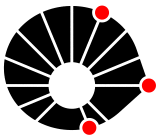

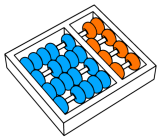

# Legendagem na Língua de Sinais Brasileira de Páginas da Internet

*G. S. Pereira* *H. Pedrini* *J. M. Martino*

Relatório Técnico - IC-PFG-25-43
Projeto Final de Graduação
2025 - Dezembro

UNIVERSIDADE ESTADUAL DE CAMPINAS
INSTITUTO DE COMPUTAÇÃO

# Legendagem na Língua de Sinais Brasileira de Páginas da Internet

Gabriel de Sousa Pereira  Hélio Pedrini  José Mario De Martino

**Resumo**

O presente trabalho desenvolveu uma infraestrutura tecnológica para viabilizar a tradução automática de conteúdos Web para a Língua Brasileira de Sinais (Libras), fundamentada na construção de um corpus multimodal de alta fidelidade com cerca de 2400 sentenças capturadas via sistemas de captura de movimento. A proposta concretizou-se na implementação de interfaces de programação de aplicações (APIs) de integração em Python, compostas por um *crawler* para mapeamento de domínios e um *parser* capaz de segmentar textos e injetar automaticamente janelas de acessibilidade, cuja solução foi validada na adaptação de páginas dinâmicas complexas como cardápios universitários.

Na vertente de interface com o usuário, o projeto entregou um *widget* modular que suporta tanto a reprodução de vídeos pré-renderizados quanto a animação de avatares tridimensionais (3D) em tempo real via WebGL, visando otimizar o consumo de dados. A arquitetura desenvolvida contempla ainda um ambiente de renderização em tempo real validado publicamente, estabelecendo um ambiente adequado de ferramentas de engenharia de dados e visualização gráfica para promover a autonomia digital da comunidade surda.

## 1 Introdução

A perda auditiva ou a surdez traz restrições à percepção auditiva e, portanto, a impossibilidade de aprender uma língua oral de forma natural e, consequentemente, dificuldades de alfabetização na modalidade escrita da língua. Para a Comunidade Surda, a língua de sinais é mais confortável e apropriada para o acesso à informação. As línguas de sinais são línguas naturais, gramaticalmente estruturadas por um conjunto de regras linguísticas e adquiridas de forma natural por pessoas surdas no convívio com seus pares.

No Brasil, a Língua Brasileira de Sinais (Libras) é reconhecida como meio cooficial de comunicação e expressão da Comunidade Surda brasileira por meio da Lei nº 10.436/2002, regulamentada pelo Decreto nº 5.626/2005. Adicionalmente, a Lei nº 13.146, de 6 de julho de 2015, também conhecida como Estatuto da Pessoa com Deficiência, estabelece ações do poder público destinadas a assegurar e a promover, em condições de igualdade, o exercício dos direitos e das liberdades fundamentais por pessoa com deficiência, visando à sua inclusão social e cidadania. Explicitamente em seu artigo 63, essa lei estabelece que é obrigatória a acessibilidade nos sítios da internet mantidos por empresas com sede ou representação comercial no país ou por órgãos de governo, para uso da pessoa com

deficiência, garantindo-lhe acesso às informações disponíveis, conforme as melhores práticas e diretrizes de acessibilidade adotadas internacionalmente.

Nesse contexto, o presente projeto visa contribuir com o desenvolvimento de mecanismo de legendagem de sites de Internet apoiado em abordagem de tradução automática com a apresentação do resultado da tradução por meio de avatar sinalizante. O projeto se insere nos esforços de desenvolvimento realizados no Centro de Ciência para o Desenvolvimento (CCD) – Tecnologia Assistiva e Acessibilidade em Libras (TAAL), financiado pela Fundação de Amparo à Pesquisa do Estado de São Paulo (FAPESP).

Este trabalho visa construir um corpus Português-Libras, composto de vídeos, dados de captura de movimento e anotado. O corpus será utilizado no desenvolvimento de algoritmos baseados em dados (*data-driven algorithms*) de tradução automática. Além disso, o projeto visa implementar mecanismos para a integração de avatar sinalizante de Libras em páginas da Internet.

# 2 Estado da Arte

## 2.1 Contexto Global

A Tradução Automática de Língua de Sinais (SLMT) consolidou-se como um campo de investigação fundamental para a promoção da autonomia e inclusão da comunidade surda, visando mitigar as barreiras comunicativas existentes entre indivíduos sinalizantes e não sinalizantes [1, 29]. Embora a tradução automática entre línguas escritas tenha atingido níveis de desempenho comparáveis ao humano por volta de 2017, impulsionada pela introdução da arquitetura *Transformer* [26], a SLMT permanece diante de obstáculos singulares decorrentes da natureza visual-gestual das línguas de sinais e da acentuada escassez de dados anotados [7]. Historicamente, a área progrediu de sistemas baseados em regras (RBMT) e métodos estatísticos (SMT) para a Tradução Automática Neural (NMT), que se estabeleceu como o paradigma dominante na última década [28].

