www.revistaft.com.br

Revista FT | ISSN 1678-0817 | v. 30, n. 164, 2026 | DOI: 10.69849/t4a8nf81

Desenvolvimento de uma plataforma educacional gamificada para auxílio no ensino de Libras com o uso de visão computacional: Handsign

Development of a gamified educational platform to support brazilian sign language (Libras) teaching using computer Vision: Handsign

Heytor Kyoshi Sato[1]
Evandro Ijanques Junior[2]
William Galvonas Apuzzo Miyaura[3]
Kelvin Fernandes Silva[4]
Carlos López Noriega

Resumo

A Língua Brasileira de Sinais (Libras) constitui o principal meio de comunicação da comunidade surda no Brasil, sendo reconhecida oficialmente pela Lei nº 10.436/2002 (BRASIL, 2002) e regulamentada pelo Decreto nº 5.626/2005 (BRASIL, 2005), que estabelece diretrizes para sua difusão e ensino. Apesar do reconhecimento, ainda há desafios significativos na difusão e no ensino da língua. Nesse contexto, o presente artigo tem como objetivo desenvolver uma aplicação digital para auxiliar no ensino de libras que utiliza técnicas de visão computacional para o reconhecimento e validação de sinais feitos pelo usuário, aliadas a conceitos de gamificação. A aplicação terá como finalidade fornecer feedback imediato durante a prática das lições, permitindo assim que o aluno consiga aperfeiçoar seus sinais e corrigir pequenos erros. Metodologicamente o trabalho caracteriza-se como uma pesquisa qualitativa aplicada de natureza experimental que envolve tanto revisão bibliográfica e documental, quanto desenvolvimento de um protótipo e realização de testes com usuários.

[1] Faculdade Engenheiro Salvador Arena, Centro Educacional da Fundação Salvador Arena - São Bernardo do Campo - SP
[2] Faculdade Engenheiro Salvador Arena, Centro Educacional da Fundação Salvador Arena - São Bernardo do Campo - SP
[3] Faculdade Engenheiro Salvador Arena, Centro Educacional da Fundação Salvador Arena - São Bernardo do Campo - SP
[4] Faculdade Engenheiro Salvador Arena, Centro Educacional da Fundação Salvador Arena- - São Bernardo do Campo - SP

Palavras-chave: Libras. Visão computacional. Gamificação. Reconhecimento de sinais de Libras. Tecnologia educacional.

Abstract

Brazilian Sign Language (Libras) is the primary means of communication for the deaf community in Brazil and was officially recognized by Law No. 10.436/2002. Despite this recognition, significant challenges still exist in the dissemination and teaching of the language. In this context, this paper aims to describe the development of a digital application for assisting the teaching of Libras that uses computer vision techniques to recognize and validate the signs performed by users, combined with gamification. The application was designed to provide immediate feedback during lesson practice, allowing students to improve their signing skills and correct small mistakes. Methodologically, the study is characterized as applied experimental research involving literature review, prototype development, and user testing.
Keywords: Brazilian Sign Language. Computer Vision. Gamification. Sign Recognition. Educational Technology.

1 Introdução

A língua brasileira de sinais (Libras) desempenha um crucial papel na inclusão da comunidade de deficientes auditivos, sendo reconhecida como meio legal de comunicação e expressão pela Lei nº 10.436 de 2002 (BRASIL, 2002), e regulamentada pelo Decreto nº 5.626/2005 (BRASIL, 2005), que estabelece diretrizes para sua difusão e ensino. De acordo com George Mead (1934), no contexto do interacionismo simbólico da Escola de Chicago, a sociedade é compreendida como resultado de um processo contínuo de interação mediado por símbolos e significados compartilhados. Entretanto, apesar do reconhecimento oficial de Libras como meio de comunicação, a barreira comunicativa entre surdos e ouvintes ainda se mantém, dificultando o acesso desta comunidade a serviços básicos e a plena integração ao restante da sociedade.

De acordo com o Censo Demográfico de 2010 do Instituto Brasileiro de Geografia e Estatística (IBGE, 2022), cerca de 2,4 milhões de pessoas apresentam deficiência auditiva severa ou completa. Apesar da magnitude desse número, a Pesquisa Nacional de Saúde (PNS) de 2019 estima que apenas cerca de 287 mil surdos sejam sinalizantes, isto é, usuários de Libras (IBGE, 2019). Isso equivale a cerca de 12% da população com deficiência auditiva severa, o que indica que a grande maioria dos surdos

possuem dificuldades na sua comunicação do dia a dia e evidencia uma lacuna no ensino de Libras no país.Essa grande diferença entre o número de falantes de libras e o número de deficientes auditivos no Brasil demonstra uma clara lacuna no ensino da língua no país. Segundo Souza e Santos (2023), o ensino de Libras ainda enfrenta desafios relacionados à inclusão escolar, à formação docente e às práticas pedagógicas, o que pode limitar o acesso à aprendizagem e o desenvolvimento da autonomia dos estudantes.

Neste cenário, surge o problema que norteia este trabalho: Como uma plataforma lúdica pode auxiliar na aprendizagem da língua brasileira de sinais? A dificuldade em obter retorno imediato durante a prática é um grande obstáculo para quem busca aprender Libras de forma autodidata, ou para pessoas que buscam complementar suas aulas convencionais, pois é comprovado que estratégias de aprendizagem ativa, nas quais o aluno recebe retorno sobre sua performance durante a execução das tarefas, demonstram resultados significativamente superiores aos modelos de aprendizagem puramente passivos (Freeman et al., 2014). Sem uma forma de confirmar a configuração da mão, do movimento e dos pontos de articulação corretos, na ausência de um docente, o aluno pode consolidar erros que prejudicam sua comunicação.

Como forma de evitar este problema, formulam-se as seguintes hipóteses: Adoção da visão computacional para validação de exercícios práticos em tempo real e o uso do elemento lúdico para incentivar a retenção e o empenho dos usuários, pois a gamificação, em contextos educativos, utiliza mecanismos provenientes dos jogos para promover o envolvimento emocional, a motivação e o engajamento dos indivíduos, além de estimular o pensamento sistemático voltado à resolução de problemas, favorecendo a criação de ambientes de aprendizagem mais eficazes e contribuindo para a otimização da curva de aprendizado dos alunos (FADEL et al., 2014).

O objetivo geral deste Projeto Final de Curso (PFC) é desenvolver uma aplicação digital de ensino de libras que utiliza técnicas de visão computacional para o reconhecimento e validação de sinais feitos pelo usuário, aliadas a conceitos de gamificação. Como objetivos específicos destacam-se: Desenvolver um algoritmo de identificação de sinais, que deverá comparar os sinais capturados por meio de visão computacional com os sinais contidos na base de dados construída no projeto. Desenvolver e implementar uma plataforma de ensino, utilizando-se de boas práticas de design e desenvolvimento de software e avaliar os usuários quanto ao uso da gamificação na plataforma por meio de testes e questionários.

A importância deste projeto fundamenta-se em sua relevância social e tecnológica, permitindo auxiliar na redução das barreiras comunicacionais entre surdos e ouvintes. Tecnologicamente, propõe-se

uma solução de aplicação que integra reconhecimento de padrões visuais baseados em gestos dos usuários, ampliando as possibilidades de uso de inteligência computacional em contextos educacionais.

O escopo deste trabalho está delimitado a um conjunto de 51 sinais da Libras, sendo 26 letras do alfabeto manual, além de outros 25 sinais adicionais de temas diversos, como saudações, frutas e outros. A análise e a validação geométrica concentram-se exclusivamente nos componentes manuais do sinal (configuração, orientação, direção e movimento da mão), não contemplando marcadores não manuais como expressões faciais, que também compõem a gramática da Libras, mas cuja avaliação automática exigiria um modelo e um gabarito de expressão próprios não previstos no escopo deste projeto.

2. Fundamentação teórica

2.1 Língua Brasileira de Sinais (Libras)

A Língua Brasileira de Sinais (Libras) é reconhecida como forma oficial de comunicação das pessoas surdas no Brasil, conforme a Lei nº 10.436 de 2002 (BRASIL 2002). Ela é uma língua visual, baseada em movimentos e expressões, com regras próprias, usada para comunicação e interação social. O Decreto nº 5.626 de 2005 (BRASIL 2005) complementa essa lei ao definir como a Libras deve ser ensinada e utilizada em instituições públicas e privadas.

