

LIBRAR: AUXÍLIO PARA TREINAMENTO DE LIBRAS

Matheus Adriano Pereira, Dalton Solano dos Reis – Orientador

Curso de Bacharel em Ciência da Computação
Departamento de Sistemas e Computação
Universidade Regional de Blumenau (FURB) – Blumenau, SC – Brasil

matheusadriano@furb.br, dalton@furb.br

Resumo: Este artigo apresenta o processo de desenvolvimento e teste de um software que tem como objetivo auxiliar no aprendizado de Libras. O software disponibiliza módulo de definições de sinais, que pode ser utilizado por profissionais da área para cadastrar quaisquer sinais de Libras e definir como eles devem ser comparados. Também disponibiliza um módulo de treinamento, que utiliza das configurações definidas pelo profissional para validar o sinal de Libras reproduzido pelo usuário, orientando-o até que conclua o sinal de forma correta. O software foi desenvolvido no Unity utilizando C#. Para validar o software foi realizado reuniões com profissionais da área de Libras para orientar os detalhes na realização dos sinais de Libras. O objetivo foi atingido possibilitando um usuário que não conhece Libras reproduzir o sinal, seja estático ou dinâmico.

Palavras-chave: Libras. Leap Motion. Aprendizado de Libras. Unity. C#.

1 INTRODUÇÃO-

A comunicação permite-se transmitir ideias, mensagens, sentimentos e emoções, podendo influenciar pessoas que, por sua vez, reagirá de acordo com suas crenças, valores, história de vida e cultura (SILVA et al., 2000). A comunicação sendo algo tão importante no dia-a-dia, nem todos se comunicam da mesma forma. Como pode ser observado no censo de 2009 (IBGE, 2012) no Brasil existem cerca de 9,7 milhões de pessoas com algum tipo de deficiência auditiva, sendo cerca de 345 mil dessas são surdas, e sua forma de comunicação é a partir da Língua BRAsileira de Sinais (Libras).

Libras se constitui em uma língua que possibilita comunicação aos seus usuários de todas as formas de ideias, sejam elas sutis, complexas ou abstratas, bem como a discussão de assuntos de variadas áreas pessoais e interpessoais (CARVALHO; TAVARES, 2010). Porém nem sempre o usuário consegue se comunicar. Segundo Lacerda (2006, p. 117), o aluno surdo é como um estrangeiro que tem acesso a conhecimentos de modo diverso dos demais e se mantém isolado. “A questão da língua é fundamental, pois, sem ela, as relações mais aprofundadas são impossíveis, não se pode falar de sentimentos, de emoções, de dúvidas, de pontos de vista diversos [...]” (LACERDA, 2006, p. 117).

Para diminuir este isolamento deve acontecer mudanças. Luz (2013) propõe essas questões: “quem ou o que deveria mudar? Onde exatamente estaria localizada a deficiência? Nos filhos? Nos pais? Nos surdos que preferem estar com outros surdos? Na sociedade? No poder público? Na legislação? Em todos?” (LUZ, 2013, p.200). Para uma mudança na sociedade é necessário interesse no aprendizado de Libras por parte dela. Esse interesse poderia ser despertado por meios digitais já na fase escolar. Pois segundo pesquisa realizada na universidade John Hopkins (LEHMANN et al., 1999), verificou-se que o uso de meios digitais aumentou a satisfação do aluno com o laboratório, também aumentando a predisposição para o estudo da disciplina.

Uma mídia digital que recentemente vem sendo explorada para aprendizado é a Realidade virtual. O LibRAR (SILVA, 2018) utiliza realidade virtual para o auxílio no ensino de Libras para aumentar a interatividade (no módulo Aprender os Sinais). O LibRAR também possui os módulos Jogo Associativo e Jogo de Raciocínio Rápido, que apresentam os conceitos básicos e treinos utilizando realidade aumentada. Outro exemplo de mídia digital que pode ser usado com alunos é leitores de movimentos. Treinamentos auxiliados por dispositivos de realidade virtual e reconhecimento de sinais possibilitam realismo e percepção de imersão (XU, 2006). Uma das formas de aumentar a percepção da imersão pode ser obtida utilizando o equipamento Leap Motion. Pois segundo a empresa Ultraleap Ltd (2019a, p. 1 tradução nossa), “Suas mãos são a interface universal original. Com o nosso acompanhamento de mãos sem precedentes, você pode alcançar a realidade virtual e aumentada para interagir com novos mundos”.

