnologia pode ajudar a reduzir a barreira de comunicação que muitas pessoas enfrentam quando tentam se comunicar com um surdo. Além disso, a tradução pode ser exibida tanto em forma de voz quanto em forma de texto, o que torna a tecnologia ainda mais acessível. Para aqueles que não têm experiência com Libras, a tradução de voz pode ser uma maneira rápida e fácil de entender a mensagem, por outro lado, para aqueles que são surdos e também não entendem Libras, a tradução em texto pode ser uma opção mais conveniente. Uma vez pronta, essa tecnologia pode ter aplicações em outras áreas, como educação e trabalho, onde a comunicação é essencial. É importante notar que essa tecnologia não é perfeita e ainda há espaço para melhorias. No entanto, é um passo importante na direção certa e mostra como a tecnologia pode ser usada para criar um mundo mais inclusivo e acessível para todos.

Palavras-chave: IA, Libras, Surdez e Inclusão Social

ABSTRACT

Deafness is a problem that affects millions of people worldwide and can be a significant obstacle in daily life. Communication is essential for human beings, but for those who have difficulty hearing, communication can be a difficult task. This can lead to problems of social isolation, difficulty accessing important information, and lack of educational and professional opportunities. Fortunately, technology is advancing rapidly, and with it, new solutions are being developed to help people who suffer from deafness. The artificial intelligence is the key for the project that will be presented in this article, because thanks to the use of artificial neural networks, TALIA is able to recognize Libras signs and translate them in real-time to the ones that do not understand this sign language. After this, our project will become an example of how technology can be used to improve deaf people's life. This technology can help to reduce the communication barrier that many people face when trying to communicate with a deaf person. In addition, the translation can be displayed both in voice and in text form, making the technology even more accessible. For those who have no experience with sign language, voice translation can be a quick and easy way to understand the message. On the other hand, for those who are deaf and cannot understand Libras too, text translation can be a more convenient option. Furthermore, this technology may have applications in other areas such as education and work, where communication is essential. It is important to note that this technology is not perfect and there is still room for improvement. However, it is an important step in the right direction and shows how technology can be used to create a more inclusive and accessible world.

Key words: AI, Libras, deafness, and social inclusion.

LISTA DE FIGURAS

Figura 1: Linguas de sinais pelo mundo.....	19
Figura 2: Diferentes sinais para "verde" dependendo do estado do Brasil.....	20
Figura 3: Gráfico de pessoas que sabem Libras com base no grau de dificuldade de ouvir.....	21
Figura 4: 120 ferramentas de IA poderosas.....	24
Figura 5: Partes de uma rede neural.....	27
Figura 6: Mediapipe: Detecção de pontos das mãos.....	30
Figura 7: Mediapipe: Detecção de pontos do corpo.....	30
Figura 8: Agenda 2030: Saúde e bem-estar.....	31
Figura 9: Agenda 2030: Educação de qualidade.....	32
Figura 10: Agenda 2030: Redução das desigualdades.....	32
Figura 11: Arquitetura do sistema.....	36
Figura 12: Logotipo do projeto.....	37
Figura 13: Python.....	38
Figura 14: TensorFlow.....	39
Figura 15: Visual Studio Code.....	39
Figura 16: OpenCV.....	40
Figura 17: MediaPipe.....	41
Figura 18: Keras.....	41
Figura 19: Pytsx3.....	42
Figura 20: Flask.....	42
Figura 21: Scikit-learn.....	43
Figura 22: Amazon EC2.....	43
Figura 23: Ubuntu.....	44
Figura 24: Notion.....	49
Figura 25: Repositório do Github.....	50
Figura 26: Pasta do Drive.....	51
Figura 27: Espaço ocupado sem o arquivo .keras e sem as amostras de sinais....	58
Figura 28: Diagrama de caso de uso.....	62
Figura 29: Diagrama de atividades.....	64
Figura 30: Diagrama de sequências.....	65
Figura 31: Pontos do corpo.....	66

Figura 32: Pontos da mão	66
Figura 33: Estrutura do arquivo .h5 I	68
Figura 34: Estrutura do arquivo .h5 II	69
Figura 35: Como ficam salvos os valores de peso e bias	69

LISTA DE TABELAS

Tabela 1: Participantes do projeto.....	34
Tabela 2: Estrutura da pilha de dados I	67
Tabela 3: Exemplo de como se parece essa pilha de dados	67

LISTA DE SIGLAS

AEDB	Associação Educacional Dom Bosco
ABNT	Associação Brasileira de Normas Técnicas
IBGE	Instituto Brasileiro de Geografia e Estatística
RJ	Rio de Janeiro
ONU	Organização das Nações Unidas
ODS	Objetivos de Desenvolvimento Sustentável
AI	Inteligência Artificial
NEAT	<i>NeuroEvolution of Augmenting Topologies</i>
IDE	<i>Integrated Development Environment</i> (Ambiente de Desenvolvimento Integrado)
iOS	<i>iPhone Operational System</i>
AI	<i>Artificial Intelligence</i>
LSTM	Long Short-Term Memory
TALIA	Tradutor Automático de Libras com Inteligência Artificial
AWS	<i>Amazon Web Services</i>
EC2	<i>Elastic Cloud Computing</i>
TCC	Trabalho de Conclusão de Curso
CPU	<i>Central Processing Unit</i>
RAM	<i>Random Access Memory</i>
PDF	<i>Portable Document Format</i>

SUMÁRIO

1.	INTRODUÇÃO	16
2.	FUNDAMENTAÇÃO TEÓRICA.....	17
2.1.	O que é a Libras, sua história e outros aspectos.....	17
2.1.2.	História e evolução da Linguagem de sinais:	18
2.1.3.	Linguagem de sinais ao redor do mundo:	18
2.1.4.	Importância da Libras:	20
2.1.5.	Aspectos linguísticos da Libras:	21
2.1.6.	Legislação e políticas públicas relacionadas à Libras:	22
2.2.	O que é a inteligência artificial, sua história e outros aspectos	23
2.2.1.	O que é inteligência artificial e a sua história	23
2.2.2.	Principais Inteligências Artificiais da atualidade:	23
2.2.3.	Conceitos da Inteligência Artificial	24
2.2.3.1.	Tipos de aprendizado	25
2.2.3.2.	Conjuntos de dados.....	25
2.2.3.3.	Outros conceitos de IA	26
2.2.3.4.	Conceitos de Rede Neural	27
2.2.3.5.	Camadas da rede neural	29
2.3.	Visão computacional dentro do projeto.....	29
2.4.	AGENDA 2030	31
3.	ESPECIFICAÇÃO DO SISTEMA	33
3.1.	DESCRIÇÃO DO(S) PROBLEMA(S).....	33
3.2.	PROPOSTA DE SOLUÇÃO	33
3.3.	PARTICIPANTES DO PROJETO	33
3.4.	USUÁRIOS PARTICIPANTES (ATORES)	34
3.5.	NECESSIDADES DOS USUÁRIOS	34
3.6.	REQUISITOS FUNCIONAIS	35
3.7.	REQUISITOS NÃO FUNCIONAIS.....	35
3.8.	ARQUITETURA ESTRUTURAL DO SISTEMA.....	36
3.9.	LOGOTIPO.....	36
3.10.	TECNOLOGIAS E FERRAMENTAS UTILIZADAS.....	37
3.10.1.	PYTHON	37
3.10.2.	TENSORFLOW	38

3.10.3.	Visual Studio Code	39
3.10.5.	MediaPipe	40
3.10.6.	Keras	41
3.10.7.	Pytsx3.....	41
3.10.8.	Flask.....	42
3.10.9.	Scikit-learn.....	42
3.10.10.	AMAZON EC2 – Elastic Compute Cloud.....	43
3.10.11.	Lubuntu	44
3.11.	DEPENDÊNCIAS	44
3.11.1.	MANUAL DO USUÁRIO	45
3.12.	REFERÊNCIAS (PARA O LEVANTAMENTO INICIAL).....	45
3.13.	APROVAÇÕES	45
4.	ESTRATÉGIAS DE RISCO	46
4.1.	LISTA DE RISCOS	46
4.2.	PRIORIZAÇÃO DOS RISCOS	46
4.4.	PLANOS DE CONTINGÊNCIA.....	47
5.	GERENCIAMENTO DE CONFIGURAÇÃO	49
5.1.	PLANO DE GERENCIAMENTO DE CONFIGURAÇÃO	49
5.2.	REPOSITÓRIO.....	49
5.2.1.	Notion	49
5.2.2.	GitHub	50
5.2.3.	Google Drive:.....	50
6.	ESTRATÉGIA DE TESTES.....	52
6.1.	PLANO DE TESTES.....	52
6.2.	TESTES DE UNIDADE AUTOMATIZADOS	52
6.3.	TESTES DE VALIDAÇÃO	53
6.4.	ACEITES DOS MÓDULOS PELOS STAKEHOLDER	55
7.	ESTRATÉGIA DE IMPLANTAÇÃO E SUPORTE	56
7.1.	NECESSIDADES DE IMPLANTAÇÃO	56
7.1.1.	ARQUITETURA DE IMPLANTAÇÃO	56
7.1.2.	CONFIGURAÇÃO DOS SERVIDORES	57
7.1.4.	INFRAESTRUTURA NECESSÁRIA.....	57
7.2.	CRONOGRAMA DE TREINAMENTOS.....	58

8. CONCLUSÃO	59
APÊNDICE A: DIAGRAMA DE CASOS DE USO.....	62
APÊNDICE B: DESCRIÇÕES DE CASOS DE USO	62
APÊNDICE D: DIAGRAMA DE ATIVIDADES	64
APÊNDICE E: DIAGRAMA DE SEQUÊNCIA.....	65
CDU 01: Traduzir um sinal de Libras.....	65
APÊNDICE G: DICIONÁRIO DE DADOS.....	66
ARQUIVOS .npy.....	66
ARQUIVO .h5	68
MANUAL DO USUÁRIO.....	71

1. INTRODUÇÃO

Com grande satisfação, apresentamos o tema do nosso trabalho de conclusão de curso, intitulado "TALIA". Este trabalho possui grande relevância para a nossa formação acadêmica e para a sociedade em geral, especialmente para a comunidade surda.

Atualmente, a tecnologia tem evoluído rapidamente, e a inteligência artificial tem sido cada vez mais utilizada para resolver problemas do dia a dia, inclusive no contexto da acessibilidade e inclusão social. A Língua Brasileira de Sinais (Libras) é uma língua oficial do Brasil, e sua utilização é fundamental para a comunicação e interação social de pessoas surdas. No entanto, muitas vezes, a comunicação entre surdos e ouvintes é difícil, o que pode gerar obstáculos na inclusão social.

Diante desse contexto, o objetivo deste trabalho é desenvolver um sistema de reconhecimento de sinais em Libras para assistência virtual, que possa facilitar a comunicação entre surdos e ouvintes, e contribuir para a inclusão social da comunidade surda. A ideia é que a assistência virtual possa compreender e interpretar perguntas e comandos em Libras e fornecer respostas apropriadas em tempo real, sem a necessidade de um intérprete humano.

2. FUNDAMENTAÇÃO TEÓRICA

2.1. O que é a Libras, sua história e outros aspectos

2.1.1. Principais medidas de acessibilidade já criadas:

Linguagem de sinais: A Libras é reconhecida como a língua oficial das pessoas surdas no Brasil, e é uma medida de acessibilidade que permite a comunicação em seu idioma natural. A lei brasileira estabelece que a Libras seja ensinada como disciplina curricular obrigatória nos cursos de formação de professores e de fonoaudiologia, e também deve ser disponibilizada em eventos e programas governamentais.

Legendas em programas de televisão e vídeos online: A disponibilidade de legendas em programas de televisão, filmes e vídeos online é uma medida de acessibilidade importante para pessoas surdas. Isso permite que elas acompanhem o conteúdo audiovisual por meio de texto, garantindo seu acesso à informação e entretenimento.

Alertas sonoros visuais: Alarmes e outros alertas sonoros podem ser adaptados para incluir componentes visuais, como luzes estroboscópicas ou sinais luminosos, para que pessoas surdas possam ser alertadas em emergências, como incêndios, terremotos ou evacuações.

Sinalização visual: Sinalização visual, como placas e avisos com letras grandes e contrastantes, é uma medida de acessibilidade que pode ser especialmente útil para pessoas surdas. Isso permite que elas obtenham informações importantes em locais públicos, como aeroportos, estações de trem e rodoviárias, e outros ambientes onde a comunicação sonora pode ser difícil.

Atendimento em Libras em serviços públicos e privados: A disponibilidade de atendimento em Libras em serviços públicos e privados, como hospitais, bancos, lojas e repartições governamentais, é uma medida de acessibilidade que permite que pessoas surdas se comuniquem em sua língua natural. Isso pode ser feito por meio de intérpretes de Libras presenciais ou por vídeo, garantindo que pessoas surdas tenham acesso a serviços essenciais.

Tecnologia assistiva: Existem várias tecnologias assistivas disponíveis para pessoas surdas, como aparelhos auditivos, implantes cocleares, dispositivos de vibração e aplicativos de tradução de Libras em tempo real, que podem facilitar a comunicação e a participação social de pessoas surdas.

Essas são apenas algumas das principais medidas de acessibilidade já criadas para pessoas surdas. É importante ressaltar que a acessibilidade é um direito fundamental e que a adoção de medidas de acessibilidade é essencial para garantir a inclusão plena e igualitária das pessoas surdas na sociedade.

2.1.2. História e evolução da Linguagem de sinais:

A história da Libras remonta ao século XIX, quando a primeira escola para surdos foi fundada no Brasil, introduzindo a língua de sinais francesa.

Ao longo do tempo, a Libras evoluiu e se desenvolveu, incorporando elementos da cultura e da língua brasileira, e se consolidando como uma língua própria utilizada pela comunidade surda brasileira.

A conquista do reconhecimento oficial da Libras como língua oficial do Brasil em 2002 e a promulgação da Lei Brasileira de Inclusão da Pessoa com Deficiência em 2015 representam marcos importantes na luta pela inclusão linguística e social da comunidade surda.