No cenário internacional atual, o estado da arte apoia-se predominantemente em arquiteturas *Transformer* para a modelagem de sequências. Estas são frequentemente integradas a Redes Neurais Convolucionais (CNNs) ou *Vision Transformers* (ViTs), responsáveis pela extração de características visuais e espaciais [13]. A ferramenta *SignJoey* [5] destaca-se como uma referência central neste domínio, oferecendo suporte ao treinamento ponta-a-ponta (*end-to-end*) tanto para reconhecimento quanto para tradução. A modalidade visual impõe complexidades técnicas significativas, tais como a simultaneidade de sinais, que envolve o uso conjunto de mãos, expressões faciais e postura corporal, a utilização de referências espaciais e a ausência de uma forma escrita padronizada [21, 14]. Essas características dificultam a aplicação direta de técnicas de tokenização convencionais, originalmente desenhadas para o processamento de texto linear.

O progresso científico na área é mensurado principalmente através de conjuntos de dados de referência internacional. O RWTH-PHOENIX-Weather 2014T, focado na Língua de Sinais Alemã, é o conjunto de dados mais amplamente utilizado para tarefas de tradução contínua, contendo um corpus de mais de 8.000 sentenças anotadas [4]. Outros conjuntos de dados relevantes incluem o CSL (Chinês) [12] e o LSFB-CONT (Franco-Belga) [11], que

contribuem para a avaliação de modelos ao oferecerem uma maior diversidade de sinalizantes e abrangência de vocabulário.

### 2.2 Cenário Brasileiro

No contexto nacional, a pesquisa em SLMT encontra forte respaldo na legislação vigente, notadamente através da Lei nº 10.436/2002, que reconhece a Libras como meio legal de comunicação e expressão, e do Estatuto da Pessoa com Deficiência (Lei nº 13.146/2015) [16, 17]. Entretanto, o desenvolvimento técnico enfrenta um entrave crítico: a inexistência de *corpora* robustos e acessíveis publicamente. Diferentemente do cenário observado com os conjuntos de dados internacionais, o Brasil carece de uma base de dados pública de Libras que possua escala, granularidade de anotação e cobertura linguística adequadas para o treinamento eficaz de modelos de tradução contínua [1].

As bases de dados existentes, embora valiosas para propósitos específicos, apresentam limitações para a tradução automática ampla. O LIBRAS-UFOP, por exemplo, foca em pares mínimos de sinais isolados capturados via Kinect, sendo útil para distinções finas, mas restrito para a tradução de sentenças completas [6]. O MINDS-Libras constitui um conjunto de dados multimodal com dados de RGB, profundidade e esqueleto, porém restringe-se a 20 sinais e 12 sinalizantes, voltando-se mais ao reconhecimento baseado em sensores [18]. Já o V-Librasil, contendo 1.364 sinais isolados, foi desenhado primariamente para consulta lexical em formato de dicionário, não sendo ideal para tarefas de tradução contínua [20].

Para suprir essa lacuna, iniciativas como as do Centro de Ciência para o Desenvolvimento – Tecnologia Assistiva e Acessibilidade em Libras (CCD-TAAL) têm desempenhado um papel central. O grupo dedica-se à construção de *corpora* Português-Libras e à investigação de abordagens orientadas a dados (*data-driven*). Entre as estratégias adotadas estão o uso de glosas enriquecidas e o desenvolvimento de avatares sinalizantes, visando viabilizar sistemas de tradução mais eficazes e adaptados à realidade brasileira.

## 3 Desenvolvimento

### 3.1 Construção de Corpus

A fundamentação de sistemas de tradução automática baseados em dados (*data-driven*) e a geração de sinalização realista requerem bases de dados de alta fidelidade. Para este fim, foi conduzida a construção de um novo corpus Português-Libras multimodal. O processo de aquisição de dados, ilustrado nas Figuras 1 e 2, utilizou um sistema óptico de captura de movimento (*motion capture*) da Vicon [27], composto por um arranjo de 15 câmeras de infravermelho de alta resolução e baixa latência, permitindo o registro preciso da cinemática corporal e manual dos sinalizantes.

Considerando a importância gramatical das expressões não-manuais na Libras, o sistema foi operado em conjunto com a solução de captura facial dedicada da Faceware [9], montada em capacete (HMC), garantindo a sincronia entre os movimentos dos membros superiores e as expressões faciais.