Diferentemente da língua portuguesa, que faz uso da modalidade oral-auditiva, Libras utiliza o canal visual-espacial para a construção das sentenças, implicando no uso de movimentos com as mãos, expressões faciais e corporais como elementos da comunicação. Segundo Ronice Müller de Quadros (2004), Libras é uma língua natural, apresentando todos os níveis linguísticos encontrados nas línguas orais, tais como fonologia, morfologia, sintaxe e semântica, o que comprova sua complexidade e autonomia linguística.

2.2 Gamificação

Segundo Kevin Werbach e Dan Hunter (2012), a gamificação envolve o uso de componentes como pontos, níveis, desafios e recompensas para tornar as atividades do cotidiano mais atrativas e interativas. Ou seja, a aplicação desses elementos e mecânicas podem ser aplicados em contextos que não sejam necessariamente jogos.

2.2.1 Gamificação na educação

A utilização de mecânicas como desafios, recompensas e progressão pode incentivar o engajamento contínuo dos estudantes, promovendo uma experiência de aprendizagem mais interativa (Werbach & Hunter 2012). Esses elementos permitem que os alunos acompanhem seu desempenho e se sintam motivados a avançar em suas atividades. Dessa forma, a gamificação pode ser utilizada como uma estratégia para tornar o processo de ensino mais dinâmico e envolvente.

2.3 Visão computacional

A visão computacional envolve métodos e técnicas que possibilitam a extração, análise e compreensão de dados visuais, permitindo que máquinas reconheçam padrões, objetos, movimentos e características presentes em imagens digitais (Richard Szeliski 2022). Com essa funcionalidade, a visão computacional pode contribuir para a compreensão e a obtenção de informações por meio de imagens e vídeos que podem ser utilizadas no desenvolvimento de ferramentas.

Dentro desse campo, o rastreamento de mãos é uma tarefa de especial interesse para aplicações voltadas às línguas de sinais. Na Libras, o significado de um sinal é construído por parâmetros como a configuração de mão, a locação, o movimento e a orientação da palma (QUADROS; KARNOPP, 2004), todos ligados à posição e à forma das mãos no espaço. Estimar esses parâmetros a partir de vídeo é difícil, porque a mão tem muitos graus de liberdade, os dedos se sobrepõem e se ocultam entre si (auto oclusão) e as condições de iluminação, fundo e câmera variam entre as situações.

2.3.1 MediaPipe

Para lidar com esses desafios, este trabalho utiliza o MediaPipe, um framework de código aberto desenvolvido pelo Google para construir pipelines de percepção sobre dados de vídeo em tempo real (LUGARESI et al., 2019). Em particular, utiliza-se a solução MediaPipe Holistic Landmarks, que integra em um único pipeline os modelos de estimativa de pose corporal, de rastreamento das mãos e de malha facial, produzindo, a partir de um vídeo, até 553 pontos de referência (landmarks) tridimensionais por quadro. Esses pontos se dividem em 33 da pose corporal, baseados no modelo BlazePose; 21 para cada uma das duas mãos e 478 da malha facial.

Essa abordagem é adequada à Libras porque captura, em uma única passagem, os elementos que compõem um sinal: as mãos fornecem a configuração e a orientação da palma, a pose corporal fornece a locação e o movimento dos braços em relação ao corpo, e a malha facial fornece as expressões não

manuais. Além disso, como a saída é um conjunto de coordenadas, e não a imagem bruta, o sistema passa a trabalhar com uma representação compacta, menos sensível a fatores como cor de pele, vestimenta ou fundo, e leve o bastante para execução em tempo real.

3. Trabalhos Relacionados

3.1 Libras4All: sistema com gamificação para apoio na aprendizagem de libras

O trabalho apresenta o desenvolvimento de um sistema web/mobile voltado ao apoio na aprendizagem de Libras por ouvintes, utilizando gamificação como estratégia motivacional. A aplicação mobile é composta por três jogos — Quiz, Meteoro e O Mestre Mandou — e integra técnicas de inteligência artificial para reconhecimento de sinais estáticos por meio da câmera do dispositivo. Para o jogo O Mestre Mandou, foi treinado um modelo de detecção de imagem baseado em transfer learning com a arquitetura SSD MobileNet V2 FPNLite, utilizando 600 imagens supervisionadas divididas em 4 classes (letras A, B, C e D). O modelo atingiu precisão de 80,2% e recall de 85,1% com 12.000 train steps. A avaliação com o público geral resultou em nota média de 4,51/5,0, e os testes em cenário educacional indicaram potencial do sistema para auxiliar no ensino a distância de Libras. Os autores concluem que os resultados são satisfatórios e que o uso combinado de gamificação e inteligência artificial se mostrou eficaz para apoiar o aprendizado de Libras, esperando-se com o sistema um maior interesse dos ouvintes pela comunicação e cultura surda.

3.2 Luva para reconhecimento da Língua brasileira de sinais (Libras) mediante sensoriamento e inteligência artificial através de um aplicativo móvel

O trabalho descreve o desenvolvimento de um protótipo de luva sensorizada para reconhecimento do alfabeto e de palavras em Libras, com tradução em texto e áudio via aplicativo móvel. A luva utiliza um microcontrolador ESP-32, cinco sensores de flexão SpectraFlex, um sensor inercial MPU-6050 com acelerômetro e giroscópio, e três chaves táteis para diferenciação de gestos similares. Os dados capturados são transmitidos ao aplicativo via Bluetooth Low Energy e processados por algoritmos de machine learning — SVM, Decision Tree, Random Forest e Logistic Regression — treinados com scikit-learn e executáveis localmente via ML.NET, em formato ONNX, ou remotamente por uma API Flask hospedada no Microsoft Azure. Os experimentos com 4 voluntários indicaram que entre 16 e 24 amostras por classe representam o volume ideal de treinamento. A abordagem ensemble learning obteve

98,21% de precisão, contra 94,64% do melhor modelo isolado (Random Forest), com tempo médio de resposta de 5,46 ms no processamento local mobile.

4 Metodologia

O presente trabalho caracteriza-se como uma pesquisa de abordagem qualitativa e de natureza aplicada, suportada por meio de pesquisas bibliográficas e documentais. Esta classificação decorre do fato de que o projeto não se limita apenas à discussão teórica do ensino de Libras, mas propõe o desenvolvimento de uma aplicação gamificada com uso de visão computacional para reconhecimento e validação de sinais, seguida de uma pesquisa de satisfação com os usuários.

A pesquisa possui abordagem qualitativa no sentido em que busca compreender a contribuição da plataforma para o processo de ensino e aprendizagem de Libras, considerando aspectos como usabilidade, percepção dos usuários, engajamento proporcionado pelos elementos de gamificação e dificuldades encontradas durante a interação com a aplicação. Essa dimensão permite interpretar a experiência dos participantes e identificar pontos de melhoria.

Quanto à natureza, esta pesquisa é classificada como aplicada, uma vez que tem como finalidade desenvolver uma solução tecnológica voltada a um problema concreto: a dificuldade de aprendizagem e prática de Libras. Ao invés de produzir apenas conhecimento teórico, o trabalho busca gerar um artefato funcional, com potencial de uso em contextos educacionais.

4.1 Quanto aos procedimentos

4.1.1 Pesquisa bibliográfica

A pesquisa bibliográfica fundamenta teoricamente o trabalho a partir da análise de artigos científicos, teses e trabalhos acadêmicos relacionados ao ensino de Libras, à gamificação em contextos educacionais, à visão computacional e ao reconhecimento de gestos. Essa etapa fornece base conceitual para a definição dos requisitos da aplicação e para a interpretação dos resultados.

4.1.2 Pesquisa documental

A pesquisa documental é empregada na análise de documentos normativos e institucionais relevantes ao tema, como documentos técnicos, repositórios, materiais de apoio e documentação das ferramentas utilizadas no desenvolvimento. Esses materiais auxiliam tanto na contextualização do problema quanto na definição dos recursos computacionais adotados.

4.1.3 Pesquisa com survey

A pesquisa de satisfação será realizada por meio de um formulário digital utilizando-se da ferramenta Microsoft Forms, estruturado em 10 questões, sendo 9 de escala Likert de 5 pontos, com o número 1 representando total discordância e o número 5 representando total concordância. No final, há também uma questão discursiva. O Quadro 1 apresenta as dimensões avaliadas pelo instrumento.