Apesar dessas conquistas, como é possível ver no gráfico do IBGE abaixo, muitos deficientes auditivos não utilizam a Libras, preferindo fazer a reabilitação com aparelhos auditivos ou até mesmo utilizar linguagens de sinais menos comuns. Dependendo do grau da deficiência alguns ainda tentam conviver com a deficiência auditiva sem tentar aprender nenhuma forma alternativa de comunicação.

2.1.3. Linguagem de sinais ao redor do mundo:

Figura 1: Línguas de sinais pelo mundo



Línguas de sinais usadas em diferentes países do mundo. SignumWeb Comunicação Inclusiva. Disponível em: <<https://blog.signumweb.com.br/curiosidades/linguas-de-sinais-pelo-mundo/>>. Acesso em: 24 nov. 2023

As línguas de sinais são línguas naturais, assim como o português. Elas são aprendidas de forma natural pelo surdo, desde que ele seja exposto a essa forma de comunicação desde cedo. As línguas de sinais também possuem códigos e regras, como qualquer outra língua.

Se uma pessoa surda for visitar outro país, ela terá que aprender ou ao menos ter uma noção básica da língua de sinais do local. Isso ocorre porque as línguas de sinais são diferentes de país para país, da mesma forma que as línguas orais. No Brasil, a Libras é a língua de sinais oficial da comunidade surda, em Portugal, a LGP é a língua de sinais oficial. Apesar de compartilharem semelhanças na língua portuguesa, as duas línguas de sinais apresentam diferenças.

Outros exemplos de línguas de sinais ao redor do mundo são:

- LAS (Língua Angolana de Sinais) - Angola
- DTS (*Kalaallisut Ussersuutit*) - Dinamarca
- ASL (*American Sign Language*) - Estados Unidos
- LSB (*Lenguaje de Señas Bolivianas*) - Bolívia
- LSC (*Lengua de Señas Colombiana*) - Colômbia
- LSQ (*Langue des Signes Québécoise*) - Quebec / Canadá
- LSM (*Lengua de signos mexicano*) - México

BIM (*Bahasa Isyarat Malaysia*) - Malásia

Монгол дохионы хэл - Mongólia

ÖGS - *Österreichische Gebärdensprache* - Áustria

Lughat al-Isharat al-Lubnaniya - Líbano

É interessante notar que, da mesma forma que a língua oral apresenta diferenças regionais e dialetos, a língua de sinais também é influenciada por essas alterações, existindo palavras que em certos estados do próprio Brasil são representadas por um sinal, enquanto em outros são representadas por outro, Ex:

Figura 2: Diferentes sinais para "verde" dependendo do estado do Brasil



Fonte: EducaLibras. **Regionalismo da Libras: A diversidade da língua de sinais brasileira.** LinkedIn. Disponível em: <<https://pt.linkedin.com/pulse/regionalismo-da-libras-diversidade->>. Acesso em: 24 set. 2023

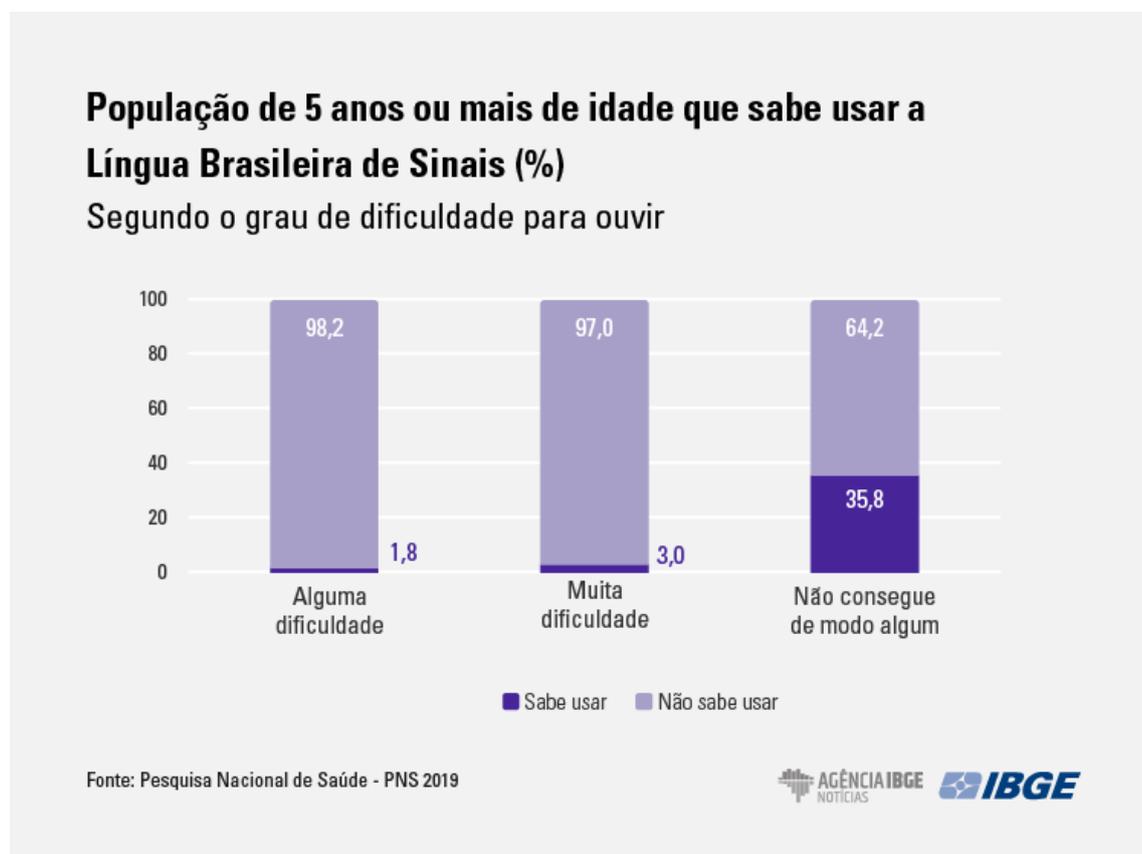
2.1.4. Importância da Libras:

A Libras é fundamental para a comunicação e interação da comunidade surda, permitindo o pleno exercício de seus direitos e acesso a informações, educação, trabalho, cultura e lazer.

A Libras é uma forma legítima de expressão linguística e cultural, com valor social e identitário para a comunidade surda, que possui sua própria cultura, história e tradições.

A promoção do uso e reconhecimento da Libras contribui para a inclusão social, o respeito à diversidade e o exercício da cidadania das pessoas surdas.

Figura 3: Gráfico de pessoas que sabem Libras com base no grau de dificuldade de ouvir



Fonte: Crônicas da surdez. **7 Coisas que você não sabia sobre a língua de sinais.** A maioria dos surdos NÃO usa Libras. Disponível em: <<https://cronicasdasurdez.com/coisas-sobre-libras/>>. Acesso em: 09 set. 2023.

2.1.5. Aspectos linguísticos da Libras:

A Libras possui uma gramática visual-espacial, diferindo em muitos aspectos da gramática das línguas faladas.

Na Libras, a informação é transmitida por meio de configuração das mãos, movimentos, orientação espacial, expressões faciais e corporais.

A Libras possui sua própria estrutura sintática, com regras específicas para formação de frases, questionamentos, negações, entre outros.

Assim como nas línguas faladas, a Libras também possui variações regionais, com diferenças na fonologia, léxico e expressões culturais, devido à diversidade cultural do Brasil.

2.1.6. Legislação e políticas públicas relacionadas à Libras:

No Brasil, a Libras é protegida por diversas legislações e políticas públicas, incluindo a Constituição Federal, a Lei Brasileira de Inclusão da Pessoa com Deficiência, o Decreto nº 5.626/2005 que regulamenta a Libras, e outras iniciativas voltadas para a promoção da acessibilidade comunicacional e inclusão da comunidade surda. Essas legislações e políticas públicas têm como objetivo garantir a igualdade de direitos.

(Nacional, 1996)

2.2. O que é a inteligência artificial, sua história e outros aspectos

2.2.1. O que é inteligência artificial e a sua história

A Inteligência Artificial (IA) é um campo de estudo que tem como objetivo criar sistemas capazes de realizar tarefas que, até então, eram exclusivas do ser humano, como reconhecimento de fala, processamento de imagem, tomada de decisão, entre outras.

As ideias de IA surgiram no início do século XX, mas foi durante a Segunda Guerra Mundial que a pesquisa nessa área ganhou força, com estudos sobre redes neurais e sistemas de raciocínio artificial. Em 1950, Claude Shannon escreveu sobre como programar uma máquina para jogar xadrez, e Alan Turing criou o famoso "Teste de Turing" para avaliar se uma máquina poderia se passar por humano em uma conversa por escrito.

O marco-zero da IA, mais próxima do que conhecemos atualmente, ocorreu em 1956 com a Conferência de Dartmouth, que reuniu várias personalidades da área. A partir dessa conferência, o campo de pesquisa da IA foi definido, com o objetivo de criar sistemas capazes de simular a inteligência humana.

Desde então, instituições privadas e governamentais investem na área de IA levando a avanços significativos em reconhecimento de fala, processamento de imagem, aprendizado de máquina e outras áreas. Hoje em dia, a IA é usada em uma ampla variedade de aplicações, desde assistentes pessoais até carros autônomos e sistemas de diagnóstico médico.

2.2.2. Principais Inteligências Artificiais da atualidade:

Muitas pessoas têm falado sobre o ChatGPT, Bing e Bard, porém existem várias outras IAs para diversas funcionalidades, como geração de imagens, criação de vídeos, criação de músicas, criação de slides, formatação de texto, na imagem abaixo podemos ver os mais diversos exemplos:

2.2.3.1. Tipos de aprendizado

É muito importante que na hora que você for programar sua Inteligência Artificial já tenha sido definido o tipo de treinamento que irá aplicar, pois podem ser usadas diferentes técnicas de aprendizado para obter diferentes resultados. Abaixo estão alguns tipos de aprendizado:

Aprendizado supervisionado: é uma abordagem de aprendizado em que o algoritmo é treinado com exemplos rotulados e, em seguida, usa esses rótulos para fazer previsões sobre novos dados. Ex: Regressão linear, árvores de decisão e redes neurais.

Aprendizado não supervisionado: é uma abordagem de aprendizado em que o algoritmo é treinado com dados não rotulados e tenta encontrar padrões ou estrutura nos dados. Ex: Agrupamento, detecção de anomalias, Kmeans e redução de dimensionalidade.

Aprendizado por reforço: é uma abordagem de aprendizado em que o algoritmo é treinado para tomar ações em um ambiente e receber recompensas ou penalidades com base em suas ações. O objetivo é maximizar a recompensa cumulativa ao longo do tempo. Ex: NEAT, agente de Markov e o algoritmo Q-Learning

2.2.3.2. Conjuntos de dados

Estando a base de dados criada, você deverá que dividir em 2 ou 3 partes, sendo que cada uma dessas partes desempenhará um papel. Você pode ou não separar um conjunto de validação, mas é importante que você separe seus dados pelo menos em dados de teste e treinamento.

É recomendável que você use pelo menos 70% dos seus dados para compor seu conjunto de treinamento, pois senão sua IA pode não ser apresentada a certos casos durante o treinamento. Em outra parte, é importante que o conjunto de testes também não seja tão pequeno, ou então você conseguirá comprovar direito se sua IA está funcionando conforme o esperado.

Conjunto de treinamento: é um conjunto de exemplos rotulados usados para treinar um algoritmo de Machine Learning, ele deve

Conjunto de validação: é um conjunto de exemplos rotulados usados para ajustar os hiperparâmetros de um modelo de Machine Learning e avaliar seu desempenho durante o treinamento.

Conjunto de teste: é um conjunto de exemplos rotulados usados para avaliar o desempenho final de um modelo de Machine Learning após o treinamento.

2.2.3.3. Outros conceitos de IA

Função de perda: é uma medida da diferença entre as previsões de um modelo de Machine learning e as respostas corretas. O objetivo do treinamento é minimizar a função de perda.

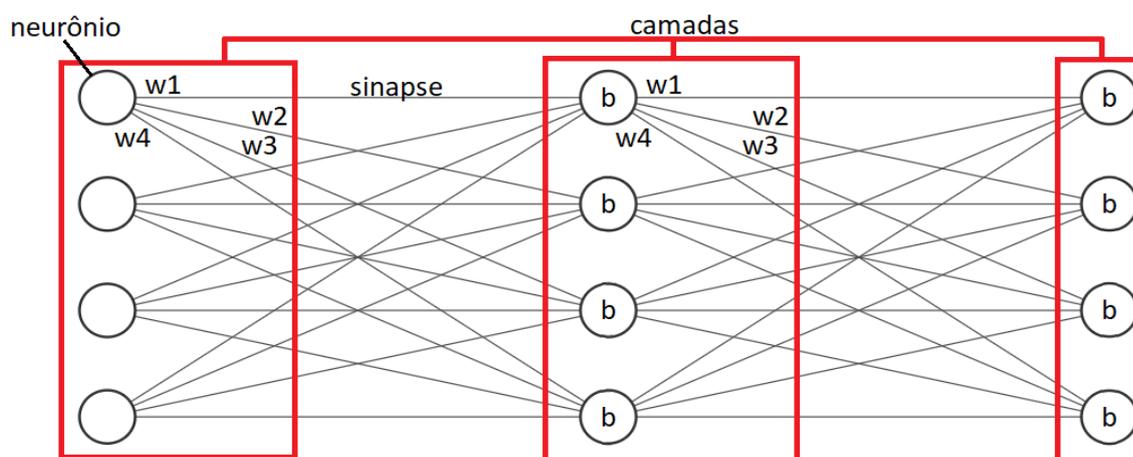
Hiperparâmetros: são parâmetros do modelo de Machine Learning que não são aprendidos durante o treinamento, mas que afetam o desempenho do modelo. Exemplos incluem a taxa de aprendizado, o número de camadas em uma rede neural e o número de árvores em um algoritmo de floresta aleatória.

Overfitting: é um problema em que um modelo de Machine Learning se ajusta muito bem aos dados de treinamento e não generaliza bem para novos dados. Isso pode ocorrer quando o modelo é muito complexo em relação ao tamanho do conjunto de dados de treinamento.