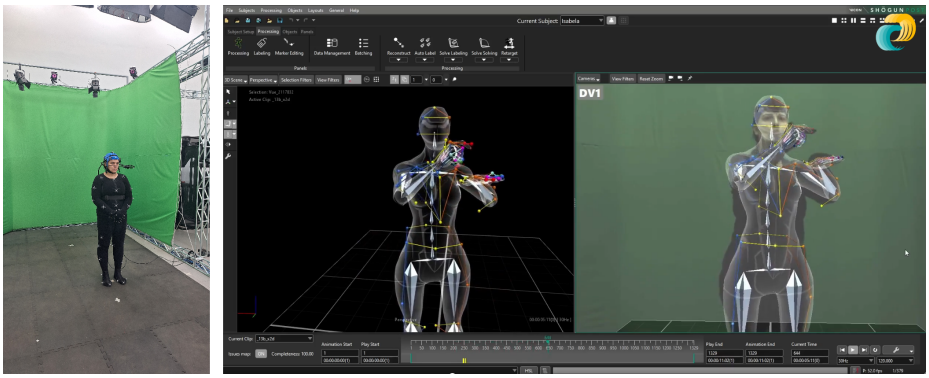


Figura 1: À esquerda, intérprete durante a captura de movimento. À direita, tela do Vicon Shogun Post, durante processamento da captura.

Figura 2: Imagem comparativa dos movimentos e expressões reais vs renderizados no avatar 3D.

O protocolo de coleta envolveu a participação de intérpretes fluentes, resultando no registro de aproximadamente 2400 sentenças de complexidade e extensão variadas. O volume total de dados brutos de captura de movimento processados atingiu a marca de 150 GB. Adicionalmente, foram realizados registros em vídeo (RGB) sincronizados das sinalizações para servir como referência visual para anotação. Este conjunto de dados foi estruturado para viabilizar duas etapas subsequentes críticas: a anotação linguística precisa dos sinais e o treinamento de modelos para a síntese de sinalização através de avatares 3D antropomórficos.

## 3.2 APIs de Integração Web

Visando a integração futura do sistema de tradução automática a web foi desenvolvido um conjunto de ferramentas de software modulares, projetadas para automatizar o fluxo de dados desde a aquisição até a apresentação final ao usuário. Estas ferramentas, denominadas aqui como APIs de Integração, foram implementadas na linguagem Python [25], escolhida

devido à sua robustez no tratamento de dados textuais e vasta disponibilidade de bibliotecas para *web scraping* e automação de navegadores.

A arquitetura da solução foi dividida em dois componentes principais: (i) um mecanismo de coleta de URLs (*crawler*) para mapeamento de domínios; (ii) um processador de conteúdo (*parser*), responsável pela extração textual, segmentação de sentenças e injeção do código do player de acessibilidade. Alternativamente um módulo específico para o tratamento da página do Cardápio do Restaurante Universitário da UNICAMP, pensando-se na aplicação a curto prazo anterior a conclusão do tradutor automático. A seguir, detalha-se o funcionamento e a implementação de cada um destes componentes.

### 3.2.1 Crawler

A primeira etapa do fluxo de processamento consiste na identificação das páginas que compõem o domínio alvo a ser tornado acessível. Para automatizar esta tarefa, foi desenvolvido o *script* `url_collector.py`, que atua como um *Web Crawler* focado na extração de links internos.

O funcionamento do algoritmo inicia-se com a definição de uma URL semente (*seed url*). O *script* utiliza a biblioteca *Requests* [15] para efetuar uma requisição HTTP GET e obter o código fonte HTML da página. Para a interpretação da estrutura do documento, emprega-se a biblioteca *Beautiful Soup* [19], que permite a navegação na árvore DOM (*Document Object Model*) e a busca eficiente por *tags* de *link* (`<a>`).

A lógica de extração implementada no *Crawler* executa os seguintes passos para cada *link* encontrado:

1. **Extração do Atributo:** O algoritmo captura o valor do atributo `href` de todas as *tags* de *link*.

2. **Normalização de URLs:** Utilizando o módulo *urllib* [25], especificamente as funções `urljoin` e `urlparse`, o sistema converte URLs relativas em absolutas, garantindo a integridade dos endereços.

3. **Filtragem de Domínio:** Para evitar que o *Crawler* navegue para sites externos, implementou-se uma verificação que compara o domínio do *link* extraído com o domínio da URL semente. Apenas links pertencentes ao mesmo domínio são mantidos.

4. **Deduplicação:** Utiliza-se uma estrutura de conjunto (*set*) para armazenar as URLs, eliminando automaticamente duplicatas e referências redundantes à mesma página.