Quadro 1 – Dimensões do instrumento de avaliação de usabilidade

Nº	Dimensão	Questão
1	Facilidade de início	Consegui começar a usar o aplicativo e entender como ele funciona sem precisar de ajuda.
2	Clareza dos exemplos	Os exemplos dos sinais (vídeos ou animações) são claros e me permitem entender bem o movimento, a posição das mãos e a expressão.
3	Uso da câmera	Foi fácil posicionar meu corpo e minhas mãos de modo que a câmera me captasse corretamente.
4	Precisão da validação	A avaliação do aplicativo sobre o meu gesto me pareceu justa e condizente com o que eu realmente fiz.
5	Clareza do feedback	O resultado da validação (nota, porcentagem, cores ou mensagens) foi fácil de entender e me mostrou o que eu precisava corrigir.
6	Tempo de resposta	O aplicativo respondeu rapidamente após eu realizar o gesto, sem travamentos ou demora excessiva.
7	Fluidez do fluxo de aprendizagem	A sequência "ver o exemplo → fazer o gesto → receber a validação" é natural e me ajudou a aprender.
8	Motivação e confiança	Sinto que o aplicativo me motiva a continuar praticando e me deixa mais confiante para usar Libras.
9	Robustez em diferentes condições	O aplicativo funcionou bem mesmo com variações de iluminação, fundo do ambiente ou distância da câmera, e as mensagens de erro me ajudaram a resolver os problemas.
10	Sugestões de melhoria (discursiva)	O que você mudaria ou acrescentaria para tornar o aplicativo mais fácil de usar e mais eficaz no aprendizado de Libras?

Fonte: Autoria própria (2026)

A participação é voluntária e a coleta é anônima: o formulário não solicita nome, e-mail ou qualquer outro dado de identificação, com as respostas sendo utilizadas exclusivamente para fins de avaliação e melhoria do aplicativo. Antes do início da coleta, os participantes recebem uma descrição do objetivo da pesquisa, do procedimento e da política de anonimato diretamente na abertura do formulário, funcionando como consentimento informado por adesão.

4.2 Criação da base de dados e armazenamento

4.2.1 Descrição da biblioteca de gestos

Para obtenção dos vídeos e modelos de movimentação 3D, foi utilizada a ferramenta de tradução para Libras chamada VLibras. Essa ferramenta é disponibilizada para acesso público por meio da plataforma do governo federal GOV.BR e acessível por meio do endereço: https://www.gov.br/governodigital/pt-br/acessibilidade-e-usuario/vlibras.

A biblioteca utilizada possui mais de 21mil sinais cadastrados e é atualizada constantemente com a assistência de colaboradores que sugerem sinais a serem adicionados e são avaliados por um time de especialistas os quais indicam se o sinal está correto e pronto para utilização na ferramenta.

Cada sinal incluído na base do VLibras passa pela avaliação de três especialistas antes de sua publicação, o que confere ao gabarito um processo de validação prévio. Adicionalmente, a ferramenta é mantida por meio de parceria institucional entre o Ministério da Gestão e Inovação em Serviços Públicos (MGISP), por intermédio da Secretaria de Governo Digital (SGD), o Ministério dos Direitos Humanos e da Cidadania (MDHC), por meio da Secretaria Nacional dos Direitos da Pessoa com Deficiência (SNDPD), e a Universidade Federal da Paraíba (UFPB), através do Laboratório de Aplicações de Vídeo Digital (LAVID).

4.2.2 Processamento dos gestos para armazenamento

Para a construção da base de dados utilizada neste trabalho, foram escolhidos 51 sinais do vocabulário de Libras, as quais foram feitas as extrações dos principais pontos e coordenadas de referência dos sinais obtidos do VLibras em formado MP4. Para essa extração, é utilizada a biblioteca MediaPipe, a qual, por meio do uso de visão computacional, permite a identificação e construção dos pontos e vetores que formam os gestos executados e presentes nos sinais.

Após a extração, os pontos e suas coordenadas são agrupados em instâncias JSON e organizados em um formato relacional na base de dados, utilizando-se de um código de identificação único, o nome do sinal que está sendo representado e o conteúdo JSON que forma os gestos executados, os quais serão posteriormente utilizados como gabarito para os exercícios.

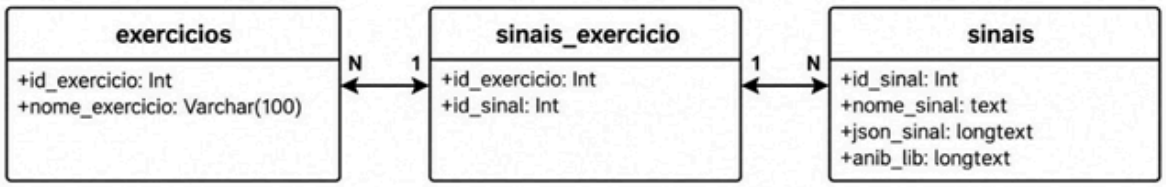
4.2.3 Descrição da base de dados

Para armazenamento das instâncias JSON geradas, foi utilizada uma aplicação MySQL presente em uma máquina virtual provisionada na Oracle Cloud Infraestructure. A plataforma da Oracle foi

escolhida por demonstrar escalabilidade e disponibilidade para utilização e requisição da aplicação desenvolvida, com a atual estrutura lógica de banco de dados representada pela figura 1.

O acesso à máquina virtual é realizado remotamente por meio do software PuTTY, utilizando-se do protocolo SSH (Secure Shell) com autenticação baseada em chave criptográfica.

Figura 1 - Representação visual da estrutura lógica do banco de dados MySQL utilizado no projeto



Fonte: Autoria própria (2026)

4.3 Desenvolvimento da aplicação

O software tem como objetivo disponibilizar um sistema de reconhecimento de gestos aplicado ao ensino de Libras por meio de uma aplicação interativa, permitindo que os usuários realizem exercícios práticos baseados em sinais da língua e recebam feedback em tempo real sobre sua execução. A aplicação integra conceitos de visão computacional e gamificação, promovendo assistência no aprendizado progressivo por meio de trilhas separadas por áreas do conhecimento.

4.3.1 Criação do Back-End e API

O back-end da plataforma foi estruturado para atuar como a camada de inteligência e integração entre a interface do usuário, desenvolvida em Godot Engine, e a base de dados hospedada em nuvem. A comunicação entre essas camadas ocorre por meio de uma API (Application Programming Interface), responsável por gerenciar o fluxo de informações e o progresso do usuário nas trilhas de aprendizado.

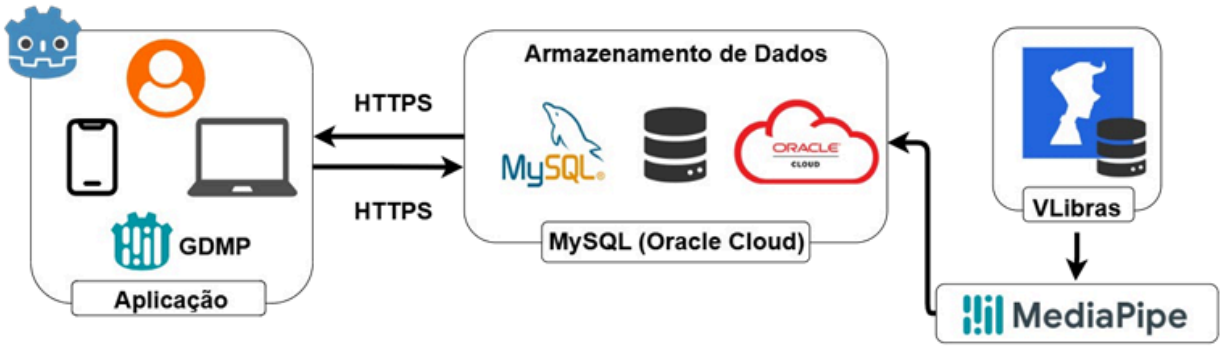
O fluxo de operação inicia-se quando o usuário seleciona uma lição na interface móvel. Nesse momento, a aplicação realiza uma requisição do tipo GET contendo os identificadores (IDs) de cinco exercícios vinculados àquela lição. Esta solicitação é interceptada por um script de processamento hospedado em uma instância da Oracle Cloud Infrastructure. O script atua como um mediador, realizando

a consulta estruturada ao banco de dados MySQL para recuperar as informações dos sinais correspondentes. Após o retorno do banco de dados, o back-end processa e encaminha os dados para a aplicação.