Underfitting: é um problema em que um modelo de Machine Learning é muito simples e não consegue capturar toda a complexidade dos dados. Isso pode ocorrer quando o modelo é muito restrito em relação ao tamanho do conjunto de dados de treinamento.

2.2.3.4. Conceitos de Rede Neural

Figura 5: Partes de uma rede neural



Fonte: Elaboração própria baseado na rede neural que pode ser vista neste endereço (<https://aws.amazon.com/pt/what-is/neural-network/>)

Época: É cada repetição onde todas as amostras de treino são mostradas para a rede neural;

Neurônio (círculo): cada setor da rede neural que recebe uma entrada e retorna uma saída, cada neurônio possui um valor para b ;

Sinapse (linha): Conexão entre dois neurônios, cada sinapse possui um valor para w ;

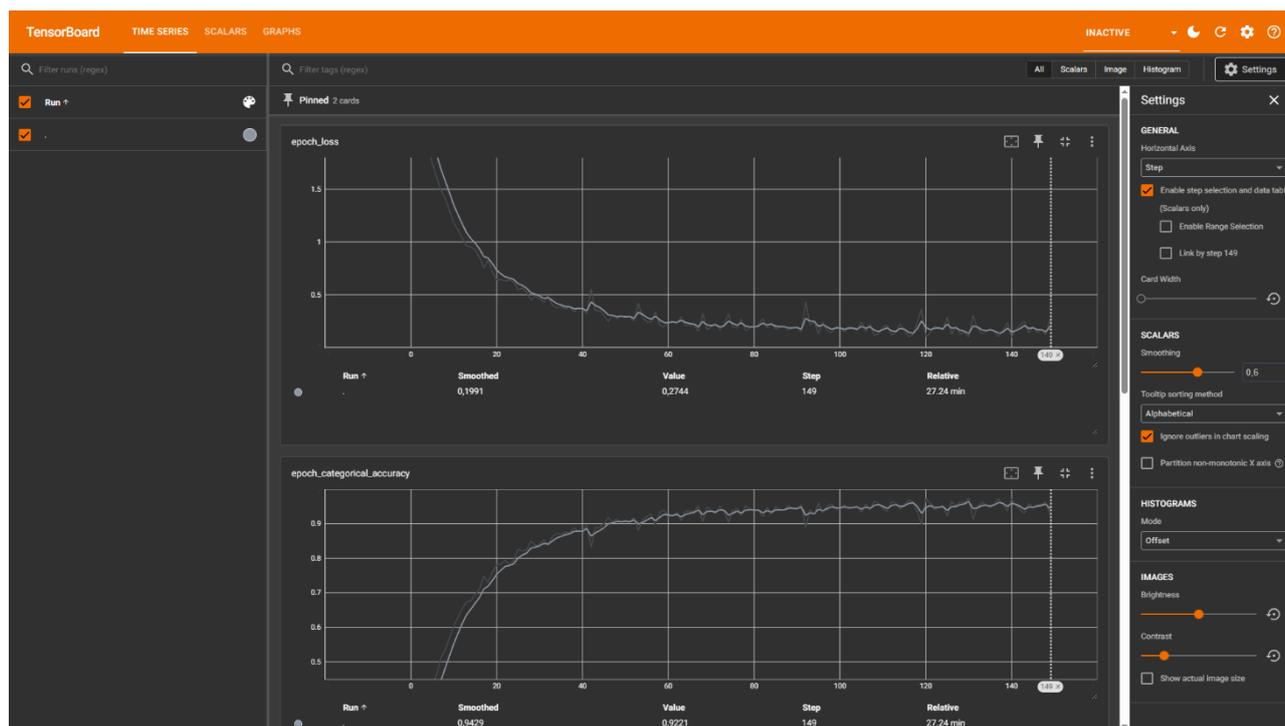
Camada (coluna): Coluna de neurônios onde todos os neurônios contidos nela possuem as mesmas pré-configurações, e nenhum neurônio dela interage entre si;

Peso (w): Cada sinapse possui um peso, que indica por quanto serão multiplicados os valores que vieram do neurônio anterior para o próximo. O valor do peso é definido durante o treinamento, tendo sido aleatoriamente estimado pelo Keras ao longo das épocas a cada vez que a rede era exposta a cada amostra, até ser encontrado o peso que considerando a estrutura da rede inteira fizesse a rede retornar a saída mais próxima da ideal para todas as entradas ensinadas.

Bias (b): Cada neurônio possui seu próprio valor de *bias*, que indica quanto será somado ao resultado da conta (valor que entrou no neurônio * peso). O valor do *bias* é definido

durante o treinamento, tendo sido estimado aleatoriamente pelo Keras ao longo das épocas a cada vez que a rede era exposta a cada amostra, até ser encontrado o *bias* que considerando a estrutura da rede inteira fizesse a rede retornar a saída mais próxima da ideal para todas as entradas ensinadas.

Figura 6: Loss e Accuracy ao longo do treinamento



Fonte: Captura de tela tirada de um gráfico do Tensorboard durante o treinamento da rede neural da TALIA

Loss: é uma métrica que diz o quão longe as previsões da IA estão da resposta certa para cada configuração de pesos e *bias*, sendo que o objetivo do Keras durante o treinamento da rede é sempre estimar os pesos e *bias* que façam a IA trazer previsões cada vez mais próximas das corretas, diminuindo assim o *Loss* ao longo do treinamento.

Categorical Accuracy: é uma métrica que mede o quanto de padrão a rede neural é capaz de entender com a configuração atual de pesos e *bias*, para ter um exemplo, em uma rede que deseja diferenciar imagens de cães e de gatos, com um categorical accuracy baixo a IA seria capaz apenas de perceber que os dois tem pelos e 4 patas mas não seria capaz de identificar maiores diferenças. O objetivo do Keras durante o treinamento da rede é sempre estimar os pesos e *bias* que façam a porcentagem identificar melhor os padrões entre os dados passados, aumentando assim o *Categorical accuracy* ao longo do treinamento.

Conforme descrito, no gráfico acima, é possível ver claramente o *loss* decaindo e o *categorical accuracy* subindo ao longo das épocas, até que ambos começam a estabilizar na época 40 e atingem um platô sem grandes mudanças na época

2.2.3.5. Camadas da rede neural

Para construir a rede neural de nossa IA, utilizamos 2 tipos de camadas:

Camada densa: Que é uma camada onde todos os neurônios de uma camada se conectam em todos os neurônios da próxima.

Camada LSTM: LSTM significa “*Long-short Term Memory*” (Memória de longo de curto prazo) e ela possui algumas configurações que diferenciam ela dos outros tipos de camada, que possui dificuldade para manter conhecimento de longo prazo.

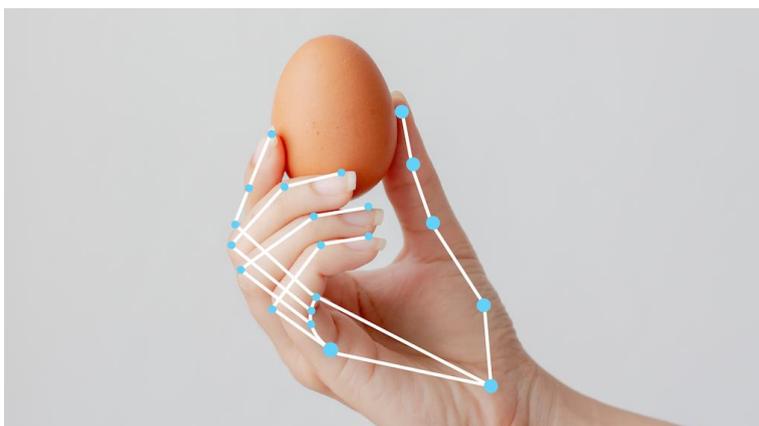
As camadas convencionais geralmente possuem um único valor de Bias e um único peso para cada sinapse, e esse Bias e peso vão se otimizando conforme a rede é exposta a novas amostras, essa configuração possui uma falha, se a rede começa a se otimizar para um certo conjunto de amostras, e no final do conjunto aparecem amostras muito diferentes, essas amostras correm o risco de fazer a rede “esquecem” o que ela aprendeu com as primeiras amostras. Camadas LSTM possuem um número maior de bias e pesos e uma configuração que permite a ela não esquecer totalmente de uma informação mais antiga.

2.3. Visão computacional dentro do projeto

Para fazer o reconhecimento dos sinais dos usuários utilizamos basicamente 2 funcionalidades diferentes do Mediapipe combinadas:

O reconhecimento das mãos (que se baseia em 21 pontos de cada mão), esta é a principal referência da nossa inteligência artificial na hora de reconhecer o sinal que está sendo feito, embora na maioria dos sinais a posição dos braços com a relação ao restante do corpo influencie muito.

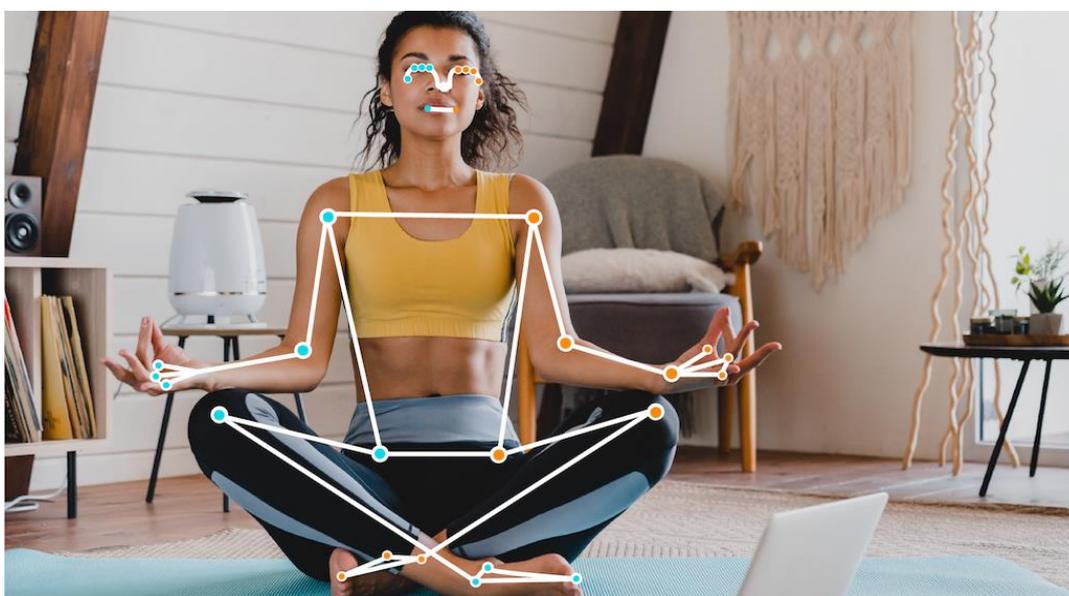
Figura 7: Mediapipe: Detecção de pontos das mãos



Fonte: MediaPipe. **Detecção de pontos das mãos**. Disponível em: https://developers.google.com/mediapipe/solutions/vision/hand_landmarker. Acesso em: 09 abr. 2023

O reconhecimento da pose (que se baseia em 33 pontos do corpo), como dito anteriormente diversos sinais da Libras dependem de como os braços estão posicionados com relação ao restante do corpo, então esta fica sendo a segunda referência principal da nossa inteligência artificial.

Figura 8: Mediapipe: Detecção de pontos do corpo



Fonte: MediaPipe. **Detecção de pontos do corpo**. Disponível em: https://developers.google.com/mediapipe/solutions/vision/pose_landmarker. Acesso em: 09 abr. 2023.

2.4. AGENDA 2030

Esse projeto pode contribuir para a implementação de diversos Objetivos de Desenvolvimento Sustentável (ODS) estabelecidos pela Organização das Nações Unidas (ONU).

Primeiramente, o projeto pode contribuir para o ODS 3 - Saúde e Bem-Estar, ao promover a inclusão social e a acessibilidade para pessoas com deficiência auditiva. Além disso, a tecnologia pode ser utilizada para ajudar pessoas com deficiência auditiva a acessarem serviços de saúde, garantindo o direito à saúde para todos.

Figura 9: Agenda 2030: Saúde e bem-estar



Fonte: **Agenda 2030 para o Desenvolvimento Sustentável | As Nações Unidas no Brasil, 2023**. Disponível em <<https://brasil.un.org/pt-br/91863-agenda-2030-para-o-desenvolvimento-sustent%C3%A1vel>>. Acesso em: 18 jun. 2023.

O projeto também pode contribuir para o ODS 4 - Educação de Qualidade, ao ajudar a superar as barreiras educacionais enfrentadas por pessoas com deficiência auditiva. Com a tecnologia de reconhecimento de sinais em Libras, é possível tornar a educação mais acessível e inclusiva para essas pessoas.

Figura 10: Agenda 2030: Educação de qualidade



Fonte: **Agenda 2030 para o Desenvolvimento Sustentável | As Nações Unidas no Brasil, 2023**. Disponível em <<https://brasil.un.org/pt-br/91863-agenda-2030-para-o-desenvolvimento-sustent%C3%A1vel>>. Acesso em: 18 jun. 2023.

Além disso, o projeto pode contribuir para o ODS 10 - Redução das Desigualdades, ao promover a inclusão social e a acessibilidade para pessoas com deficiência auditiva e ao reduzir as barreiras de comunicação que essas pessoas enfrentam. Dessa forma, é possível garantir que todas as pessoas tenham as mesmas oportunidades e possam exercer plenamente seus direitos.

Figura 11: Agenda 2030: Redução das desigualdades



Fonte: **Agenda 2030 para o Desenvolvimento Sustentável | As Nações Unidas no Brasil, 2023**. Disponível em <<https://brasil.un.org/pt-br/91863-agenda-2030-para-o-desenvolvimento-sustent%C3%A1vel>>. Acesso em: 18 jun. 2023

3. ESPECIFICAÇÃO DO SISTEMA

Como dito, nossa IA será capaz de reconhecer sinais de Libras e traduzi-los para o português. Isso ocorrerá da seguinte forma, o usuário irá acessar o sistema, uma vez dentro do sistema haverá uma biblioteca que irá reconhecer os movimentos do usuário por meio da câmera do notebook ou webcam e enviará quais os movimentos foram identificados para a IA propriamente dita. Essa IA será responsável por interpretar os sinais e dizer o seu significado dentro da Libras, por fim, ela irá retornar o significado na forma de texto para o sistema, que fará o uso dessa vez do Pyttsx3 (biblioteca do Python responsável por converter texto em áudio) para exibir esse significado em forma de áudio, lembrando que o sistema também exibirá a tradução escrita para os deficientes auditivos que não souberem a Libras.