O *script* conta ainda com uma interface de linha de comando baseada na biblioteca *argparse*, permitindo ao usuário especificar a URL alvo e o diretório de saída dinamicamente. O resultado final é um arquivo de texto contendo a lista sanitizada de todas as URLs internas encontradas, servindo de insumo direto para o módulo subsequente, o *Parser*.

### 3.2.2 Parser

O componente central da arquitetura é o *Parser*, um *script* em Python responsável por transformar as páginas web brutas coletadas pelo *Crawler* em interfaces acessíveis e interativas. Este módulo opera em duas versões distintas, adaptadas para as modalidades de saída do sistema: reprodução de vídeo pré-gravado (`parser_video.py`) e renderização de avatares 3D em tempo real (`parser_webgl.py`).

O fluxo de processamento inicia-se com a renderização da página alvo. Diferentemente de abordagens estáticas simples, o sistema utiliza a biblioteca *Selenium* [22] para instanciar um navegador real (Google Chrome) e carregar a página. Esta abordagem justifica-se pela necessidade de processar sites dinâmicos modernos, onde o conteúdo é frequentemente gerado via JavaScript no lado do cliente, e para garantir o download correto de todos os ativos (*assets*) visuais como folhas de estilo (CSS) e imagens.

Após a obtenção do DOM (*Document Object Model*) renderizado, o processamento textual é realizado com o auxílio da biblioteca *Beautiful Soup* [19]. O algoritmo identifica elementos de conteúdo textual (como parágrafos, títulos e listas) e aplica uma etapa crítica de segmentação linguística utilizando o *Natural Language Toolkit* (NLTK) [2]. O NLTK permite dividir parágrafos longos em sentenças gramaticais individuais, assegurando que a tradução para Libras ocorra com a granularidade adequada para a compreensão do usuário.

Para cada sentença identificada, o *parser* realiza uma anotação no código HTML, envolvendo o texto em uma estrutura `<span>` com a classe identificadora `.tracked` e um atributo de metadados que vincula aquele texto a um arquivo de mídia específico (vídeo ou animação), gerando uma página modificada, como a ilustrada na Figura 3. Simultaneamente, o sistema gera um arquivo CSV contendo o roteiro de todas as sentenças extraídas, facilitando o gerenciamento da produção do conteúdo em Libras.

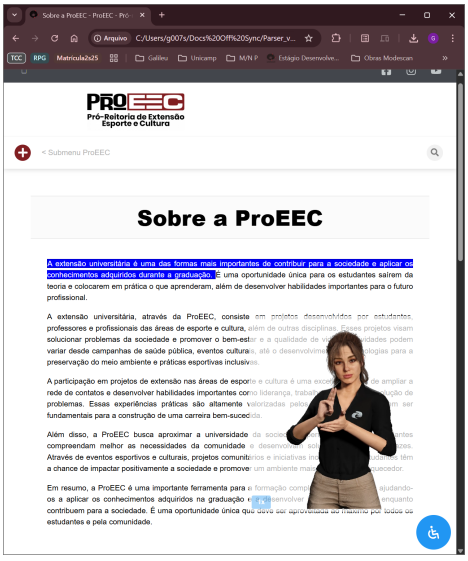


Figura 3: Página da PROEEC após adição de acessibilidade.

**Versão para Vídeo vs. WebGL:** A versão `parser_video.py` prepara a página para a exibição de vídeos no formato WebM, ideal para reprodução com canal alfa (fundo transparente) em navegadores modernos. Já a versão `parser_webgl.py` adapta o sistema para a renderização de computação gráfica. Esta versão altera as referências de mídia para arquivos de animação 3D (`.glb`) e injeta automaticamente as bibliotecas *Three.js* [3] e *GLTFLoader* no cabeçalho da página, habilitando o suporte a gráficos 3D baseados em WebGL diretamente no navegador.

**Interface Front-end (Player Talita):** A interação do usuário com a página adaptada é mediada pelo componente *Player Talita*, implementado como um *widget* da biblioteca *jQuery UI* [23]. O script `talita.js` define uma janela flutuante e redimensionável que se sobrepõe ao conteúdo original, assim como configura um controle de velocidade da sinalização exibida.

O comportamento do player é definido por estilos CSS específicos (`talita.css`), que utilizam posicionamento fixo (`position: fixed`) para garantir que a janela de tradução permaneça visível durante a rolagem da página. Além disso, o arquivo `talita-text-interaction.css` fornece *feedback* visual, destacando as sentenças traduzíveis com bordas interativas ao passar do mouse, indicando intuitivamente ao usuário surdo quais elementos possuem tradução disponível.