Diante do exposto se desenvolveu uma extensão do software LibRAR (SILVA, 2018), que tinha como objetivo auxiliar no ensino de Libras. Integrando com o hardware Leap Motion para reconhecimento dos sinais de Libras. Esta extensão permitirá aos usuários treinarem de forma prática os conceitos aprendidos, utilizando o hardware Leap Motion.

Este trabalho tem como objetivos auxiliar no aprendizado de Libras, possibilitando o profissional de Libras cadastrar sinais que serão usados para treino do aluno. Os objetivos específicos são: disponibilizar ao usuário animações que explicam os conceitos básicos da Libras; disponibilizar uma ferramenta ao profissional de Libras que permitirá cadastrar e definir os critérios de aceitação do sinal; permitir que os usuários treinem suas configurações de mão na reprodução do sinal de Libras de acordo com os critérios estabelecidos pelo profissional de Libras.

2 FUNDAMENTAÇÃO TEÓRICA

Esta seção descreve brevemente os assuntos que fundamentaram o estudo realizado neste projeto. Na subseção 2.1 é comentado sobre Libras e como ela é formada. Na subseção 2.2 é apresentado o Leap Motion explicando o seu uso e descrevendo suas configurações. Na subseção 2.3 é apresentado a versão anterior do software, demonstrando as funcionalidades já presentes. Na subseção 2.4, são apresentados três trabalhos correlatos ao projeto desenvolvido.

2.1 LIBRAS

Libras é um sistema linguístico de modalidade visual-espacial, baseado na Língua de Sinais Francesa, de forma independente possui uma gramática própria, porém sofre influência direta da língua portuguesa brasileira por meio de adaptações, isso ocorre por serem línguas em contato (QUADROS; KARNOPP, 2004). Assim, como possui uma gramática própria também possui estruturas sintáticas, semânticas e morfológicas próprias, exigindo um processo de aprendizagem (ARAÚJO, 2009). Em alguns casos onde não há um sinal correspondente, como no caso dos nomes, são sinalizadas as letras separadamente (ARAÚJO, 2009).

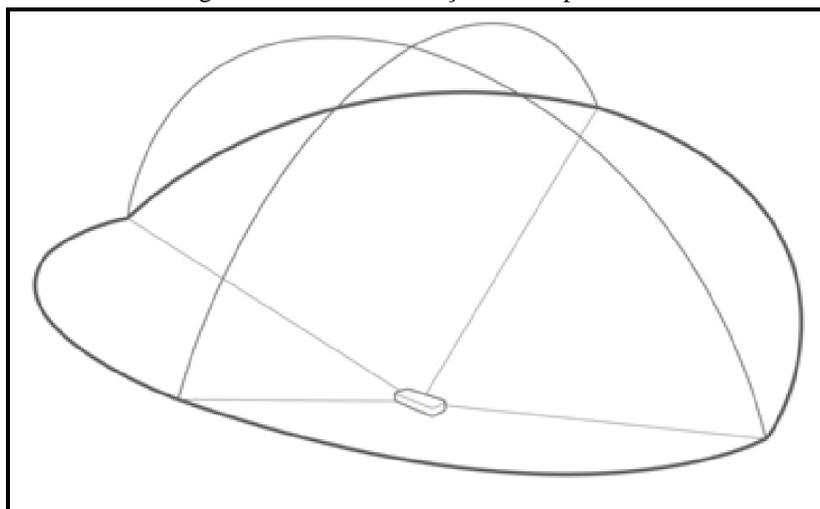
Um sinal pode ser articulado com uma mão ou duas mãos, podendo ser estático (fixos) ou dinâmico (com movimentos). Um mesmo sinal não se distingue se for reproduzido com a mão direita ou a mão esquerda, mantendo-se o significado. Isto é possível pois os sinais são formados por parâmetros fonológicos de configuração de mãos, ponto de articulação, movimento, orientação e expressão não-manual. A expressão facial/corporal ou não-manual é de extrema importância por definir a entonação na Língua de Sinais (BENTO, 2010).

A configuração de mãos, formas das mãos, é um articulador primário, sendo o parâmetro mais primitivo, pois não existe sinal sem configuração de mão, que também é utilizada para datilologia (alfabeto manual) (BENTO, 2010). O ponto de articulação acontece no espaço diante do emissor, podendo estar desde a linha da cintura até o alto da cabeça (FERREIRA et al., 2011). O movimento acontece no espaço com o deslocamento da configuração de mãos (BRITO, 1995). A orientação é a direção na qual a palma da mão aponta durante o sinal, podendo ser para cima, para baixo, para dentro, para fora ou para os lados (QUADROS; KARNOPP, 2004).