3.1. DESCRIÇÃO DO(S) PROBLEMA(S)

A dificuldade dos surdos de se comunicarem de igual para igual com ouvintes, visto que a linguagem humana é quase toda sonora. Os surdos possuem dificuldades para ingressarem no mercado de trabalho, pois muitas empresas não dão tanto foco ao surdo e nem tem uma intenção real de contratá-lo, pois caso contratem um surdo teriam que fornecer um (a) intérprete, visto que é uma obrigação, conforme mencionado no tópico 2.2.5 da fundamentação teórica.

3.2. PROPOSTA DE SOLUÇÃO

Para tentar amenizar o problema dito acima, nós pensamos em criar um sistema de inteligência artificial capaz de reconhecer sinais de Libras, e exibir o seu significado em forma de áudio e texto para o usuário que não entende Libras, tudo isso em tempo real.

T.A.L.I.A é um acrônimo para Tradutor Automático de Libras com Inteligência Artificial, que é basicamente a funcionalidade principal do projeto.

3.3. PARTICIPANTES DO PROJETO

Os participantes do projeto são os desenvolvedores e o orientador.

Tabela 1: Participantes do projeto

Nomes	Atividades	Responsável
Carlos Henrique Barros Silva Campos	Desenvolvedor back end	Responsável por auxiliar no desenvolvimento do Back-end e da interface. Além de ajudar no treinamento da IA
Pedro Henrique Correa da Rocha	Desenvolvedor back end	Responsável por auxiliar no desenvolvimento do Back-end e na configuração da rede neural. Além de ajudar no treinamento
Rafael Henrique Nogueira Alves	Documentação e pesquisas	Responsável por parte da documentação e pesquisas sobre o projeto.
Victor Hugo Pacheco de Lima	Desenvolvedor front end e testador	Responsável por auxiliar no desenvolvimento do front end do projeto e pelos treinamentos e testes
Bruno Saldanha Sampaio	Orientador	Avaliar os processos e orientar nas possíveis mudanças do projeto

Fonte: Elaboração própria.

3.4. USUÁRIOS PARTICIPANTES (ATORES)

Os atores principais do projeto TALIA são as pessoas com deficiência auditiva e mudez que usam Libras como forma de comunicação. Elas são as principais usuárias da tecnologia, e sua participação é essencial para garantir que a TALIA seja acessível e inclusiva

Para garantir a participação dessas pessoas no projeto, é importante criar oportunidades para que elas possam fornecer feedback sobre o desenvolvimento da TALIA. Isso pode ser feito através de grupos de foco, testes de usabilidade e o desenvolvimento de materiais de treinamento. Por meio da participação desses atores, além de descobrir onde a TALIA está errando, também poderemos descobrir quais sinais são importantes e ainda faltam serem ensinados para ela.

3.5. NECESSIDADES DOS USUÁRIOS

Embora o ator principal seja o usuário de Libras, o principal usuário que se beneficiaria do bom funcionamento da TALIA serão as pessoas que não entendam Libras, e por conta disso o que mais vai importar essas pessoas serão: velocidade da tradução, se o sinal foi traduzido de forma correta e se a TALIA já está treinada com os sinais que eles precisam entender.

3.6. REQUISITOS FUNCIONAIS

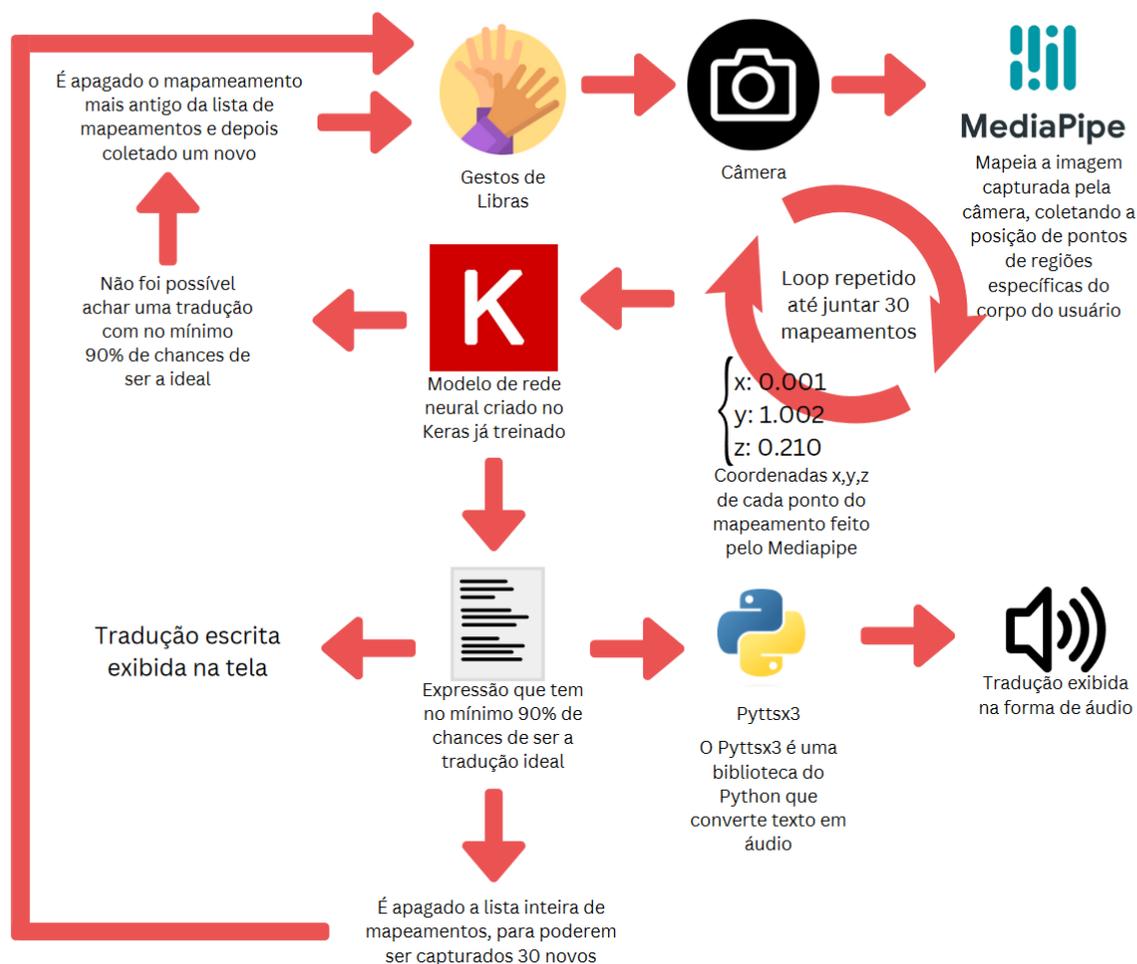
- RF01 – Reconhecimento e tradução de sinais estáticos da Libras;
- RF02 – Reconhecimento e tradução de sinais da Libras com movimento;
- RF03 – Tradução exibida tanto em voz quanto em texto;
- RF04 – Tradução de texto digitado para Libras;
- RF05 – Histórico de traduções já feitas.

3.7. REQUISITOS NÃO FUNCIONAIS

- RNF01 – Tradução precisa e em tempo real;
- RNF02 – O sistema não terá utilidade caso os dois usuários não saibam Libras;
- RNF03 – O usuário precisa de um computador com câmera para rodar o sistema;
- RNF04 – O usuário precisa ter acesso à internet em qualquer dispositivo que possua câmera;
- RNF05 – Facilidade de manutenção;
- RNF06 – Interface de usuário intuitiva.

3.8. ARQUITETURA ESTRUTURAL DO SISTEMA

Figura 12: Arquitetura do sistema



Fonte: Elaboração própria.

3.9. LOGOTIPO

Escolhemos esse logótipo pois queríamos algo que fosse ao mesmo tempo minimalista e que conseguisse passar uma ideia de acessibilidade, tanto é que o sinal retratado no logotipo é justamente o sinal que nas libras quer dizer “Acessibilidade”, que é usado em vários eventos e lugares onde a tradução para libras está presente.

As cores preta e laranja foram escolhidas por um motivo, a cor laranja significa a sociabilidade do ato de se comunicar e o preto significa a seriedade do projeto.

Figura 13: Logotipo do projeto



Fonte: Elaboração própria

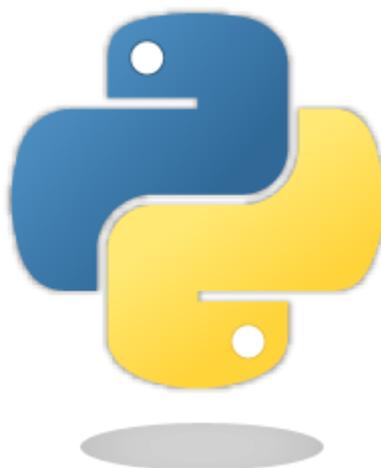
3.10. TECNOLOGIAS E FERRAMENTAS UTILIZADAS

3.10.1. PYTHON

Python é uma linguagem de programação de alto nível, interpretada, dinâmica e orientada a objetos. Sua sintaxe é relativamente simples e fácil de compreender, o que a torna uma opção popular entre desenvolvedores e profissionais de outras áreas, como engenheiros, matemáticos, cientistas de dados e pesquisadores.

O fato de possuir um grande número de biblioteca, nativas e desenvolvidas por outros desenvolvedores, torna-se muito útil em uma grande variedade de setores dentro de desenvolvimento web, e também em áreas como análise de dados, Machine Learning e simulação.

Figura 14: Python



Fonte: **Python**. Disponível em: <https://www.python.org>. Acesso em: 26 set. 2023

3.10.2. TENSORFLOW

O TensorFlow é uma biblioteca de software de código aberto para computação numérica através de grafos computacionais. Foi inicialmente desenvolvido pela equipe do Google Brain na organização de pesquisa Machine Intelligence do Google para aprendizado de máquina e pesquisa de redes neurais profundas (Deep Learning), mas sua ampla gama de recursos torna-a aplicável em uma grande variedade de outros domínios também.

Lançado como software livre em 2015, o TensorFlow alcançou a versão 1.0 em fevereiro de 2017 e desde então teve um desenvolvimento e adoção rápidos, com muitos colaboradores externos contribuindo para aprimorá-lo. Hoje em dia, o TensorFlow é considerado a biblioteca padrão para desenvolvimento em Deep Learning e aplicações de Inteligência Artificial.

Este artigo apresenta o TensorFlow, incluindo sua comunidade e ecossistema de software livre, demonstrando como a biblioteca tem sido amplamente adotada e contribuído para avanços significativos em aprendizado de máquina e Inteligência Artificial. Com o auxílio do TensorFlow, profissionais de várias áreas podem criar soluções mais eficientes e eficazes para uma variedade de problemas complexos.

Figura 15: TensorFlow

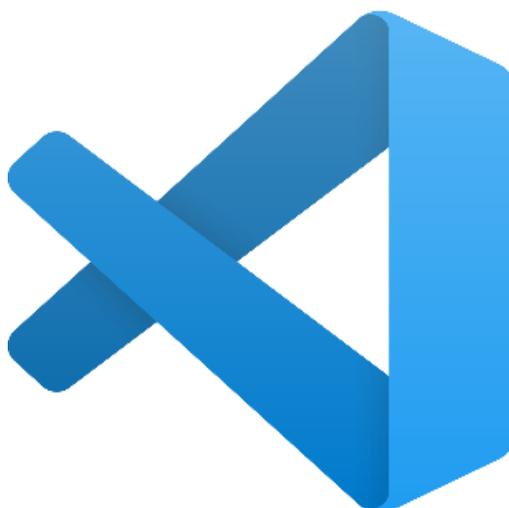


Fonte: **TensorFlow**. Disponível em: <https://www.tensorflow.org/>. Acesso em: 26 set. 2023

3.10.3. Visual Studio Code

O Visual Studio Code é um editor de código-fonte desenvolvido pela Microsoft para Windows, Linux e macOS. Ele inclui suporte para depuração, controle de versionamento Git incorporado, realce de sintaxe, complementação inteligente de código, snippets e refatoração de código.

Figura 16: Visual Studio Code

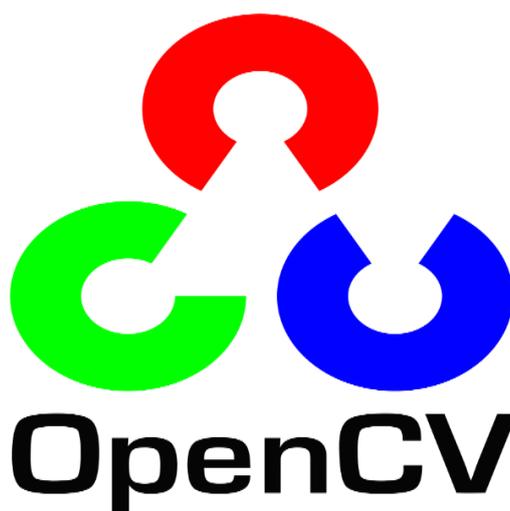


Fonte: **Visual Studio Code**. Microsoft Apps. Disponível em: <https://apps.microsoft.com/store/detail/visual-studio-code/XP9KHM4BK9FZ7Q?hl=pt-br&gl=br>. Acesso em: 26 set. 2023

3.10.4. OpenCV:

OpenCV (Open Source Computer Vision) é uma biblioteca de programação de código aberto, inicialmente desenvolvida pela Intel para tornar a visão computacional acessível a desenvolvedores e hobistas. Com mais de 500 funções disponíveis, ela pode ser usada em diversas linguagens de programação, incluindo C++, Python, Ruby e Java, para realizar uma variedade de análises em imagens e vídeos. Entre suas aplicações, destacam-se a detecção, rastreamento e reconhecimento facial, a edição de fotos e vídeos, a detecção e análise de textos, entre outras. Com a ajuda do OpenCV, é possível implementar algoritmos sofisticados de visão computacional de forma eficiente e com grande flexibilidade.