A separação entre a lógica de processamento (Python) e a lógica de apresentação (JavaScript/CSS) confere ao sistema alta modularidade, permitindo que a mesma infraestrutura de *parsing* suporte diferentes tecnologias de visualização (vídeo ou avatar 3D) com alterações mínimas no código fonte.

### 3.2.3 Parser para o Cardápio do Restaurante Universitário

Dada a arquitetura específica da página do Cardápio do Restaurante Universitário da UNICAMP (`https://prefeitura.unicamp.br/cardapio/`), que utiliza carregamento dinâmico de conteúdo encapsulado em um *iframe* e navegação por abas controlada via JavaScript, foi necessária a implementação de um módulo de *parsing* especializado. Este componente foi desenvolvido para superar as limitações de *crawlers* estáticos, que não conseguem executar as interações necessárias para acessar o conteúdo dos diferentes dias da semana.

O algoritmo de extração, implementado no *script* dedicado, utiliza a biblioteca *Selenium* para instanciar um navegador completo e simular a interação do usuário. O processo inicia-se com a identificação do *iframe* alvo e a alternância do contexto de execução do *driver*. Em seguida, o sistema mapeia os elementos de navegação (abas correspondentes aos dias) e executa eventos de clique sequenciais, aguardando a renderização do DOM (*Document Object Model*) para cada dia antes de capturar o código HTML resultante.

Para o gerenciamento do conteúdo acessível, visto que a tradução automática por Inteligência Artificial ainda está em fase de desenvolvimento, optou-se por uma abordagem baseada em base de conhecimento. O sistema utiliza a biblioteca *Pandas* [24] para gerenciar um arquivo CSV (`text2video.csv`) que atua como um dicionário de correspondência entre as sentenças em português (ex: nomes de pratos, observações) e os arquivos de vídeo ou

animação correspondentes. Quando uma nova sentença é encontrada no cardápio, o *parser* gera automaticamente um registro de *fallback* utilizando um identificador *hash*, permitindo que a equipe de produção identifique e produza o conteúdo faltante sem interromper o funcionamento do sistema.

O tratamento textual inclui ainda refinamentos específicos por meio de expressões regulares, como a decomposição de datas (convertendo "dd/mm" para dia e mês por extenso, permitindo o reuso de vídeos de meses) e a segmentação de listas baseada na tag HTML `<br>`. Por fim, para garantir a fidelidade visual da página modificada, o *script* converte todos os caminhos relativos de folhas de estilo (CSS) e imagens para URLs absolutas, assegurando que a interface acessível gerada localmente mantenha a identidade visual da página original da universidade, como ilustrado na Figura 4.

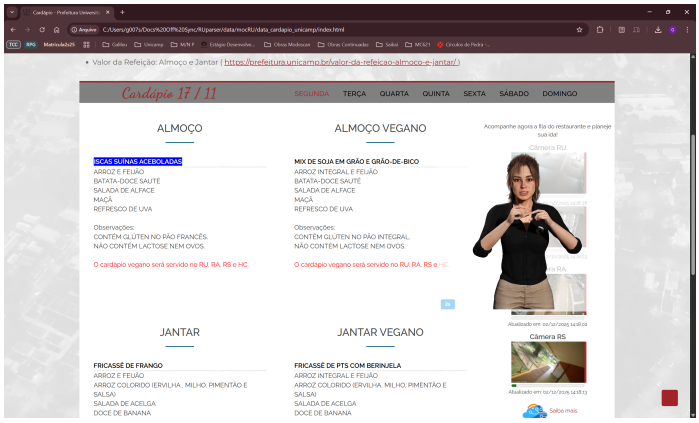


Figura 4: Página do Restaurante Universitário após adição de acessibilidade.

## 3.3 Animação em Tempo Real

Além da captura de dados para processamento *offline* (pós-processamento e anotação), o projeto estabeleceu um fluxo de trabalho (*pipeline*) para a animação de avatares 3D em tempo real. Esta configuração foi desenvolvida em ambiente controlado de laboratório, visando aplicações que exigem *feedback* visual imediato ou interação ao vivo.

O núcleo deste sistema é a *Unreal Engine 5* [8], um motor gráfico de alto desempenho capaz de renderizar personagens digitais com fidelidade fotorealista. Dentro deste ambiente, foi construída uma cena virtual contendo o modelo 3D do avatar humano Clara, devidamente equipado com uma estrutura óssea (*rigging*) compatível com os padrões de captura.

A integração entre os sensores físicos e o ambiente virtual ocorre através de protocolos de transmissão de dados de baixa latência. Para a captura corporal, utilizou-se o plugin *Vicon Live Link*, que transmite as coordenadas espaciais dos marcadores e a rotação das articulações do software de processamento da Vicon, passando por uma operação de *retarget* para o esqueleto do avatar Clara, para a *Unreal Engine*. Simultaneamente, a captura facial é gerida pelo plugin *Faceware Live Link*, que mapeia as expressões faciais do intérprete para os *blendshapes* do avatar em tempo real. Esta arquitetura permite que o avatar mimetize,

com atraso imperceptível, a expressão facial e os sinais manuais executados pelo intérprete humano.