2.2 LEAP MOTION

O Leap Motion é um dispositivo desenvolvido pela empresa Ultraleap Ltd. A origem da empresa se dá pela fusão das empresas Ultrahaptics e Leap Motion em maio de 2019 (ULTRALEAP LTD, 2019a, p. 1). Esse dispositivo promete precisão milimétrica para detecção das mãos (WEICHERT et al., 2013). O dispositivo consiste em três diodo emissor de Luz (Light-Emitting Diode - LED) de infravermelho para iluminar as mãos do usuário. Também possui dois sensores de infravermelho para mapear os pontos dentro do campo de visão do dispositivo. Este campo de visão é de aproximadamente 2,5 metros cúbicos, podendo capturar pontos até aproximadamente 60 centímetros do dispositivo em ângulos de, 150° para os lados e 120° de profundidade, conforme pode ser visto na Figura 1 (BATISTA, 2015).

Figura 1 – Área de interação do Leap Motion

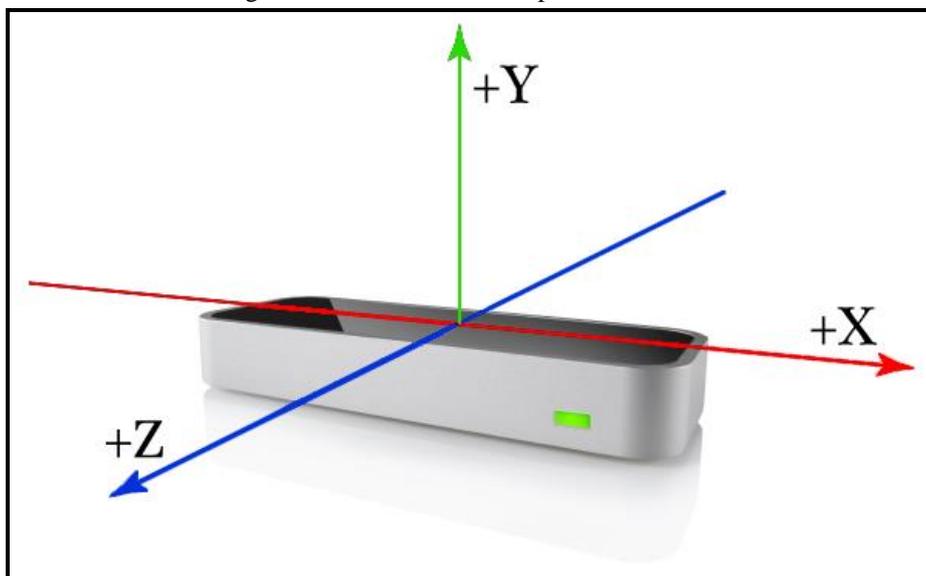


Fonte: Batista (2015, p.22).

Segundo Melo (2015, p. 41), “Cada quadro possui uma lista de dados referentes aos objetos mapeados pelo dispositivo, bem como os gestos reconhecidos por ele”. Os objetos capturados são mantidos em memória e para realização de ajustes na resolução e renderização. Posteriormente são transferidos por meio do Universal Serial Bus (USB) do dispositivo para o computador. Os dados podem ser gerenciados via uma biblioteca nativa disponibilizada pela empresa Ultraleap Ltd. ou via WebSocket. A biblioteca nativa pode ser incorporada na construção do software que

se pretende integrar o Leap Motion. A biblioteca WebSocket desenvolvida em JavaScript, permite um software web gerenciar e receber os dados do Leap Motion (ULTRALEAP LTD, 2019c).