Figura 17: OpenCV



Fonte: **OpenCV University**. Disponível em: <https://opencv.org/resources/media-kit/>. Acesso em 26 set. 2023

3.10.5. MediaPipe

É um framework para construção de pipelines de aprendizado de máquina aplicadas, multimodais (como vídeo, áudio, dados de séries temporais etc.), multiplataforma (por exemplo, Android, iOS, Web e dispositivos de borda).

Figura 18: MediaPipe



Fonte: **MediaPipe**. MediaPipe | Google for Developers. Disponível em: <<https://developers.google.com/mediapipe/solutions>>. Acesso em: 26 set. 2023

3.10.6. Keras

O Keras é uma biblioteca de rede neural de código aberto escrita em Python. Ele é capaz de rodar em cima de TensorFlow, Microsoft Cognitive Toolkit, R, Theano, ou PlaidML. Projetado para permitir experimentação rápida com redes neurais profundas, ele se concentra em ser fácil de usar, modular e extensível.

Figura 19: Keras



Fonte: **Keras: Deep Learning for humans**. Disponível em: <https://keras.io>. Acesso em: 26 set. 2023.

3.10.7. Pyttsx3

É uma biblioteca Python que permite converter texto em fala.

Figura 20: Pyttsx3



Fonte: **Simple Text to Speech Demo in Python 3 (with pyttsx3)**. Disponível em: <https://www.geeks3d.com/hacklab/20220630/simple-text-to-speech-demo-in-python-3-with-pyttsx3/>. Acesso em: 26 set. 2023

3.10.8. Flask

Flask é um pequeno framework web escrito em Python. É classificado como um *microframework* porque não requer ferramentas ou bibliotecas particulares, mantendo um núcleo simples, porém, extensível.

Figura 21: Flask



Fonte: **Python API deployment with RStudio Connect: Flask**. Jumping rivers. Disponível em: <https://www.jumpingrivers.com/blog/python-api-deployment-rstudio-flask/>. Acesso em: 27 set. 2023

3.10.9. Scikit-learn

É uma biblioteca de Python que traz ferramentas auxiliares bem úteis durante o desenvolvimento de uma IA, como funções que ajudam a separar as amostragens entre

dados de treino e dados de teste e funções que permitem medir a eficiência da IA de forma automática

Figura 22: Scikit-learn

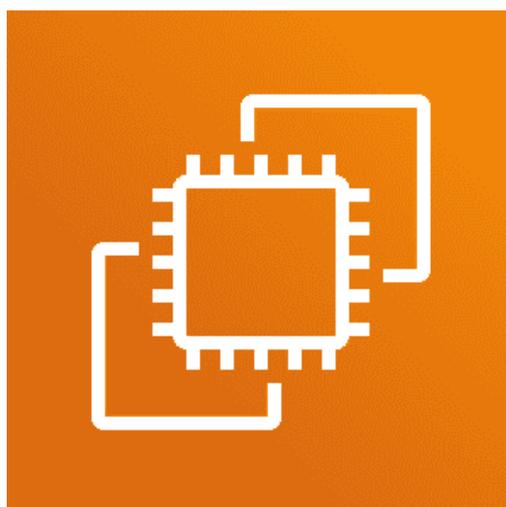


Fonte: **scikit-learn**. Wikipedia. Disponível em <https://en.wikipedia.org/wiki/Scikit-learn>. Acesso em: 27 set. 2023

3.10.10. AMAZON EC2 – Elastic Compute Cloud

Serviço da AWS que permite criar máquinas virtuais dotadas de elasticidade, ou seja, a capacidade de crescer ou diminuir conforme a demanda. Pode ser usado para criar servidores ou hospedar aplicações e deixar o *link* dessas aplicações disponível para acesso. Essas máquinas virtuais também possuem outras configurações que podem ser personalizadas pelo usuário.

Figura 23: Amazon EC2



Fonte: **How to change or upgrade an EC2 instance type**. Logicata. Disponível em: <https://www.logicata.com/blog/aws-ec2-change-instance-type/>. Acesso em: 27 set. 2023

3.10.11. Lubuntu

Distribuição do Linux bastante leve e que pode ser facilmente instalada em um pendrive, tornando-se um sistema operacional de bolso que pode ser usado em praticamente qualquer máquina, apenas plugando o pendrive.

Figura 24: Lubuntu



Fonte: **Lubuntu**. Wikipedia. Disponível em: <https://pt.m.wikipedia.org/wiki/Lubuntu>. Acesso em: 27 set. 2023

3.11. DEPENDÊNCIAS

HARDWARE: Para ser utilizado, nosso projeto precisa de um computador, notebook ou celular com câmera e com um acesso à internet estável. Se forem usadas câmeras com resolução muito baixa a IA pode ter dificuldades para detectar os movimentos e se a internet não for tão boa, a IA pode não conseguir reconhecer corretamente os sinais devido à latência.

SOFTWARE: Durante a fase de desenvolvimento, nosso projeto depende do terminal para rodar, uma vez finalizado ele poderá ser simplesmente acessado pelo *link*.

PEOPLEWARE: É necessário que pelo menos um dos usuários saiba Libras, ou então nossa ferramenta se tornará inútil.

3.11.1. MANUAL DO USUÁRIO

Hardware: Precisar ser capaz de suportar a execução do software de IA, com capacidade de processamento e armazenamento suficientes;

Câmera: Para que a IA de Libras possa interpretar a linguagem de sinais, é necessário ter uma câmera com qualidade mínima para que seja possível pelo menos reconhecer bem o corpo, as mãos e os dedos na imagem;

Internet: O usuário deverá ter acesso a internet.

3.12. REFERÊNCIAS (PARA O LEVANTAMENTO INICIAL)

As dificuldades enfrentadas pela comunidade surda incluem a falta de oportunidades de emprego e preconceito, além da falta de compreensão da sociedade em geral. A comunicação entre os usuários de Libras e os não usuários de Libras sempre foi uma situação complicada, uma vez que é necessário que os dois entendam a Libras para poderem conversar de forma prática, ou um deles pode precisar de um intérprete, que na maioria das vezes pode não estar disponível.

Pensando nisso, decidimos criar um tradutor capaz de interpretar a Libras por meio de uma simples câmera e um computador ou celular, facilitando assim tanto a comunicação entre deficientes auditivos que usam Libras com os que não usam, quanto a comunicação entre usuários de Libras e falantes.

3.13. APROVAÇÕES

As fases de aprovações serão marcadas por 3 seminários e pela apresentação final, nos intervalos desses marcos a TALIA será desenvolvida e serão realizados testes pelos próprios desenvolvedores do projeto, usuários, orientador do projeto e seminários ao decorrer do ano. Os desenvolvedores do projeto irão validar a técnicas utilizadas no desenvolvimento do projeto, serão realizados testes unitários e de interação com os usuários que fornecerão feedbacks sobre o projeto.

O orientador também avaliará cada etapa do projeto tendo acesso as documentações e testes do projeto, e irá verificar se o mesmo atende os requisitos propostos desde o início do ano.

No decorrer do ano será avaliado através de seminários, no qual apresentaremos nosso projeto a banca do TCC onde poderão ver o projeto em funcionamento e fornecerão feedbacks

4. ESTRATÉGIAS DE RISCO

4.1. LISTA DE RISCOS

Sinais ensinados errados: Esse problema está relacionado a imprecisões de tradução que podem ocorrer devido à rede neural ter sido treinada com amostras erradas ou amostras com pouca variação da posição da pessoa com relação a câmera, o que faz com a rede tenha dificuldades para reconhecer um sinal de longe quando todas as amostras passadas foram gravadas de perto;

Regionalismo: Esse problema está relacionado a variações dos sinais que podem não terem sido ensinadas, já que por exemplo, podemos ensinar um sinal que no RJ signifique “Oi”, porém no Acre “Oi” pode ser representado por outro sinal, que pode não ter sido ensinado para a IA;

Aceitação e adoção: Esse problema está relacionado ao quão bem a TALIA vai conseguir atender as necessidades de seus usuários e se haverá muitas críticas referentes a problemas encontrados durante o uso.

4.2. PRIORIZAÇÃO DOS RISCOS

Riscos com impacto maior:

Sinais ensinados errados: Este é o quesito mais importante, pois se as amostras não forem gravadas de forma correta e em um ambiente que tenha boa visibilidade, podem ocorrer imprecisões de tradução e dificuldade de reconhecer até mesmo sinais básicos, atrapalhando o principal critério de qualidade do nosso produto;

Riscos com impacto menor:

Aceitação e adoção: Embora a aceitação e adoção pelos usuários finais sejam críticas para o sucesso do projeto, isso pode não ter um impacto imediato na segurança ou precisão dos resultados;

Regionalismo: Apesar de também ser um problema ele só irá atrapalhar no reconhecimento de variações de sinais que não forem comuns na nossa região;

4.3. PLANOS DE MITIGAÇÃO

Sinais ensinados errados: Antes de gravar as amostras de cada sinal faremos pesquisas em diversas fontes para garantir que os sinais serão gravados de forma correta, também tomaremos o cuidado de realizar as gravações em um local bem iluminado com uma roupa que se destaque bem no ambiente;

Aceitação e adoção: Para garantir a aceitação e adoção é importante envolver os usuários finais no projeto desde o início e garantir que o sistema atenda às suas necessidades e expectativas. Também é importante fornecer treinamento adequado e suporte para os usuários finais, além de comunicar claramente os benefícios do sistema.

Regionalismo: Treinando a IA com diferentes sinais que tenham o mesmo significado.

4.4. PLANOS DE CONTINGÊNCIA

Os planos de contingência da inteligência artificial em Libras são importantes para garantir que a nossa equipe possa responder rapidamente a possíveis problemas e minimizar o impacto de possíveis riscos.

Sinais ensinados errados: Se for percebido que um determinado sinal está sendo interpretado sempre errado devido a uma amostragem ruim durante a fase de treinamento, esse sinal deverá ser removido temporariamente e as amostras deverão ser gravadas novamente;

Aceitação e adoção: Iremos fazer uma lista dos feedbacks negativos, vamos organizar os problemas relatados por nível de prioridade e feito isso, iremos começar a solucionar cada problema imediatamente, por ordem de prioridade.

Regionalismo: Treinando com diferentes variações do mesmo sinal.

É importante lembrar que os planos de contingência devem ser revisados regularmente e atualizados conforme necessário para garantir que a equipe esteja preparada para lidar com possíveis problemas.

5. GERENCIAMENTO DE CONFIGURAÇÃO

5.1. PLANO DE GERENCIAMENTO DE CONFIGURAÇÃO

Para gerenciar melhor o desenvolvimento da TALIA utilizamos GitHub, Notion e Drive, conforme explicaremos a seguir.

5.2. REPOSITÓRIO

5.2.1. Notion

O Notion é uma ferramenta de produtividade e gestão de projetos que utilizamos para organizar as informações do nosso TCC como todo. Através do Notion, criamos documentos estruturados para registrar as etapas do projeto, anotações, referências bibliográficas, cronogramas e tarefas. Essa plataforma colaborativa permitiu uma gestão eficiente do trabalho e a fácil visualização do progresso.

Figura 25: Notion

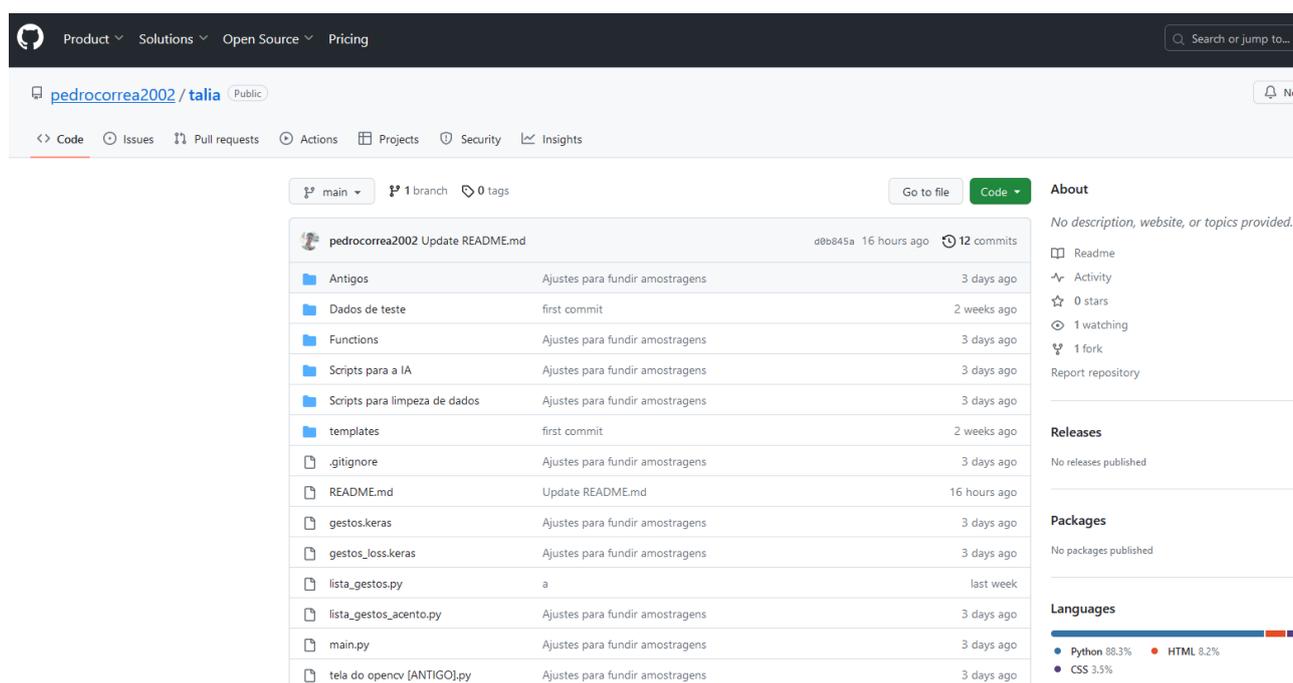


Fonte: Elaboração própria. TALIA. Notion. Disponível em: <<https://merciful-panda-ef3.notion.site/T-A-L-I-A-IA-de-reconhecimento-de-sinais-e-tradu-o-para-libras-a3c6072be0a0449a805fd260534b8f35?pvs=4>>

5.2.2. GitHub

O GitHub é uma plataforma de hospedagem de código-fonte e colaboração. Utilizamos o GitHub para armazenar o código-fonte do nosso projeto de TCC. Através do GitHub, pudemos controlar as alterações feitas no código, facilitando o trabalho em equipe e o acompanhamento do progresso do projeto.