Nenhuma informação de imagem do usuário é enviada para o servidor, pois toda a validação dos exercícios realizados pelo usuário é feita localmente na aplicação, sendo necessária apenas a requisição da instância JSON com os dados de gabarito do exercício.

As etapas descritas estão presentes visualmente na arquitetura do projeto, seguindo o modelo em 3 camadas para aplicações web representada pela Figura 2.

Figura 2 – Representação visual da arquitetura do projeto em 3 camadas



Fonte: Autoria própria (2026)

4.3.2 Criação do Front-End

A interface do usuário foi desenvolvida com a ferramenta de código aberto Godot Engine. A escolha da ferramenta se dá por sua capacidade de desenvolvimento multiplataforma de código aberto, além de recursos nativos para manipulação gráfica, animações e interações em tempo real.

4.3.2.1 Telas da aplicação

No front-end da aplicação, foram desenvolvidas páginas específicas como uma trilha de aprendizado e telas de exercício para a realização de exercícios práticos, estruturados de forma a proporcionar uma progressão guiada.

A aplicação realiza requisições à API desenvolvida no back-end e hospedada em uma aplicação cloud, responsável por gerenciar dados dos usuários, progresso nas atividades e conteúdo dos exercícios, garantindo a persistência e sincronização das informações no sistema.

A página de trilha de aprendizado organiza os conteúdos em níveis sequenciais, permitindo ao usuário acompanhar sua evolução de forma contínua. Já as telas de exercícios integram recursos de visão computacional que utilizam a câmera do dispositivo para capturar os gestos do usuário que serão disponibilizados para validação pelo comparador de sinais desenvolvido, possibilitando feedback ao usuário.

4.4 Validador geométrico de sinais

O validador recebe duas gravações já convertidas em séries de *landmarks* pelo HollisticLandmarker do MediaPipe, uma com a execução do aluno e outra com o gabarito, previamente extraído da base do VLibras, e devolve internamente uma nota de similaridade por grupo circulatório e uma nota global para o aluno. Cada gravação, é um objeto {video_info, frames}, em que cada quadro carrega um timestamp em mili segundos. O processamento é determinístico e puramente geométrico, sendo ele organizado em cinco etapas:

Quadro 2 – Passo a passo do validador geométrico

Etapa	**Atuação**	**Invariância introduzida**
Suavização	Filto One-Euro, por landmark e por eixo.	Tremor do MediaPipe
Normalização	Reexpressa os *landmarks* numa base afim construida a partir do tronco do usuário	Formato da imagem, distância da câmera, inclinação e translação do aparelho
Extração de features idiossincráticas	Os pares de *landmarks* tornam-se vetores unitários, as mãos ganham a normal da palma e a direção global	Comprimentos e escalas dos Membros
Alinhamento temporal	DTW multivariado com recuperação de caminho ótimo	Taxas de quadros inconstantes, atrasos de reação e velocidade de execução
Agregação das notas	Ângulos médios por ossos ao longo do caminho	

Fonte: Autoria própria (2026)

4.4.1 Ancoragem nos parâmetros

Os desenhos do validador partem de parâmetros intrínsecos a cada sinal em Libras. Sendo elas:

Quadro 3 – Parametrização dos pesos por grupo

Parâmetro	Medição	Peso Agregado ao Grupo
Configuração da mão	Vetores unitários das 15 falanges de cada mão	1.2 - 1.5 por osso
Orientação da palma	Normal da palma (Produto vetorial entre o pulso e a base do dedo médio e o pulso e a base do polegar)	3.5
Direcionalidade da mão	Alinhamento por DTW e comprimento total da trajetória	1.5
Movimento	Indiretamente pelos ossos do braço com referencial no tronco	0.6-1,3
Ponto de articulação		
Expressões faciais	Não avaliado*	

Fonte: Autoria própria (2026)

*Embora o HollisticLandmarker forneça landmarkers faciais, a avaliação de expressão exige um modelo de análise facial próprio e um gabarito de expressão, o que extrapola o escopo do trabalho.

4.4.2 Suavização temporal

4.4.2.1 A escolha do filtro

A estimativa de *landmarks* do Media PIpe oscila entre quadros consecutivos mesmo com o usuário estando imóvel. Esta oscilação é pequena em valor absoluto, mas as características do validador a amplificam: a normal da palma é um produto vetorial de 3 pontos vizinhos, um deslocamento em qualquer um deles modifica a normal em vários graus. Sem uma filtragem, o termo de maior peso acaba também virando o com mais ruído.

Um filtro passa-baixa de corte fixo não resolve o problema, porque teria de escolher entre dois erros: Um corte muito baixo suaviza o tremor mas atrasa e amortece o gesto. Já um corte alto preserva o gesto mas mantém o tremor. Foi adotado o filtro One-Euro (CASIEZ; ROUSSEL; VOGEL, 2012), cuja ideia central é fazer a frequência do corte ser variável com a velocidade do sinal, calculada conforme a Equação 1:

$$\mathbf{F}c(t) = fcmin + \beta|\dot{x}(t)| \quad \alpha(fc, \Delta t) = \frac{1}{\left(1 + \frac{\tau}{\Delta t}\right)} \quad \tau = \frac{1}{2\pi fc} \quad \hat{x}(t) = \alpha x(t) + (1 - \alpha)\hat{x}(t-1) \quad \textbf{(1)}$$

Com o landmark parado, a velocidade estimada é próxima de zero e o corte cai para fcmin e a filtragem é agressiva. Durante o gesto, a velocidade cresce, o corte se abre e o filtro deixa de atrasar o movimento. A derivada que pilota o corte é ela mesma suavizada por um filtro passa-baixa de corte fixo em 1Hz, porque a velocidade crua é ruidosa demais para servir de controle.

Cada escalar é filtrado de forma independente, um filtro por *landmark* e eixo.

4.4.3 Normalização espacial pelo referencial do tronco

4.4.3.1 O problema do formato do vídeo

O MediaPipe retorna coordenadas normalizadas pela imagem: $x = px / W$ e $y = py / H$. Quando $H \neq W$, esse espaço é anisotrópico, ou seja, as distancias físicas retratadas valem valores diferentes a depender da direção, logo ângulos medidos nele não são grandezas físicas.

O efeito não é nem um pouco sútil. Um vetor físico em 45 graus é medido como –29,4 graus em um vídeo retratado de 1080x1920 e como 60.6 graus em modo paisagem de 1920x1080. São 31 graus de erro somente pelo formato da captura.

Somam-se a isso a dependência da distância à câmera (um aluno mais próximo produz segmentos maiores), a translação do usuário no quadro e a rotação no plano causada pelo celular apoiado torto.

4.4.3.2 A base do tronco

De cada quadro, extrai-se uma base a partir de quatro *landmarks* de pose, calculada conforme a Equação 2:

$$o = (P23 + P24)/2,\ e1 = P11 - P12,\ e2 = (P11 + P12)/2 - o \qquad (2)$$

Isto é, origem no ponto médio dos quadris, primeiro eixo na largura dos ombros e segundo eixo do centro dos quadris ao centro dos ombros. Cada ponto P passa a ser escrito como $P = o + a \cdot e1 + b \cdot e2$, e o par (a,b) substitui (x,y). Os coeficientes saem pela regra de Cramer, com o $det = e1x \cdot e2y - e2x \cdot e1y$.

4.4.3.3 Por que isso resolve o problema do formato?

A projeção da imagem se aplica em todos os pontos da mesma matriz diagonal $D = diag(1/W, 1/H)$. Sejam v um vetor físico qualquer e E1, E2 os eixos físicos do tronco. O que se mede na imagem é Dv, DE1 e DE2. Resolver conforme Equação 3:

$$Dv = a(DE1) + b(DE2) \quad \textbf{(3)}$$

É, linearmente o mesmo que resolver, $D(aE1 + bE2)$; e como D é inversível, isso equivale a $v = aE1 + bE2$.

Logo (a,b), são coordenadas do vetor físico na base física do tronco e por tanto não dependem nem de W e nem de H.

Entretanto, a base (e1, e2) não é ortonormalizada, $|\boldsymbol{e1}|$ é a largura dos ombros e $|\boldsymbol{e2}|$ é a altura do tronco, grandezas diferentes entre si e não necessariamente perpendiculares. Os coeficientes (a,b) são, como demonstrado anteriormente, os coeficientes físicos, mas o produto escalar calculado sobre os coeficientes não é um produto escalar físico, portanto os ângulos medidos no espaço normalizado não são ângulos físicos. Isso não invalida a técnica já que o método garante que a posição relativa do corpo seja comparável.