A aplicabilidade deste sistema estende-se a diversos cenários. Primariamente, ele serve como ferramenta de validação durante as sessões de captura, permitindo que o intérprete surdo ou ouvinte verifique a clareza da sinalização do avatar instantaneamente (autoscopia digital). Além disso, viabiliza a atuação de "intérpretes virtuais" em transmissões ao vivo, onde a identidade do sinalizante humano pode ser preservada ou substituída por um personagem estilizado, como observado nas Figuras 5 e 6.

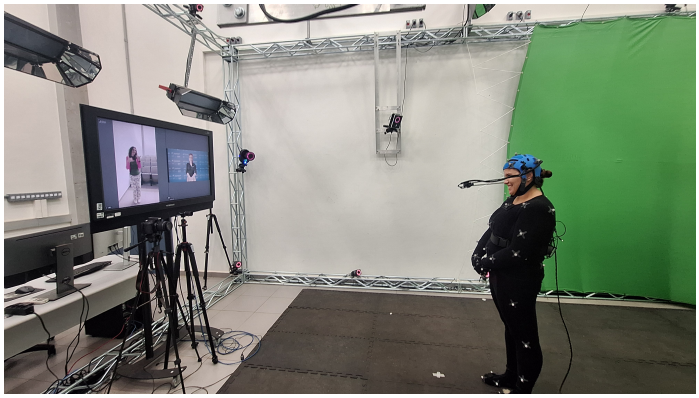

Figura 5: Intérprete no sistema de captura de movimento para animação em tempo real do avatar.

Figura 6: Interação do público com o avatar via vídeo chamada.

A robustez e a viabilidade desta solução foram demonstradas publicamente durante a Reatech Brasil 2025 – Feira Internacional de Tecnologias em Reabilitação, Inclusão e Acessibilidade [10]. Na ocasião, o sistema foi utilizado para apresentar o trabalho desenvolvido pelo CCD-TAAL de tradução em Libras aos visitantes, comprovando a eficácia da integração entre os sistemas de captura de movimento e a renderização gráfica na promoção da acessibilidade comunicacional.

# 4 Resultados e Discussões

A implementação da arquitetura proposta e as atividades de coleta de dados permitiram não apenas a viabilização técnica da ferramenta de acessibilidade, mas também a constituição de ativos de pesquisa significativos para a área de Tradução Automática de Língua de Sinais (SLMT).

O esforço de captura de movimento resultou na consolidação de um corpus Português-Libras de alta fidelidade. O volume de dados brutos processados, totalizando aproximadamente 150 GB e abrangendo cerca de 2400 sentenças, representa um avanço significativo em relação aos recursos atualmente disponíveis no cenário brasileiro.

Ao contrastar este resultado com o estado da arte nacional descrito na literatura, observa-se que o novo corpus supera as limitações de bases como a LIBRAS-UFOP e a V-Librasil. Enquanto estas focam majoritariamente em sinais isolados ou pares mínimos para reconhecimento lexical, o material capturado neste trabalho contempla sentenças completas e contínuas. A inclusão simultânea de dados de cinemática corporal e expressões faciais (via *Faceware*) enriquece a base de dados com as nuances prosódicas e gramaticais essenciais para a modelagem de tradução automática contínua, uma lacuna crítica nos atuais modelos.

O desenvolvimento de dois módulos distintos de visualização — o *parser* baseado em vídeo (`parser_video.py`) e o baseado em computação gráfica (`parser_webgl.py`) — permitiu uma avaliação qualitativa das vantagens e desvantagens de cada abordagem para a acessibilidade Web.

**Abordagem baseada em Vídeo (WebM com Canal Alfa):**

- *Vantagens:* Apresenta fidelidade visual absoluta à sinalização humana original, garantindo naturalidade sem a necessidade de retargeting para avatares. A implementação é tecnicamente mais simples, dependendo apenas de tags HTML5 nativas.
- *Desvantagens:* O consumo de largura de banda é elevado, visto que cada sentença demanda o download de um arquivo de vídeo relativamente pesado. Além disso, a transparência (canal alfa) em vídeos WebM ainda apresenta inconsistências de suporte em alguns navegadores, notadamente no ecossistema Apple (Safari/iOS).