Figura 26: Repositório do Github

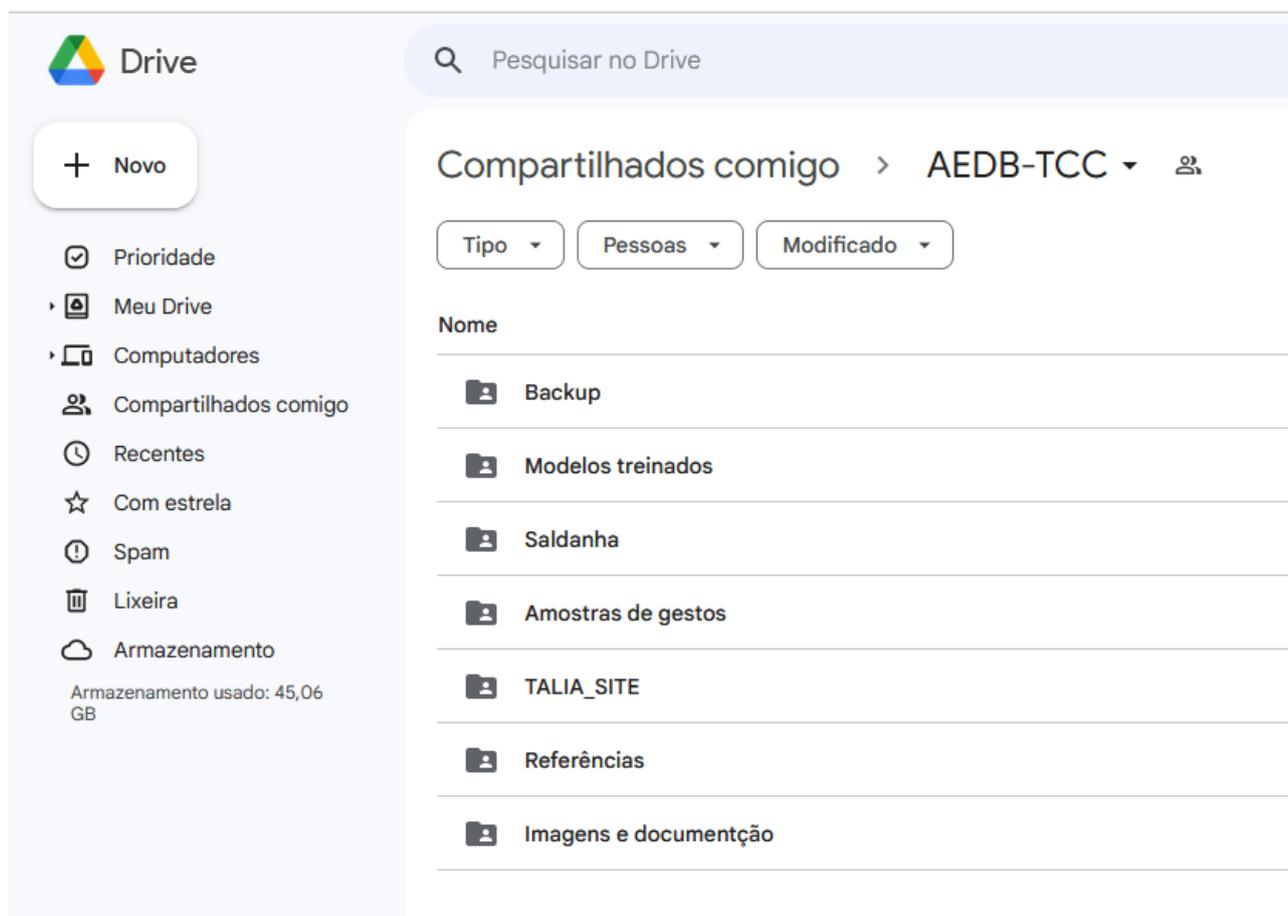


Fonte: Elaboração própria. **Repositório do GitHub**. GitHub. Disponível em: <<https://github.com/pedrocorrea2002/talia>>

5.2.3. Google Drive:

O Google Drive é um serviço de armazenamento em nuvem que utilizamos para armazenar e compartilhar diversos tipos de arquivos relacionados ao TCC. Ele nos permitiu fazer backup dos documentos, como textos, planilhas, apresentações, imagens e outros recursos relevantes para o projeto. Além disso, utilizamos o Google Drive para compartilhar esses arquivos com o orientador, colaboradores e outros envolvidos no processo de desenvolvimento do TCC. Isso facilitou a colaboração e o acesso remoto aos materiais necessários para o projeto.

Figura 27: Pasta do Drive



Fonte: Elaboração própria. Pasta do Drive. Drive.

6. ESTRATÉGIA DE TESTES

6.1. PLANO DE TESTES

Antes de iniciar o teste, treinamos 6 redes neurais distintas, para assim poder comparar o resultado das redes neurais treinadas. Todas as redes neurais foram treinadas utilizando as mesmas configurações de rede e os mesmos dados, tendo como diferença apenas os pesos e *bias* que são definidos aleatoriamente pelo Keras durante o treinamento.

Outras diferenças foram os valores finais do *Loss* e do *Categorical accuracy*, sendo ambas métricas que vão sendo calculadas durante o treinamento, sendo que a rede neural enquanto treina sempre tem o objetivo de diminuir o *Loss* e aumentar o *Categorical accuracy*.

Antes do treinamento, separamos 20% das amostras para serem usadas para teste e 80% das amostras para treinar a IA sendo que esses 20% não seriam ensinadas para a IA, seriam apenas usadas depois do treinamento para verificar se a IA entendeu a lógica ou apenas decorou as respostas, fenômeno esse chamado de overfitting.

6.2. TESTES DE UNIDADE AUTOMATIZADOS

Para fazer um teste unitário da funcionalidade de tradução utilizamos a matriz de confusão, que é uma técnica que pega amostras que não foram ensinadas para a IA e testa em quantos casos ela retorna o valor correto, classificando os resultados nas 4 categorias abaixo:

VP = Verdadeiro Positivo

VN = Verdadeiro Negativo

FP = Falso Positivo

FN = Falso Negativo

Além de usar a matriz de confusão, também medimos a quantidade bruta de erros e acertos das 6 redes neurais.

Abaixo é possível ver o valor final de *Loss* e *Categorical accuracy* de cada rede neural após o treinamento, a quantidade de acertos, erros, total de amostras usadas no teste além dos resultados da matriz de confusão para cada uma das seguintes palavras: “Olá”, “Boa noite”, “Criar”, “Aprender”, “Meu nome é”, “TALIA”.

Tabela 2: Matriz de confusão: rede neural

Rede neural 01				
Sinal	VP	VN	FP	FN
Olá	29	138	1	0
Boa noite	27	141	0	0
Criar	28	139	1	0
Aprender	26	141	0	1
Meu nome	31	136	1	0
TALIA	24	142	0	2
<i>Loss:</i>		0,3596		
<i>Categorical accuracy:</i>		0,0869		
Acertos:		165		
Erros:		3		
Total de amostras usadas no teste:		168		

Fonte: Elaboração própria

Rede neural 02				
Sinal	VP	VN	FP	FN
Olá	23	139	0	1
Boa noite	26	142	0	01
Criar	28	134	1	0
Aprender	23	140	0	1
Meu nome	30	133	1	0
TALIA	23	141	0	2
<i>Loss:</i>		0,3587		
<i>Categorical accuracy:</i>		0,0875		
Acertos:		164		
Erros:		4		
Total de amostras usadas no teste:		168		

Fonte: Elaboração própria

Rede neural 03				
Sinal	VP	VN	FP	FN
Olá	29	134	0	1
Boa noite	30	139	0	0
Criar	27	138	0	0
Aprender	29	145	0	1
Meu nome	33	135	1	0
TALIA	29	143	0	2

<i>Loss:</i>	0,3574
<i>Categorical accuracy:</i>	0,0888
Acertos:	163
Erros:	5
Total de amostras usadas no teste:	168

Fonte: Elaboração própria

Rede neural 04				
Sinal	VP	VN	FP	FN
Olá	34	142	0	0
Boa noite	29	138	1	1
Criar	32	137	1	0
Aprender	33	141	1	1
Meu nome	29	138	0	0
TALIA	29	140	0	2
<i>Loss:</i>				
0,3574				
<i>Categorical accuracy:</i>				
0,0869				
Acertos:				
164				
Erros:				
5				
Total de amostras usadas no teste:				
168				

Fonte: Elaboração própria

Rede neural 05				
Sinal	VP	VN	FP	FN
Olá	32	129	1	0
Boa noite	25	144	0	1
Criar	32	137	1	0
Aprender	29	143	0	1
Meu nome	30	137	0	1
TALIA	22	139	1	2
<i>Loss:</i>				
0,3587				
<i>Categorical accuracy:</i>				
0,0863				
Acertos:				
162				
Erros:				
6				
Total de amostras usadas no teste:				
168				

Fonte: Elaboração própria

Rede neural 06				
Sinal	VP	VN	FP	FN
Olá	32	141	0	1
Boa noite	29	139	0	1
Criar	31	137	1	0
Aprender	32	142	1	1
Meu nome	30	137	0	0
TALIA	26	140	0	2
Loss:				
		0,3574		
Categorical accuracy:				
		0,0887		
Acertos:				
		164		
Erros:				
		4		
Total de amostras usadas no teste:				
		168		

Fonte: Elaboração própria

6.3. TESTES DE VALIDAÇÃO

Na fase de testes de validação, envolvemos pessoas que não estavam presentes no desenvolvimento do projeto, porém que entendessem Libras, como intérpretes. Dessa forma, conseguimos validar se a forma como ensinamos a TALIA está de acordo com a forma que realmente são feitos os sinais.

6.4. ACEITES DOS MÓDULOS PELOS STAKEHOLDER

O sistema desenvolvido passará por uma validação final dos desenvolvedores, orientador, intérpretes de Libras e outros potenciais usuários finais. Nos dias que antecederão o último seminário daremos acesso ao sistema para que a banca examinadora faça os testes e avalie o funcionamento da TALIA.

Outros aceites já vêm sendo feitos pelas bancas presentes nos três seminários e na apresentação final, na forma da apresentação da evolução do projeto para os mesmos.

7. ESTRATÉGIA DE IMPLANTAÇÃO E SUPORTE

7.1. NECESSIDADES DE IMPLANTAÇÃO

Para a TALIA poder funcionar as únicas coisas que são necessárias é um dispositivo com câmera, que pode ser um celular, tablet, notebook ou desktop, e uma conexão com a internet de no mínimo 5mbps.

Pois para utilizar a TALIA basta acessar o *link* da instância da Amazon EC2 que hospedamos a IA. Uma vez no site, basta clicar no botão com a logo da TALIA que fica abaixo do frame onde aparece a imagem da câmera.

7.1.1. ARQUITETURA DE IMPLANTAÇÃO

Para que a TALIA funcione, hospedamos ela em uma instância criada no Amazon EC2, nela criamos um ambiente virtual contendo todas as bibliotecas de python necessárias para rodar a TALIA, são essas:

Pytttsx3: Converte texto em fala;

MediaPipe: Analisa imagens e se houver pessoas na imagem ela capta os pontos do corpo da pessoa, existe a opção de captar também os pontos do rosto, porém preferimos captar apenas os pontos das duas mãos e do corpo para que a TALIA precisasse processar menos dados e assim garantíssemos o fator tempo real;

Flask: Foi usado para o desenvolvimento da parte web do projeto;

Scikit-learn: Foi usado para a divisão entre dados de testes e dados de treinamento antes da execução do treinamento propriamente dito e também para medir a eficiência da TALIA de forma automática por meio da matriz de confusão, sem que fosse necessário que alguém realmente fizesse vários testes na mesma;

Keras e TensorFlow: Foram usados para fazer as configurações da rede neural, executar o treinamento e gerar o arquivo .keras ao fim do treinamento.

Também precisamos instalar o próprio python e usando ele e o gerenciador de pacotes *pip*, instalamos as bibliotecas acima e executamos a TALIA, para que ela possa ser acessada pelo *link* da instância do EC2.

7.1.2. CONFIGURAÇÃO DOS SERVIDORES

A TALIA será rodada em uma instância do Amazon EC2, cada instância equivale a um computador individual com uma quantidade definida de CPU, de RAM, de GPU, de armazenamento e uma placa de rede própria. A instância do EC2 é elástica, ou seja, ela consegue automaticamente aumentar ou diminuir as configurações de RAM, armazenamento e CPU.