4.4.3.4 Tratamento da coordenada Z

A coordenada Z recebe tratamento distinto e mais fraco, calculada conforme Equação 4:

$$zcorpo = (pz - oz) / max(|e1x|, \varepsilon) \quad \textbf{(4)}$$

O MediaPipe entrega Z na escala de X, e por isso dividi-lo pela largura de ombros medida em X preserva a invariância de formato e de escala. Mas trata-se apenas de translação e escala, a profundidade é apenas normalizada.

Uma rotação do tronco fora do plano da imagem, não é compensada. A escolha é consciente, pois construir um terceiro eixo confiável exigiria um Z de qualidade que o MediaPipe monocular não consegue fornecer.

Existe um risco númerico. O divisor de Z tem piso de $\boldsymbol{10^{-3}}$ e não é protegido pelo teste de determinante da próxima sessão. Quando o usuário rotaciona o tronco para o perfil E1x tende a zero, mas o determinante $E1x \cdot E2y - E2x \cdot E1y$ pode continuar acima do limite de $\boldsymbol{10^{-6}}$ por causa da segunda parcela. O quadro passa no teste, ganha uma base válida (a,b) e um Z dividido pelo piso, portanto muito maior em várias ordens de grandeza. Como os ossos são vetores tridimensionais normalizados, um Z desses domina a direção de todos eles e estraga os ângulos do quadro. Este caso não está coberto por nenhuma proteção.

4.4.3.5 Quando a normalização é aplicada

Quando $|det| < 10^{-6}$, os dois eixos do tronco estão quase na mesma direção (Usuário de perfil com o tronco escondido) logo a conta ficaria instável. Nesse caso o quadro é descartado e seus pontos tornam-se "não detectados".

Quando a pose "pisca", as mãos costumam continuar detectadas. Para não perder o quadro inteiro, a última base válida é mantida por 500ms. O tronco normalmente se move mais devagar do que em relação às mãos, então uma base levemente atrasada é justificável nessa janela.

É preciso que as duas gravações tenham base válida em pelo menos metade dos quadros. Normalizar só um lado seria pior que não normalizar nenhum: as duas séries ficariam em espaços diferentes e qualquer ângulo entre elas perderia função.

4.4.3.6 Proporções humanas

Sobre a dependência das proporções do corpo, especificamente sobre a razão entre largura dos ombros e a altura do tronco, a definição do formato da base. Dois usuários com proporções diferentes geram bases de formatos diferentes e essa diferença não é cancelada.

O "resíduo" pode ser estimado com a mesma fórmula da seção 4.4.3.6, porque o mecanismo é o mesmo: uma diferença de proporção corporal atua como um descasamento de fator K entre as duas bases.

Quadro 4 – Demonstração de erro máximo

Diferença na razão Ombro/Tronco	k	Erro angular máximo	Nota resultante (k = 3)
10%	1.1	2,7	95,5%
15%	1.15	4,0	93,5%
25%	1.25	6,4	89,7%

Fonte: Autoria própria (2026)

Mesmo com uma diferença grande de porte físico, de 25%, o erro residual fica em 6,4° , o que é cerca de um quinto do que o formato de vídeo causava, e um custo de dez pontos na nota em vez de quarenta e quatro.

O valor da diferença de proporção não foi medido neste trabalho. O quadro mostra a sensibilidade ao parâmetro, justamente porque não há aqui levantamento antropométrico da população de usuários nem fonte que dê diretamente a variação dessa razão específica. A conclusão que o texto sustenta é a da forma da curva de crescimento na qual o resíduo cresce devagar com a diferença de porte e não um valor numérico particular.

4.4.4 Extração de features

4.4.4.1 Vetores de osso

Cada par (pai, filho) de pontos define um osso, e cada osso gera, por quadro, um vetor unitário, calculado conforme Equação 5:

$$\hat{b}k(t) = (pfilho(t) - ppai(t)) / \|pfilho(t) - ppai(t)\| \quad \textbf{(5)}$$

Comparar direções em vez de posições é a decisão central da representação. Posição absoluta depende de onde o usuário está no quadro e do tamanho dos membros dele. Já a direção não. Um aluno de braços mais curtos executa o mesmo sinal com as mesmas direções de segmento e posições finais diferentes. Comparar direções torna a métrica imune ao porte físico sem nenhum ajuste extra.

São 15 ossos de pose e 15 falanges por mão, num total de 45 séries de vetores.

4.4.4.2 Pontos ausentes

Quando um ponto não é encontrado, ou quando o osso tem comprimento zero, o sistema devolve um vetor (∞, ∞, ∞), e não um vetor nulo. Não utilizamos um vetor zerado, pois zero é um valor válido de produto escalar e passaria pela nota como se fosse medida.

4.4.4.3 Pesos por osso

Os pesos vão de 0,1 a 1,5 e refletem relevância:

Quadro 5 – Pesos por grupos de ossos

Faixa	Ossos	Motivo
1,2–1,5	Falanges	Determina a configuração da mão
1,3	Antebraço	Determina onde a mão está
0,6–1,0	Braço, dorso da mão	Informação parcialmente repetida pelo antebraço
0,1–0,3	Tronco, ombros, quadril	Postura, quase sem valor distintivo
0,1	Pernas	Não participam da articulação na Libras

Fonte: Autoria própria (2026)

As pernas mostram por que os pesos são necessários. O MediaPipe detecta pernas, e 6 dos 15 ossos de poses são de membro inferior. Sem pesos, esses 6 responderam por 40% da nota de pose. E são os

ossos mais fáceis de acertar, porque quase todo aluno fica de pé e parado, ou seja, seriam uma fonte de nota alta sem relação com a qualidade do sinal.

Há também um motivo para valorizar segmentos distais. O erro de um segmento proximal se propaga para toda a cadeia abaixo dele, mas o contrário não acontece. Um erro medido no antebraço já contém o efeito de um erro no braço.

4.4.5 O problema de alinhamento temporal:

Não podemos comparar quadro n da execução com o quadro n do gabarito. Isso supõe três coisas que não acontecem juntas:

- Mesma taxa de quadros. O gabarito é gravado a 30 fps; a captura no celular fica entre 12 e 25 fps, dependendo do aparelho e da luz. O mesmo gesto ocupa números diferentes de quadros.
- Início ao mesmo tempo. O usuário tem um tempo de reação antes de reagir ao comando, e a gravação costuma incluir o movimento de tirar a mão do botão.
- Mesma velocidade. Executar o sinal mais devagar não é erro de configuração de mão, mas a comparação por índice transforma isso em erro em todos os quadros.

Qualquer uma das três coisas vira erro durante toda a gravação, e esse erro tende a ser maior que o erro de execução que se quer medir.

4.4.5.1 Formulação:

O alinhamento usa *Dynamic Time Warping*, formulado por Sakoe e Chiba (1978) para reconhecimento de fala e adaptado para séries temporais gerais por Berndt e Clifford (1994). Para cada grupo, o custo entre o quadro i da execução e o quadro j do gabarito é a média ponderada da distância de corda sobre os ossos detectados nos dois lados, calculada conforme Equação 6:

$$c(i,j) = \Sigma k \in D(i,j)\ wk(1 - \hat{a}k(i)\cdot\hat{b}k(j)) \ /\ \Sigma k \in D(i,j)\ wk \qquad \textbf{(6)}$$

onde D(i,j) são os ossos com vetor válido nos dois quadros. A matriz acumulada segue a recursão padrão, calculada conforme Equação 7:

$$D(i,j) = c(i,j) + min\{D(i\text{-}1,j),\ D(i,j\text{-}1),\ D(i\text{-}1,j\text{-}1)\} \qquad \textbf{(7)}$$

E o caminho ótimo é recuperado voltando do canto (*na*-1, *nb*-1) até a origem.