**Abordagem baseada em Avatar 3D (Three.js/WebGL):**

- *Vantagens:* O consumo de banda é drasticamente reduzido após o carregamento inicial do modelo (`avatar.glb`), pois os arquivos de animação contêm apenas dados vetoriais de movimento. Oferece maior flexibilidade, permitindo a troca de avatares ou ângulos de câmera em tempo real.
- *Desvantagens:* Exige maior poder de processamento do dispositivo do usuário (Unidade de Processamento Gráfico - *Graphics Processing Unit* (GPU)) para a renderização em tempo real, o que pode drenar bateria em dispositivos móveis. A qualidade final da sinalização depende da precisão do *rigging* do avatar, podendo ocorrer artefatos visuais ("clipping") se não houver um refinamento manual.

A eficácia do sistema de animação em tempo real foi validada empiricamente durante o evento *Reatech Brasil 2025*, realizado em São Paulo, no dia 08 de novembro de 2025. A demonstração da tradução ao vivo, utilizando a captura de movimento para animar o avatar, obteve receptividade positiva por parte dos visitantes surdos. O *feedback* qualitativo indicou que a preservação das expressões não-manuais, capturadas pelo sistema facial, foi determinante para a compreensibilidade e a aceitação da tecnologia como uma ferramenta de apoio à comunicação, e não apenas uma novidade tecnológica.

Quanto a aplicação da ferramenta na página do Restaurante Universitário da Unicamp evidenciou desafios práticos da implementação em ambientes não controlados. O *parser* desenvolvido demonstrou robustez na navegação e extração de dados através de *Iframes* dinâmicos. Contudo, a estratégia de tradução baseada em dicionário (base de conhecimento `text2video.csv`) revelou limitações de escalabilidade.

Durante os testes de campo, notou-se que, mesmo com acesso prévio à lista oficial de pratos, o cardápio real apresentava itens inéditos ou variações de grafia não previstas (e.g., "Isca de Frango" vs. "Iscas de Frango"). Como o sistema atual não utiliza Inteligência Artificial generativa para a tradução, essas variações resultavam em falhas de correspondência (misses). Para mitigar este problema sem a complexidade de um tradutor neural completo, propõe-se um mecanismo de augmentação automática da base de conhecimento. Verificada a equivalência semântica ou visual do sinal em Libras para as variações textuais, o sistema pode ser atualizado para mapear múltiplas chaves de texto (entradas) para um único arquivo de vídeo (saída). Esta abordagem otimiza o reaproveitamento do acervo de vídeos já produzidos e aumenta a resiliência do sistema frente a inconsistências nos dados de entrada.

## 5 Conclusões

O presente trabalho cumpriu o objetivo de estabelecer a infraestrutura tecnológica fundamental para a viabilização da legendagem automática em Língua Brasileira de Sinais (Libras) em páginas da Internet. As ferramentas desenvolvidas, especificamente os módulos *Crawler* e *Parser*, demonstraram eficácia na automação do processo de adaptação de sites, permitindo a segmentação granular de conteúdo textual e a injeção de interfaces de acessibilidade de forma não intrusiva. A validação prática no portal do Restaurante Universitário da Unicamp demonstrou a robustez da solução, que foi capaz de preservar o *layout* original e garantir a navegação fluida em páginas dinâmicas, representando um avanço significativo em relação a abordagens estáticas.

Adicionalmente, a construção do novo corpus multimodal, composto por aproximadamente 2.400 sentenças e 150 GB de dados de alta fidelidade de corpo e face, constitui um ativo científico de longo prazo que supera as limitações de escala das bases de dados nacionais existentes. Contudo, as validações de campo evidenciaram que a atual estratégia de tradução baseada em base de conhecimento (*dictionary-based translation*) possui limitações de escalabilidade frente à variabilidade da linguagem natural em ambientes não controlados. Esses resultados indicam que, embora a infraestrutura de captura e exibição esteja consolidada, a evolução para uma cobertura universal dependerá da futura integração com modelos de tradução neural treinados sobre o corpus aqui estabelecido.

## Referências

[1] S. Alyami, H. Luqman, and M. Hammoudeh. Reviewing 25 years of continuous sign language recognition research: Advances, challenges, and prospects. *Information Processing & Management*, 61(5):103774, 2024. 2, 3

[2] Steven Bird, Ewan Klein, and Edward Loper. *Natural Language Processing with Python*. O'Reilly Media, Sebastopol, CA, 2009. 6

[3] Ricardo Cabello and Three.js Authors. Three.js – javascript 3d library. https://threejs.org/, 2024. Acessado em: 01 dez. 2024. 7

[4] N. C. Camgoz, S. Hadfield, O. Koller, H. Ney, and R. Bowden. Neural sign language translation. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 7784–7793, 2018. 2

[5] N. C. Camgoz, O. Koller, S. Hadfield, and R. Bowden. Sign language transformers: Joint end-to-end sign language recognition and translation. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10023–10033, 2020. 2