A instância que criamos para hospedar a TALIA possui as seguintes configurações:

Tabela 3: Configurações necessárias para rodar a TALIA na máquina

Tipo de instância	t2.micro
Sistema operacional	Ubuntu Server 22.04 LTS
Largura da banda	100 gigabytes
Tamanho do armazenamento	8 GiB
Tipo do armazenamento	gp2 (SSD)

Fonte: Elaboração própria

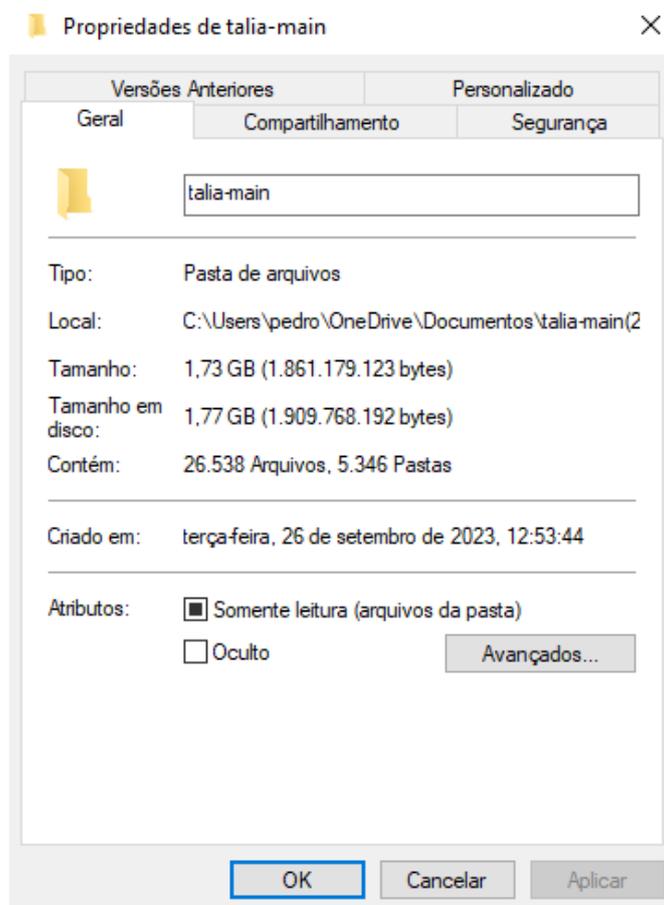
7.1.3. CONFIGURAÇÃO DOS CLIENTES

Os usuários necessitarão de um dispositivo com câmera e de uma conexão de internet com no mínimo 5mbps.

7.1.4. INFRAESTRUTURA NECESSÁRIA

Espaço em disco: O hardware que hospedar a TALIA precisa ter um espaço mínimo de 2,20gb reservado no seu disco, pois a cada passo que alteramos a sua versão vai consumindo mais espaço no disco.

Figura 28: Espaço ocupado sem o arquivo .keras e sem as amostras de sinais



Fonte: Elaboração própria

Memória RAM: Após alguns testes dentro da instância do EC2 percebemos que rodar a TALIA não consome mais que do que 4,5gb ram;

CPU: utilizando como exemplo um processador de 3.6Ghz, 4 núcleos e 4 threads, com os 4 núcleos sendo usados ao mesmo tempo e mantendo alguma porcentagem na faixa de 20% a 50% para rodar apenas a TALIA mais o sistema operacional (Lubuntu);

7.2. CRONOGRAMA DE TREINAMENTOS

Por ser tratar de um sistema com apenas um único caso de uso, que é a funcionalidade de tradução, achamos desnecessário fazer o treinamento com os usuários, porém definimos todo o passo a passo e requisitos para a utilização da TALIA no manual do usuário.

8. CONCLUSÃO

Neste trabalho, apresentamos o projeto TALIA, uma inteligência artificial capaz de traduzir sinais da Língua Brasileira de Sinais (Libras) em tempo real. O projeto surgiu como um desafio de aprendizagem e inovação, pois enfrentamos diversas dificuldades ao longo do processo, tais como:

- A falta de conhecimento nas técnicas e teorias relacionadas ao *Machine Learning*, configurações de redes neurais e a falta de conhecimento sobre os sinais da Libras;
- A falta de conhecimento do grupo como um todo sobre IA e Libras;
- O regionalismo da Libras, que dificultou a padronização dos sinais e a pesquisa de referências.

Apesar desses obstáculos, conseguimos obter diversos resultados positivos ao decorrer do ano com o projeto TALIA, onde através de sinais a IA foi capaz de realizar diversas traduções dos sinais em tempo real. Além disso, ampliamos nosso conhecimento sobre novas tecnologias e aprendemos a construir uma rede neural do zero. Para isso, buscamos sempre referências em PDF's, sites, canais no Youtube e blogs sobre inteligência artificial, contando também com o apoio e a confiança do nosso orientador Bruno Saldanha. Membros do grupo também buscaram conhecimentos em cursos sobre *Machine* e *Deep Learning* na *Data Science Academy*, *Coursera* e projetos parecidos com o nosso no Youtube e GitHub para usar de inspiração.

Para vencer as dificuldades que encontramos, adotamos as seguintes estratégias:

- Utilizamos as recém-lançadas ferramentas de pesquisa a base IA, juntamente com pesquisas no Google e a orientação do professor Bruno Saldanha, que tem conhecimento no assunto, para superar a falta de conhecimento em IA;
- Pesquisamos em canais de Libras do Youtube e em PDFs sobre o assunto, encontrados na Internet, para superar a falta de conhecimento em Libras. Alguns intérpretes de Libras que conhecemos também se propuseram a dar orientações quando tínhamos dúvidas entre variações de sinais, dizendo qual a versão que é usada na nossa região;
- Otimizamos o processo de treinamento da TALIA, utilizando técnicas para tornar nosso processo de coleta de amostras mais eficiente, como por exemplo, registrar no Notion uma listagem de quantidades de amostras de cada sinal e quem gravou cada

quantidade, deixar o Windows de lado e optar pelo Lubuntu para conseguir mais eficiência de processamento e fazer gravações mais rápidas, diminuimos também a carga de amostras por pessoa, que antes era de 140 amostras por sinal para 20, implantamos um mecanismo que espelha o frame quando capta ele, para assim obter o dobro de amostras com a mesma quantidade de gravações, fizemos gravações mais diferenciadas, evitando ficar na mesma posição em todas as amostras, para assim deixar a amostragem mais rica e evitar o fenômeno do *overfitting*. Devido à falta de prática com IA, também juntamos todo o conhecimento que conseguíamos no Notion, para ter uma base centralizada de consulta de referências e de anotações.

Com base nesses resultados e dificuldades, concluímos que o projeto TALIA foi uma experiência enriquecedora e desafiadora, que nos permitiu desenvolver habilidades técnicas e criativas na área de inteligência artificial. Também contribuimos para a inclusão social das pessoas surdas, oferecendo uma ferramenta que facilita a comunicação entre elas e as pessoas ouvintes. Por fim, sugerimos que futuros grupos que tenham interesse em dar continuidade ao projeto busquem alguém com conhecimento profundo em Libras e procurem se aprofundar mais no conhecimento de inteligência artificial para evitar os problemas que tivemos devido à falta de conhecimento. Esperamos que nosso trabalho possa inspirar outras iniciativas semelhantes e promover o avanço da ciência e da tecnologia.

REFERÊNCIAS

PDFs e sites:

https://ufsj.edu.br/portal2-repositorio/File/incluir/libras/curso_de_libras_-_graciele.pdf

https://cesad.ufs.br/ORBI/public/uploadCatalago/14463525082022LIBRA_-_Aula_10.pdf

<https://www.handtalk.me/br/aplicativo/>

<https://www.redhat.com/pt-br/topics/automation/what-is-configuration-management>

<https://aws.amazon.com/pt/what-is/overfitting/>

<https://didatica.tech/underfitting-e-overfitting/>

<https://blog.signumweb.com.br/curiosidades/linguas-de-sinais-pelo-mundo/>

https://en.wikipedia.org/wiki/List_of_sign_languages

Canais de Libras no Youtube

<https://www.youtube.com/@incluirtecnologia>

<https://www.youtube.com/@SinaisdiariosdeLibras>

<https://www.youtube.com/@netolibras/about>

Ferramentas usadas

<https://pypi.org/project/pyttsx3/>

<https://www.tensorflow.org/install?hl=pt-br>

<https://keras.io/>

<https://flask.palletsprojects.com/en/2.3.x/>

<https://developers.google.com/mediapipe>

<https://opencv.org/>

<https://scikit-learn.org/stable/>

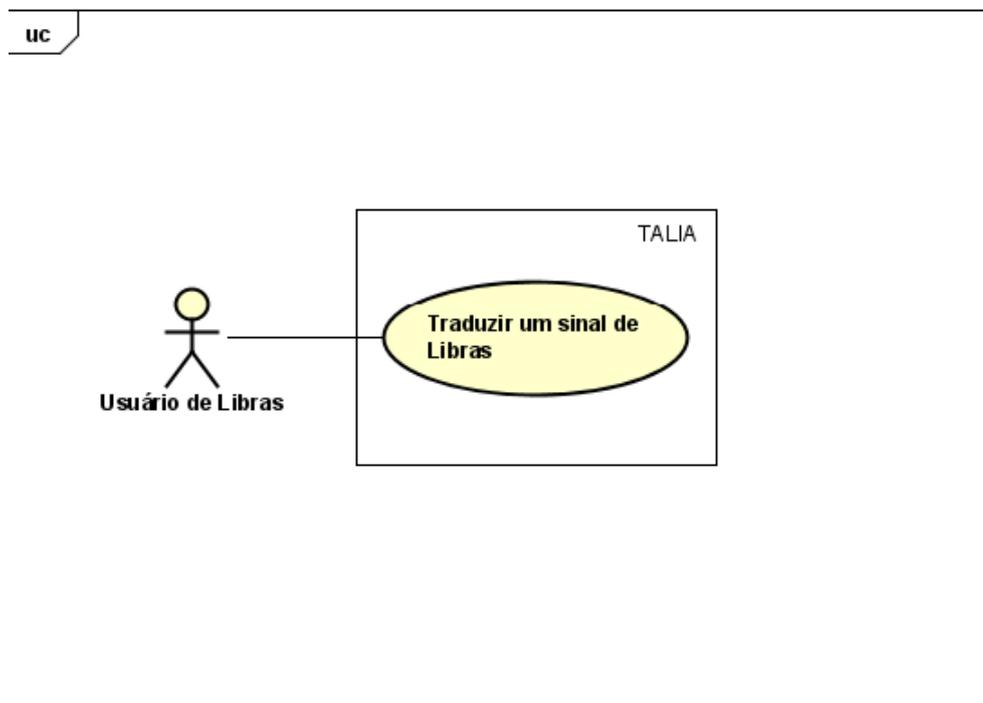
<https://numpy.org/>

<https://aws.amazon.com/pt/ec2/>

<https://lubuntu.me/>

APÊNDICE A: DIAGRAMA DE CASOS DE USO

Figura 29: Diagrama de caso de uso



Fonte: Elaboração própria

APÊNDICE B: DESCRIÇÕES DE CASOS DE USO

CDU001 – Traduzir um sinal de Libras

Sumário: O ator usuário de Libras se posiciona em frente a câmera e começa a fazer sinais de Libras para serem traduzidos.

Ator primário: Usuário de Libras

Pré-condições:

- O dispositivo deve possuir uma conexão estável com a internet;
- O dispositivo que está acessando o sistema deve possuir uma câmera;
- O ator Usuário de Libras deve estar posicionado dentro da área de visão da câmera, de modo que seu corpo da cintura para cima esteja visível, assim como seus braços. Seu rosto não é totalmente necessário, já que os pontos do rosto não foram usados nas amostras de treinamento.

Fluxo principal

1. O sistema começa a monitorar a câmera;

2. O ator Usuário de Libras começa a fazer sinais de Libras;
3. O sistema, em tempo real, coleta as coordenadas dos pontos do corpo presentes em cada frame, até atingir 30 frames;
4. Ao atingir 30 frames, o sistema envia uma lista com as coordenadas dos pontos coletados nesses 30 frames para o modelo de IA tentar traduzir;
5. Caso o sistema tenha pelo menos 90% de certeza de que uma expressão específica é a tradução ideal e caso ela não for a mesma coisa que a última tradução feita, é pego uma lista com as 5 traduções mais prováveis;
6. O sistema exibe um áudio da expressão que é aquela com maior probabilidade de ser a correta;
7. A expressão correta juntamente com as outras 4 são exibidas na esquerda do frame da câmera na ordem do mais provável para o menos provável, juntamente com uma barra representando a probabilidade de cada expressão;
8. A expressão com maior probabilidade é adicionada em uma lista que contém todas as traduções anteriores;
9. Essa lista alimenta uma barra que fica na direita do frame da câmera que mostra um histórico de traduções;
10. O sistema limpa a lista de frames e volta para o item 2.

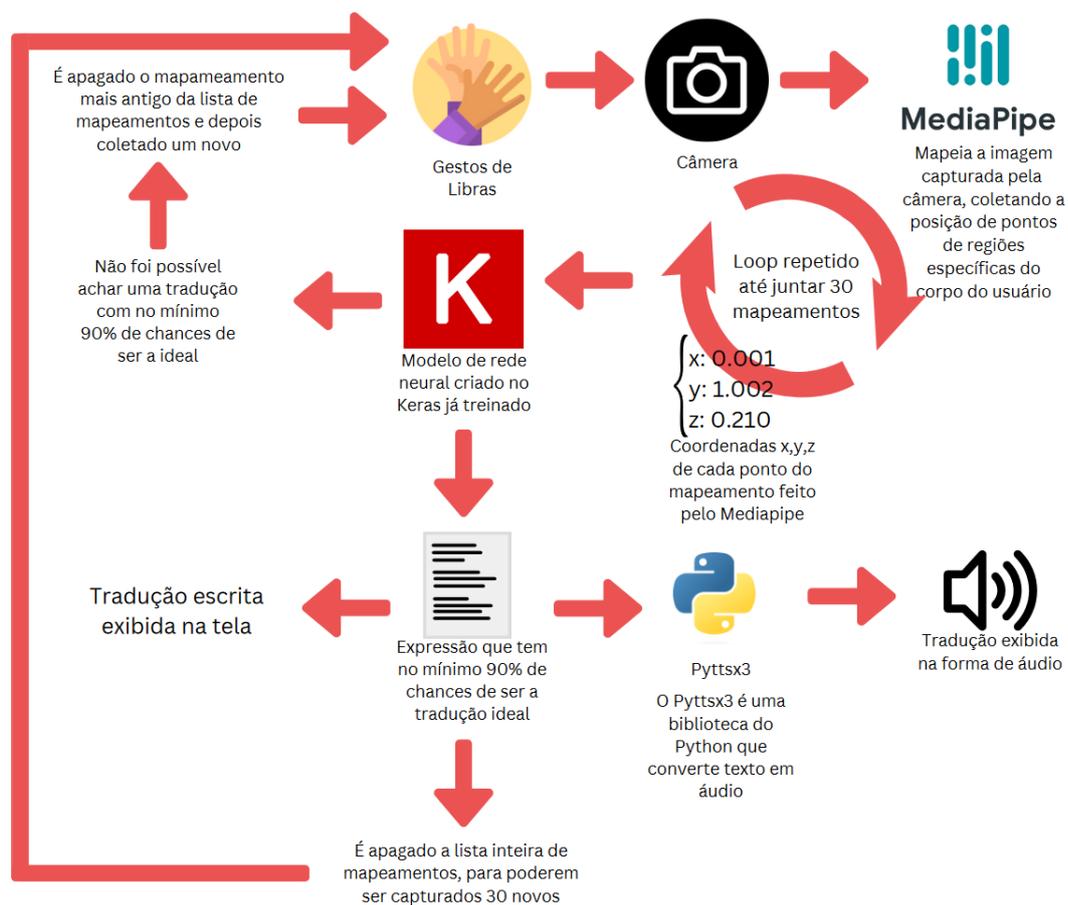
Fluxo alternativo (5): O sistema não encontrou expressão nenhuma que tenha pelo menos 90% de chances de ser a tradução ideal ou a expressão encontrada já é a mesma que a última tradução feita

- A. O sistema vai coletar as coordenadas de mais um frame, vai adicionar essas coordenadas na última posição da lista de frames e vai esquecer o frame mais antigo da lista;
- B. Essa lista de frames é novamente enviada para o modelo de IA tentar traduzir;
- C. Os itens A e B são repetidos até que o sistema tenha pelo menos 90% de certeza de que uma expressão específica é a tradução ideal e essa tradução não seja a mesma coisa que a última tradução feita;
- D. O fluxo prossegue para o passo (5) do fluxo principal.