4.4.6. Cálculo da nota

4.4.6.1 Ângulo médio ao longo do caminho

Para cada osso k, calculado conforme Equação 8:

$$\bar{\theta}k = \Sigma(i,j) \in P\ \theta k(i,j) \cdot \omega(i,j) \ /\ \Sigma(i,j) \in P\ \omega(i,j) \quad \textbf{(8)}$$

A ordem das operações não é neutra: primeiro se calcula a média dos ângulos e só depois se aplica a curva de similaridade. Como a curva $(1 - \theta/180)\hat{}\kappa$ é convexa para $\kappa > 1$, a desigualdade de Jensen garante que, conforme Equação 9:

$$S(\bar{\theta}) \leq S(\theta) \quad \textbf{(9)}$$

Ou seja, a ordem adotada é sempre a mais severa das duas. A diferença não é pequena. Um osso que fica metade dos quadros perfeito (0°) e metade a 90° recebe 42,2% pela ordem adotada e 56,3% pela outra. A escolha é intencional: uma execução que acertar metade do tempo e erra feio na outra metade não é meio sinal correto.

4.4.6.2 Pesos de repouso

Toda gravação começa e termina com o usuário em posição neutra. Esses quadros são parecidos entre quaisquer duas gravações, então incluí-los com peso cheio infla a nota e premia gravações mais longas.

O peso sobe de 0,1 a 1,0 ao longo do primeiro segundo e desce simetricamente no último. A rampa é contínua de propósito: um corte seco criaria uma descontinuidade em que um quadro a mais de repouso mudaria a nota de repente. Como o tempo de repouso é dado em segundos e convertido em quadros pela taxa de quadros de cada vídeo, a região desvalorizada é a mesma em tempo real dos dois lados. Para um par alinhado, o peso é o menor dos dois, calculado conforme Equação 10:

$$\omega(i,j) = min(wa(i),\ wb(j)) \quad \textbf{(10)}$$

4.4.6.3 Curva de similaridade

O ângulo médio vira nota, calculado conforme Equação 11:

$$S(\bar{\theta}) = 100 \cdot max(0, (1 - \bar{\theta}/180)^{\wedge}\kappa) \quad \textbf{(11)}$$

Uma conversão linear (κ=1) seria complacente demais: 45° de erro médio dariam 75%, e 45° de erro em todas as falanges descrevem uma mão visivelmente errada. O expoente aperta a faixa intermediária, onde está a maior parte dos erros que interessam, sem tornar a métrica binária.

São usados os valores de k:

Quadro 6 – Valor de κ aplicado a grupos de ossos

κ	Aplicado a	Motivo
2,0	Ossos de pose	Variação de postura corporal dentro de certa margem é aceitável entre sinalizantes
3,0	Falanges e normal da palma	A mesma margem na mão altera o sinal
3,5	Direção global da mão	É a medida menos ruidosa, então um erro observado ali provavelmente é erro real, o que permite uma curva mais severa

Fonte: Autoria própria (2026)

4.4.6.4 Nota do grupo:

As notas para os grupos de mão são calculados conforme a Equação 12:

$$Sgrupo = [\Sigma k \; S(\theta k) \cdot wk + Spalma \cdot 3{,}5 + Sdir \cdot 4{,}0 + Sfase \cdot 1{,}5] \; / \; [\Sigma k \; wk + 3{,}5 + 4{,}0 + 1{,}5] \quad \textbf{(12)}$$

A fórmula vale para os grupos de mão. O grupo de pose não tem normal de palma nem direção global, e sua nota é só ossos mais o termo de fase.

Os termos de palma, direção e fase entram no mesmo somatório dos ossos, e não como multiplicadores ou bônus separados. Assim a nota continua sendo uma média ponderada legítima, limitada a 100 e lida como "porcentagem de acerto ponderada pela relevância de cada aspecto". Termos indisponíveis saem do numerador e do denominador, o que preserva a escala.

Palma e direção são medidas ao longo do mesmo caminho de alinhamento usado pelos ossos daquela mão. Medir a palma num casamento temporal diferente dos dedos descreveria, de novo, uma mão que não existe.

Uma dependência entre os dois termos. Na implementação atual, o cálculo da direção só acontece se a similaridade de palma for válida. Se faltarem os pontos 5 ou 17, inviabilizando a normal, mas o par pulso/médio existir, a direção é descartada junto. As duas medidas usam pontos da mesma mão e quase sempre falham juntas, então o caso é raro, mas a regra implementada é mais restritiva que a descrita acima.

4.4.6.5 Quantidade de movimento:

Este termo mede se o aluno se moveu tanto quanto o sinal exige. Usa-se o comprimento total da trajetória, somando os deslocamentos médios dos pontos entre quadros seguidos, calculado conforme Equação 13:

$$L = \Sigma t=1..n-1 \; \| p(t) - p(t-1) \| , \quad Sfase = 100 \cdot min(La, Lb) \; / \; max(La, Lb) \qquad \textbf{(13)}$$

A razão entre percursos pune tanto mover de menos quanto mover de mais, e chega a 100% quando são iguais.

Um desenho anterior do validador separava as duas séries em regiões de atividade, casava o segmento i da execução com o segmento i do gabarito e rodava um DTW de uma dimensão sobre a componente x de um único osso escolhido como referência. Era frágil por dois motivos. Uma piscada de detecção criava segmentos falsos e deslocava todos os pares seguintes. E o movimento de abaixar o braço depois de tocar na tela era detectado como atividade, virava "segmento 1" e era comparado com o início real do sinal.

4.4.7.1 Tratamento de dados ausentes

O gabarito define o quais grupos o sinal exige. Se um grupo aparecer em pelo menos 20% dos quadros do gabarito, o sinal o exige. Um sinal de uma mão não exige a outra, e a mão ausente não entra na média.

Se o grupo é exigido e a nota do aluno é “NAN” por falta de pares válidos, a nota vira 0. O comportamento anterior era tirá-lo da média, com uma consequência ruim: um aluno parado, com as

mãos fora do quadro, tinha a nota calculada só sobre a pose. E como a pose em repouso do aluno é parecida com a do gabarito, ele herdava uma nota alta por não ter feito nada.

O grupo também recebe a marca missing_in_user quando a cobertura de detecção é exatamente zero. A marca é dada pela detecção, e não pela nota, porque o termo de fase pode já ter zerado o grupo por outro motivo. "Nunca apareceu na execução" é um diagnóstico diferente.

4.4.7.2 Piso de cobertura

Detecção esparsa também é um problema. Uma mão vista em três quadros de uma gravação de trezentos pode gerar três pares válidos, todos com ângulo pequeno, e pontuar quase 100% com peso cheio.

Portanto, define-se então uma cobertura mínima como fração da cobertura do gabarito, e um fator contínuo, calculado conforme Equação 14:

$$req = cob_b \cdot 0{,}3, \quad f = clamp(cob_a \;/\; req,\ 0,\ 1), \quad Sgrupo \leftarrow Sgrupo \cdot f \quad \textbf{(14)}$$

O aluno precisa alcançar pelo menos 30% da cobertura do gabarito para o grupo valer nota cheia. Abaixo disso, a nota é reduzida proporcionalmente. O limiar é bem permissivo de propósito, porque a cobertura do aluno é legitimamente menor: iluminação não ideal, enquadramento pior, mão saindo do quadro, oclusão.

Um limiar rígido traria de volta a descontinuidade que se quer eliminar, com um quadro a mais mudando a nota de zero para cheia. Com escala linear, a nota cai suavemente conforme a evidência diminui.

4.4.7.3 Piso de cobertura:

O piso de cobertura é calculado conforme Equação 15:

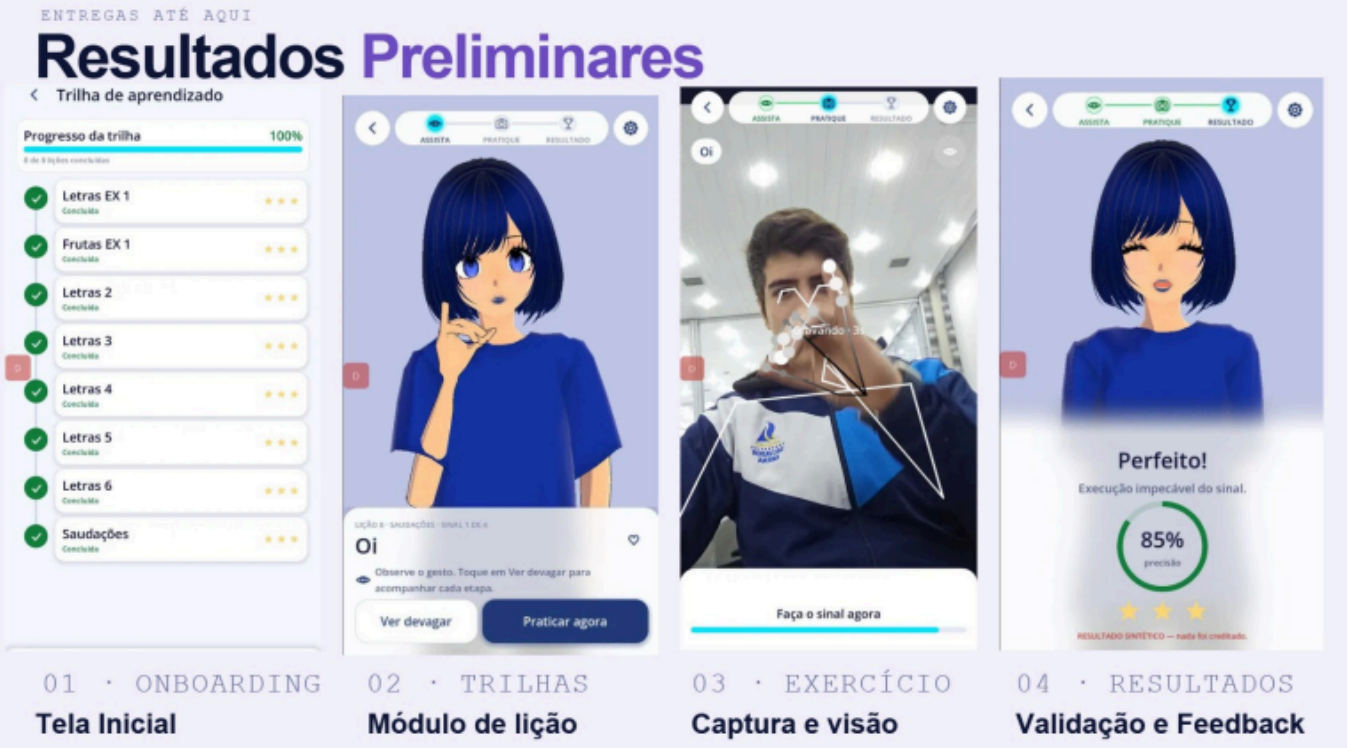
$$Sglobal = \Sigma g\ Sg{\cdot}Wg \;/\; \Sigma g\ Wg, \quad W = \{pose:\ 0{,}3;\ mão\ esq.:\ 1{,}0;\ mão\ dir.:\ 1{,}0\} \quad \textbf{(15)}$$

A pose vale menos de um sexto do total. Ela não é descartada, porque cobre em parte o ponto de articulação através da posição dos braços, mas o sinal, na Libras, em grande parte está nas mãos.

5 Resultados preliminares

O protótipo desenvolvido já integra as três camadas da arquitetura proposta, interface em Godot Engine, API hospedada na Oracle Cloud e a base de dados MySQL alimentada por sinais extraídos do VLibras via MediaPipe. Dessa forma, é possível testar o fluxo completo do sistema, desde a seleção de uma lição até a exibição da nota de similaridade calculada pelo validador geométrico. A Figura 3 ilustra esse fluxo por meio de capturas de tela da aplicação, desde a tela inicial de onboarding até a tela de resultados com o feedback do desempenho do usuário

Figura 3 – Compilado com capturas das telas no estado atual da plataforma



Fonte: Autoria própria (2026)

Nesse estágio, foi possível verificar que a captura de gestos pela câmera, o processamento geométrico (suavização, normalização pelo tronco, extração de vetores de osso e alinhamento temporal por DTW) e a geração de uma nota de desempenho ocorrem de forma integrada e em tempo real, para um subconjunto de sinais utilizado na depuração.

Além da demonstração funcional, foram conduzidos testes controlados para verificar empiricamente o comportamento dos mecanismos, em especial o alinhamento temporal (DTW), a suavização, a normalização e os limiares de cobertura.

5.1 Validação geométrica: estudo de ablação

Os cenários foram construídos a partir do gabarito real do sinal "Abacaxi" e de dados sintéticos gerados por fórmula, de modo a isolar o efeito de cada componente do pipeline. O Quadro 7 descreve como cada cenário foi construído e o resultado obtido.

Tabela 1 – Cenários de validação do validador geométrico

Cenário	Construção	Similaridade
Mesmo sinal repetido	Gravação comparada com ela mesma	99,99%
Atraso de 0,5 s, sem DTW → com DTW	Gabarito real comparado com ele mesmo, com 6 quadros de repouso adicionados no início	83,14% → 99,98% (+16,84 pontos)
Taxa de captura diferente, sem DTW → com DTW	Mesmo gabarito, descartando 1 quadro a cada 2	69,38% → 93,89% (+24,51 pontos)
Execução canhota (espelhada)	Gabarito comparado com sua versão espelhada horizontalmente	99,99%, reconhecida
Usuário parado (falso positivo)	Dados sintéticos: pontos gerados por fórmula, sem nenhuma mão detectada	21,77% (0 estrela)
Metade do percurso do movimento	Dados sintéticos	48,3%
Captura a 15 fps contra gabarito a 30 fps	Dados sintéticos	94,9%

Fonte: Autoria própria (2026)

O alinhamento temporal por DTW recupera entre 16,8 e 24,5 pontos percentuais de similaridade em cenários de dessincronia temporal e de taxas de quadros distintas, que de outra forma penalizavam incorretamente uma execução correta apenas por diferenças de tempo ou de fps. O mecanismo de penalização por imobilidade atribui corretamente uma nota baixa (21,77%, equivalente a 0 estrela) a um usuário que não executa o sinal, evitando o falso positivo descrito na Seção 4.4.7.1. O termo de quantidade de movimento também se mostra sensível à amplitude do gesto, atribuindo 48,3% a uma execução com metade do percurso esperado. Por fim, a invariância da métrica ao membro dominante (execução canhota reconhecida com 99,99%) confirma que a comparação por direção de osso, e não por posição absoluta, generaliza corretamente para sinalizantes canhotos.

5.2 Desempenho computacional

Os testes de desempenho foram executados em ambiente desktop, utilizando um processador Intel Core 5 210H de 12 threads. O Quadro 8 apresenta o tempo médio de inferência do MediaPipe por quadro,

calculado a partir de 40 execuções. Como o quadro utilizado no benchmark estava vazio, apenas o detector de corpo foi executado nesta medição, sem o custo adicional da detecção de mãos.

Tabela 2 – Tempo médio de inferência do MediaPipe por quadro (corpo, quadro vazio)

Configuração	Tempo
CPU sem aceleração (1 thread)	18,4 ms
2 threads	9,1 ms
4 threads	5,8 ms (3,2× mais rápido)
8 threads	8,0 ms

Fonte: Autoria própria (2026)

O ganho de desempenho não é monotônico com o número de threads: o melhor resultado ocorre em 4 threads, com piora em 8, provavelmente por sobrecarga de sincronização. O carregamento do modelo do MediaPipe leva, em média, 200 ms.

O Quadro 9 apresenta o custo total da validação geométrica em função da duração da gravação, decompondo o tempo entre as etapas de filtragem e normalização, alinhamento temporal (DTW) e cálculo final da nota.

Tabela 3 – Custo da validação completa por duração da gravação

Duração	Pipeline completo	Sem DTW	Sem filtro e normalização
2,5 s	151 ms	110 ms	18 ms
5 s	409 ms	167 ms	26 ms
10 s	737 ms	271 ms	40 ms

Fonte: Autoria própria (2026)

O crescimento do tempo de execução com a duração da gravação é predominantemente atribuível ao DTW, cujo custo cresce de forma quadrática com o número de quadros. Como a duração média dos gabaritos utilizados é de aproximadamente 2 segundos, o custo típico de uma validação em produção fica em torno de 150 ms, valor compatível com o uso em tempo real dentro do fluxo de prática da aplicação.

Em relação à API, o carregamento de uma lição leva em média 351 ms (mediana de 323 ms, variando entre 216 e 679 ms), enquanto o carregamento do catálogo de lições leva, em média, 307 ms.

5.3 Características técnicas do protótipo

O Quadro 10 resume as principais características técnicas do protótipo na versão atual.

Quadro 7 – Características técnicas do protótipo

Variável	Valor
Modelo de visão (holistic_landmarker.task)	15,0 MB
Efeitos sonoros	11
Plataforma-alvo	Android arm64, tela em retrato 1080×1920, Godot 4.7
Recompensas	20 XP por estrela, 500 XP por nível

Fonte: Autoria própria (2026)

Os resultados desta seção são preliminares quanto à eficácia pedagógica do sistema: eles demonstram que os mecanismos geométricos se comportam como projetado e que o desempenho computacional é compatível com uso em tempo real, porém não substituem a avaliação com usuários reais.