[6] L. R. Cerna, E. E. Cardenas, D. G. Miranda, D. Menotti, and G. Camara-Chavez. A multimodal LIBRAS-UFOP brazilian sign language dataset of minimal pairs using a microsoft kinect sensor. *Expert Systems with Applications*, 167:114179, 2021. 3

[7] M. De Coster, D. Shterionov, M. Van Herreweghe, and J. Dambre. Machine translation from signed to spoken languages: State of the art and challenges. *Universal Access in the Information Society*, 23(3):1305–1331, 2024. 2

[8] Epic Games. Unreal engine 5: The world's most open and advanced real-time 3d creation tool. https://www.unrealengine.com/, 2024. Acessado em: 01 dez. 2024. 8

[9] Faceware Technologies. Faceware facial motion capture system. https://facewaretech.com/, 2024. Acessado em: 01 dez. 2024. 3

[10] Fiera Milano Brasil. Reatech brasil 2025: Feira internacional de tecnologias em reabilitação, inclusão e acessibilidade. https://reatechbrasil.com.br/, 2025. São Paulo, SP. Acessado em: 01 dez. 2024. 9

[11] J. Fink, B. Frénay, L. Meurant, and A. Cleve. LSFB-CONT and LSFB-ISOL: Two new datasets for vision-based sign language recognition. In *International Joint Conference on Neural Networks*, pages 1–8. IEEE, 2021. 2

[12] J. Huang, W. Zhou, Q. Zhang, H. Li, and W. Li. Video-based sign language recognition without temporal segmentation. In *AAAI Conference on Artificial Intelligence*, volume 32, 2018. 2

[13] A. Khan, S. Jin, G.-H. Lee, G. E. Arzu, T. N. Nguyen, L. M. Dang, W. Choi, and H. Moon. Deep learning approaches for continuous sign language recognition: A comprehensive review. *IEEE Access*, 2025. 2

[14] C. Loos, A. German, and R. P. Meier. Simultaneous structures in sign languages: Acquisition and emergence. *Frontiers in Psychology*, 13:992589, 2022. 2

[15] Kenneth Reitz. Requests: Http for humans. https://requests.readthedocs.io, 2024. Acessado em: 01 dez. 2024. 5

[16] República Federativa do Brasil. Lei nº 10.436, de 24 de abril de 2002. dispõe sobre a língua brasileira de sinais - libras e dá outras providências, 2002. Diário Oficial da União, Brasília, DF. 3

[17] República Federativa do Brasil. Lei nº 13.146, de 6 de julho de 2015. institui a lei brasileira de inclusão da pessoa com deficiência (estatuto da pessoa com deficiência), 2015. Diário Oficial da União, Brasília, DF. 3

[18] T. M. Rezende. *Reconhecimento Automático de Sinais da Libras: Desenvolvimento da Base de Dados MINDS-Libras e Modelos de Redes Convolucionais*. PhD thesis, Universidade Federal de Minas Gerais, 2021. 3

[19] Leonard Richardson. Beautiful soup documentation. https://www.crummy.com/software/BeautifulSoup/bs4/doc/, 2024. Acessado em: 01 dez. 2024. 5, 6

[20] A. J. Rodrigues. V-LIBRASIL: Uma base de dados com sinais na língua brasileira de sinais (libras). Master's thesis, Universidade Federal de Pernambuco (UFPE), Recife, Brazil, 2021. 3

[21] W. Sandler and D. Lillo-Martin. Natural sign languages. In *The Handbook of Linguistics*, pages 533–534. 2003. 2

[22] Selenium Project. Selenium browser automation. https://www.selenium.dev/, 2024. Acessado em: 01 dez. 2024. 6

[23] The jQuery Foundation. jquery ui. https://jqueryui.com/, 2024. Acessado em: 01 dez. 2024. 7

[24] The Pandas Development Team. pandas-dev/pandas: Pandas, February 2020. Acessado em: 01 dez. 2024. 7

[25] Guido Van Rossum and Fred L. Drake. *Python 3 Reference Manual*. CreateSpace, Scotts Valley, CA, 2009. 4, 5

[26] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin. Attention is all you need. In *Advances in Neural Information Processing Systems*, volume 30, 2017. 2

[27] Vicon Motion Systems. Vicon optical motion capture systems. https://www.vicon.com/, 2024. Acessado em: 01 dez. 2024. 3

[28] H. Wang, H. Wu, Z. He, L. Huang, and K. W. Church. Progress in machine translation. *Engineering*, 18:143–153, 2022. 2

[29] A. Way, L. Leeson, and D. Shterionov. *Sign Language Machine Translation*. Springer, 2024. 2