APÊNDICE D: DIAGRAMA DE ATIVIDADES

CDU 01: Traduzir Libras

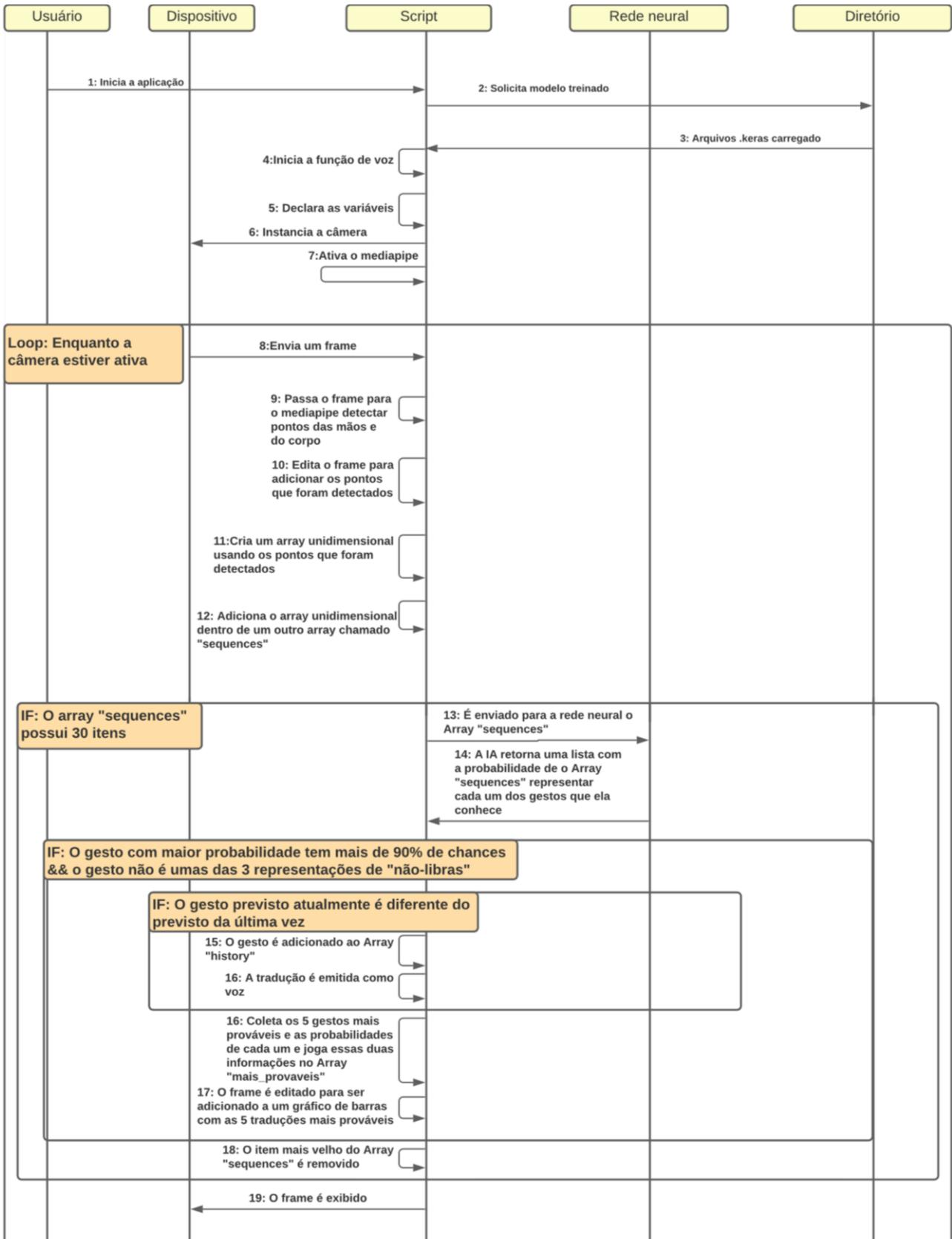
Figura 30: Diagrama de atividades



Fonte: Elaboração própria

APÊNDICE E: DIAGRAMA DE SEQUÊNCIA
 CDU 01: Traduzir um sinal de Libras

Figura 31: Diagrama de seqüências



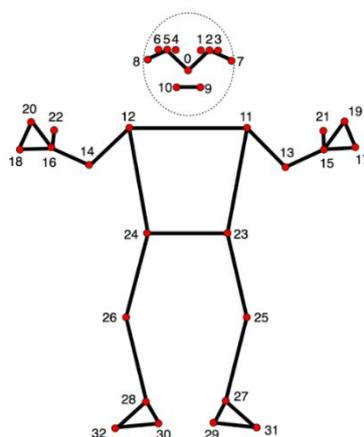
APÊNDICE G: DICIONÁRIO DE DADOS

ARQUIVOS .npy

Os dados que a nossa IA vai usar para aprender um determinado sinal são 30 arquivos do tipo .npy por amostra de cada sinal, ou seja, cada sinal pode possuir centenas de pastas contendo 30 arquivos .npy cada uma. Esses 30 arquivos juntos compõem as coordenadas dos pontos do corpo e mãos coletados ao longo de 2 segundos, sendo cada arquivo referente às coordenadas de 1 frame desses 2 segundos.

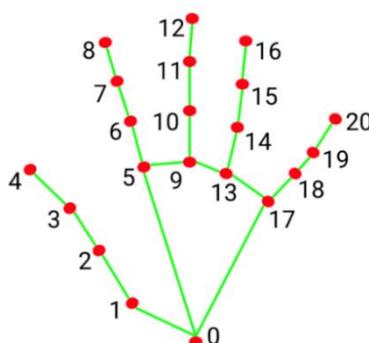
Cada arquivo contém um Array criado pela biblioteca Numpy durante a etapa de gravação de sinais. Esses Arrays possuem 258 linhas, sendo cada 3 linhas (mãos) ou 4 linhas (corpo) referentes a um ponto. Esses pontos são coletados das seguintes partes do corpo e mãos:

Figura 32: Pontos do corpo



Fonte: Pose landmarker model. MediaPipe. Disponível em: https://developers.google.com/mediapipe/solutions/vision/pose_landmarker. Acesso em: 27 set. 2023

Figura 33: Pontos da mão



Fonte: Hand landmark model bundle. MediaPipe. Disponível em: https://developers.google.com/mediapipe/solutions/vision/gesture_recognizer#get_started. Acesso em: 27 set. 2023

Sabendo disso o Array salvo em cada arquivo .h5, é uma pilha de 258 linhas que segue a seguinte estrutura:

Tabela 4: Estrutura da pilha de dados I

Descrição	Estrutura
21 pontos da mão esquerda	Coordenada X do ponto Coordenada Y do ponto Coordenada Z do ponto
21 pontos da mão direita	Coordenada X do ponto Coordenada Y do ponto Coordenada Z do ponto
33 pontos do corpo	Coordenada X do ponto Coordenada Y do ponto Coordenada Z do ponto Grau de visibilidade do ponto de 0 a 1

Fonte: Elaboração própria

Tabela 5: Exemplo de como se parece essa pilha de dados

0.002121	X do ponto 1 da mão esquerda
0.122142	Y do ponto 1 da mão esquerda
0.421212	Z do ponto 1 da mão esquerda
0.624341	X do ponto 2 da mão esquerda
0.242412	Z do ponto 21 da mão esquerda
0.241144	X do ponto 1 da mão direita
0.762311	Z do ponto 21 da mão direita
0.002140	X do ponto 1 do corpo
0.241242	Y do ponto 1 do corpo
0.713201	Z do ponto 1 do corpo
0.542211	Visibilidade do ponto 1 do corpo
0.784211	Visibilidade do ponto 33 do corpo

Elaboração própria

ARQUIVO .h5

O arquivo .h5 é gerado depois que a IA é treinada usando as amostras de sinais mencionadas anteriormente. Esse arquivo .h5 guarda toda a estrutura da rede neural treinada além de cada peso e cada *bias*.

Figura 34: Estrutura do arquivo .h5 I



Fonte: Elaboração própria

Dentro de um arquivo .h5 temos a seguinte estrutura:

Temos a pasta *model_weights*, que guarda uma pasta para cada camada da rede neural, onde cada uma dessas pastas contém os valores de peso e *bias* de todas as sinapses e neurônios daquela camada:

Figura 35: Estrutura do arquivo .h5 II



Fonte: Elaboração própria

Figura 36: Como ficam salvos os valores de peso e bias

	0	1	2	3	4	5	6
0	0.0754314	0.06340...	-0.01800...	0.04921...	0.07645...	-0.04947...	7.79339... -0.002
1	-0.05228...	0.14988...	-0.05902...	-0.05673...	0.15230...	-0.12811...	-0.11589... -0.052
2	0.08524...	0.01573...	0.06390...	0.04893...	-0.07944...	-0.06813...	0.00741... -0.068
3	0.08347...	-0.06287...	0.00937...	-0.02169...	0.11303...	-0.12326...	-0.17896... -0.086
4	0.13025...	0.07416...	0.10171...	-0.08122...	0.12674...	-0.07518...	-0.15939... 0.101
5	0.01050...	-0.05806...	-0.09004...	0.12333...	0.03402...	0.10928...	0.08021... 0.120
6	0.14136...	-0.01088...	0.14710...	-0.01852...	-0.15401...	0.00574...	0.03057... -0.152
7	0.11708...	-0.09817...	0.14150...	-0.00387...	-0.03773...	0.1503159	0.11220... -0.122
8	0.10938...	-0.08026...	-0.08424...	-0.15270...	-0.08857...	-0.00267...	0.08993... 0.079
9	0.18441...	0.09466...	0.09197...	-0.01071...	-0.12500...	-0.11272...	0.08775... 0.066
10	-0.02539...	0.1097106	0.00817...	0.02813...	-0.07731...	-0.08629...	-0.02878... 0.001

0	0.04601...
1	-0.06612...
2	-0.01668...
3	0.07545...
4	-0.04364...
5	0.08672...
6	-0.01679...
7	-0.02345...
8	0.1825696
9	0.15952...
10	0.14500...
11	-0.06654...

Fonte: Elaboração própria

Na esquerda temos uma matriz de pesos, que possui 1 linha para cada neurônio da camada anterior e 1 (camadas densas) ou 4 (camadas LSTM) colunas para cada neurônio da camada atual. Está organizado como matriz justamente porque todo neurônio da camada anterior está conectado com todo neurônio da camada atual, sendo 1 peso para cada conexão;

Na direita temos a lista de *bias*, sendo 1 *bias* por neurônio daquela camada;

Abaixo da pasta *model_weights* temos a pasta *optimizer_weights*, os valores referentes ao otimizador escolhido, que no nosso caso foi o Adam, que possui duas versões paralelas da rede neural (v) e (m) que são usadas para aumentar a precisão da rede principal durante o treinamento.

MANUAL DO USUÁRIO

Pré requisitos

Possuir dispositivo com câmera para a TALIA ter acesso a sua imagem.

Entrar no link

Ter acesso a internet para poder ter acesso a TALIA.

Login da TALIA

Para acessar a TALIA, você precisará inserir seu endereço de e-mail e senha. Na página de login, selecione se é seu primeiro acesso ou se você já acessou antes. Tela de login da TALIA com as opções de primeiro acesso e já acessou antes

Se for seu primeiro acesso, insira seu nome, endereço de e-mail e senha, confirme sua senha na caixa reescreva essa senha, logo em seguida clique no botão cadastrar.

Para acessar a TALIA se você já possui uma conta:

Na página de login, insira seu endereço de e-mail e senha e clique no botão entrar, ao finalizar o processo será direcionado a tela de tradução.

Figura 36: Login TALIA.



A imagem mostra a interface de login da TALIA, dividida em duas seções principais: 'Primeiro acesso' (fundo laranja) e 'Já acessei antes' (fundo branco). Ambas as seções possuem campos de entrada para 'Qual seu nome?' e 'Qual sua senha?'. A seção 'Primeiro acesso' também possui um campo para 'Reescreva essa senha:' e um botão 'Cadastrar'. A seção 'Já acessei antes' possui um botão 'Entrar'. No topo da interface, há um ícone de uma mão e o nome 'TALIA'.

Fonte: Elaborado pelo grupo.

Tela do tradutor

Com os primeiros passos feitos você será direcionado para a tela de tradução.

Se posicionar no meio da tela para que a captura seja feita de forma certa, a imagem aparecerá no meio da tela no momento que a câmera for reconhecida.

Apertar no botão de gravar para dar início a tradução.

Minhas amostras

Apertar em minhas amostras no canto superior direito dentro da tela do tradutor você terá acesso as suas amostras gravadas, no canto inferior direito da tela tem o histórico das amostras que a TALIA acabou de traduzir.

Abaixo possui um espaço para o texto traduzido para que o usuário possa ler.

Figura 37: Tradução TALIA



Fonte: Elaborado pelo grupo.