6 Considerações finais

O uso da tecnologia na educação pode ampliar as possibilidades de estudo e prática, oferecendo aos usuários novas formas de interação com os conteúdos aprendidos. No contexto da Língua Brasileira de Sinais (Libras), ferramentas digitais podem atuar como recursos complementares, permitindo que os estudantes retomam sinais previamente estudados e pratiquem sua execução de maneira interativa. Entretanto, o desenvolvimento de soluções dessa natureza também apresenta desafios, especialmente, na avaliação dos movimentos realizados e na representação adequada dos componentes dos sinais.

Diante desse cenário, o HandSign foi desenvolvido com a proposta de oferecer um ambiente gamificado para a prática e o reforço de sinais de Libras. A aplicação integra exercícios, validação dos movimentos e elementos de progressão, fornecendo retorno ao usuário durante as atividades. Ressalta-se, contudo, que o HandSign não tem como objetivo substituir professores, cursos ou a interação com pessoas proficientes em Libras, mas complementar as experiências de aprendizagem já existentes.

Os resultados ainda são preliminares, uma vez que parte das avaliações utilizou dados sintéticos e um conjunto limitado de referências reais. Dessa forma, eles não permitem afirmar que o sistema reconhece adequadamente diferentes sinais em todas as condições de uso, nem comprovar que a aplicação melhora a aprendizagem, a motivação ou a retenção dos conteúdos.

Assim, considera-se que o desenvolvimento deste trabalho é uma etapa inicial para a integração de prática de sinais, visão computacional e gamificação em um recurso digital complementar. A continuidade do projeto permitirá investigar de forma mais abrangente a confiabilidade da avaliação e da

experiência dos usuários, contribuindo para que a ferramenta evolua de acordo com as necessidades identificadas durante sua utilização.

REFERÊNCIAS

BERNDT, D. J.; CLIFFORD, J. **Using dynamic time warping to find patterns in time series.** *In*: AAAI WORKSHOP ON KNOWLEDGE DISCOVERY IN DATABASES, 1994, Seattle. **Anais** [...]. Menlo Park: AAAI Press, 1994. p. 359-370.

BRASIL. Governo Digital. **VLibras**. Disponível em: https://www.gov.br/governodigital/pt-br/acessibilidade-e-usuario/vlibras. Acesso em: 07 abr. 2026.

BRASIL. Lei nº 10.436, de 24 de abril de 2002. Dispõe sobre a Língua Brasileira de Sinais - Libras e dá outras providências. Brasília, DF: Diário Oficial da União, 2002. Disponível em: https://www.planalto.gov.br/ccivil_03/leis/2002/l10436.htm. Acesso em: 29 abr. 2026.

BRASIL. Decreto nº 5.626, de 22 de dezembro de 2005. Regulamenta a Lei nº 10.436, de 24 de abril de 2002, que dispõe sobre a Língua Brasileira de Sinais - Libras, e o art. 18 da Lei nº 10.098, de 19 de dezembro de 2000. Brasília, DF: Diário Oficial da União, 2005. Disponível em: https://www.planalto.gov.br/ccivil_03/_ato2004-2006/2005/decreto/d5626.htm. Acesso em: 29 abr. 2026.

CASIEZ, Géry; ROUSSEL, Nicolas; VOGEL, Daniel. **1 € Filter: a simple speed-based low-pass filter for noisy input in interactive systems**. In: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '12). New York: Association for Computing Machinery, 2012. p. 2527-2530. DOI: 10.1145/2207676.2208639.

FADEL, Luciane Maria; ULBRICHT, Vania Ribas; BATISTA, Claudia Regina; VANZIN, Tarcísio (org.). **Gamificação na educação**. São Paulo: Pimenta Cultural, 2014. 300 p.

FREEMAN, Scott et al. Active learning increases student performance in science, engineering, and mathematics. Proceedings of the National Academy of Sciences, Washington, v. 111, n. 23, p. 8410–8415, 2014. DOI: 10.1073/pnas.1319030111.

GALEOTE, Sidney (orient.). **Libras4All: sistema com gamificação para apoio na aprendizagem de libras**. Ítalo Martins Tomaz ROCHA, Milena Ferreira de PAULA, Patricia Amanda de CASTRO.

Disponível em: https://biblioteca.sophia.com.br/terminal/9972/acervo/detalhe/40217. Acesso em: 15 abr. 2026.

GODOT ENGINE. **Godot Engine Documentation**. Disponível em: https://docs.godotengine.org/. Acesso em: 24 mar. 2026.

GRACILENE DE SOUZA, Francisca; SANTOS, Juliana Iraci Gomes da Rocha. UMA REVISÃO DA LITERATURA PARA A INCLUSÃO DOS SURDOS: O ENSINO DE LIBRAS E A INCLUSÃO ESCOLAR SOB O OLHAR DA PRODUÇÃO ACADÊMICA. **Revista Ibero-Americana de Humanidades, Ciências e Educação**, [S. l.], v. 9, n. 10, p. 6181–6192, 2023. DOI: 10.51891/rease.v9i10.12131. Disponível em: https://periodicorease.pro.br/rease/article/view/12131. Acesso em: 1 ago. 2026.

INSTITUTO BRASILEIRO DE GEOGRAFIA E ESTATÍSTICA (IBGE). **Censo Demográfico 2010/2022.** Disponível em: https://www.ibge.gov.br. Acesso em: 16 fev. 2026.

INSTITUTO BRASILEIRO DE GEOGRAFIA E ESTATÍSTICA (IBGE**). Pesquisa Nacional de Saúde 2019:** ciclos de vida: Brasil. Rio de Janeiro: IBGE, 2021. Disponível em: https://biblioteca.ibge.gov.br/visualizacao/livros/liv101846.pdf. Acesso em: 15 out. 2026.

LUGARESI, Camillo et al. **MediaPipe:** a framework for building perception pipelines. arXiv, 2019. Disponível em: https://arxiv.org/abs/1906.08172. Acesso em: 15 out. 2026.

MARTIN, Robert C. Clean Code: **A Handbook of Agile Software Craftsmanship.** Prentice Hall, 2008.

MEAD, George Herbert. **Mind, Self and Society:** From the Standpoint of a Social Behaviorist. Edited by Charles W. Morris. Chicago: University of Chicago Press, 1934.

NORIEGA, Carlos Enrique López (orient.). **Luva para reconhecimento da Língua brasileira de sinais (Libras) mediante sensoriamento e inteligência artificial através de um aplicativo móvel**. Rafael Fantinati Pignatari, Larissa Dias de Souza, Arthur Silva Maloyama. Disponível em: https://biblioteca.sophia.com.br/terminal/9972/acervo/detalhe/40415. Acesso em: 15 abr. 2026.

ORACLE. **Oracle Cloud Infrastructure Documentation**. Disponível em: https://docs.oracle.com/enus/iaas/. Acesso em: 09 abr. 2026.

PESSÔA, Everton. QUADROS, Ronice Müller de. Libras. São Paulo: Parábola, 2020. 192f. ISBN 97885-7934-166-3. Cadernos de Linguagem e Sociedade, [S. l.], v. 22, n. 2, p. 496–499, 2021. DOI: 10.26512/les.v22i2.40872. Disponível em: https://periodicos.unb.br/index.php/les/article/view/40872. Acesso em: 3 maio. 2026.

ROCHA, P. S. R. **Uma ferramenta computacional gamificada como estratégia de apoio para aprendizagem da língua brasileira de sinais (Libras).** Dissertação (Mestrado) - UFERSA, 2018.

QUADROS, Ronice Müller de; KARNOPP, Lodenir Becker. **Língua de sinais brasileira:** estudos linguísticos. Porto Alegre: Artmed, 2004.

SAKOE, H.; CHIBA, S. Dynamic programming algorithm optimization for spoken word recognition. **IEEE Transactions on Acoustics, Speech, and Signal Processing**, v. 26, n. 1, p. 43-49, 1978.

UCHÔA, A. K. R. **Gamificação no ensino de Libras:** explorando ferramentas digitais e estudos de caso. TCC (Graduação) - Universidade Federal do Amazonas, 2024.

Werbach, K., & Hunter, D. **For the Win:** How game thinking can revolutionize your business. Wharton Digital Press, 2012.