Figura 2 – LibRAR módulo Aprender os Sinais



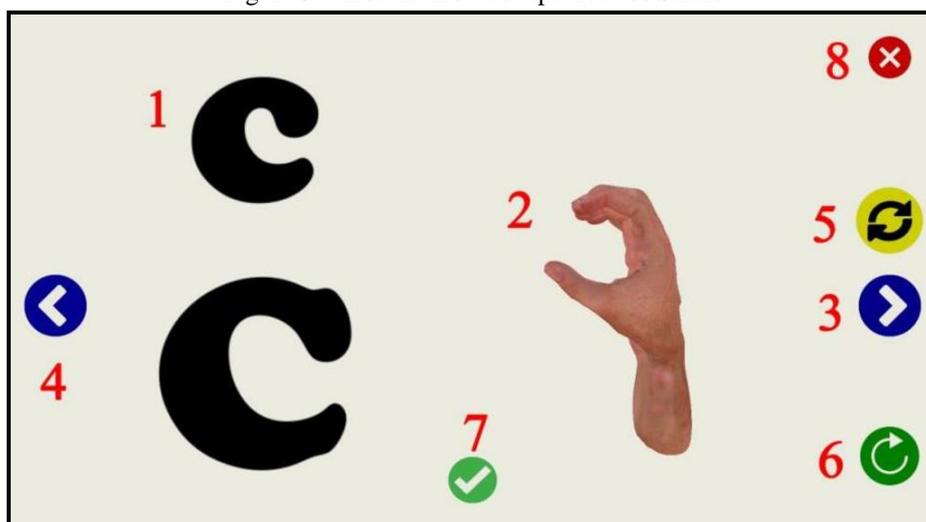
Fonte: Ultraleap Ltd (2019b).

O sistema de referência do Leap Motion (Figura 2) é o plano cartesiano para definir as coordenadas das mãos. Os valores reportados pelo dispositivo são medidos com origem do ponto focal no centro do dispositivo. O eixo X e eixo Z formam um plano horizontal, onde o eixo X é paralelo ao lado mais extenso do dispositivo. Além disso o eixo Y positivo é a vertical do dispositivo (ULTRALEAP LTD, 2019b, tradução nossa).

2.3 VERSÃO ANTERIOR DO SOFTWARE

O LibRAR (SILVA, 2018) possui três módulos, que tem como objetivo proporcionar aprendizado de Libras, utilizando demonstrações TriDimensionais (3D) e jogos para a prática da língua. O primeiro módulo (Figura 3), Aprender os Sinais, se dedica a uma apresentação 3D com animação, para cada letra e algarismo, possuindo uma visualização do sinal que a representa.

Figura 3 – LibRAR módulo Aprender os Sinais

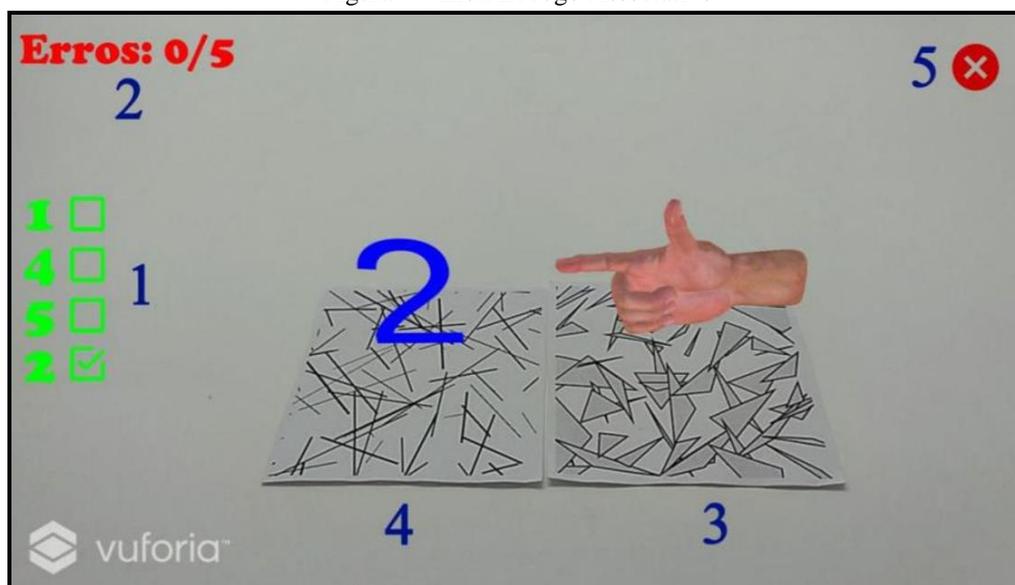


Fonte: Silva (2018).

Na Figura 3 (item 1) é exibido o significado do sinal atual. O item 2 mostra a mão 3D realizando a animação do sinal na Libras. Os itens 3 e 4 são botões para o usuário visualizar o próximo/anterior sinal na Libras, respeitando a ordem alfabética. O item 5 é um botão com o objetivo de reexecutar a animação da mão 3D do sinal atual, caso seja um sinal com animação. O item 6 é um botão para o usuário voltar a tela para escolher um novo conteúdo para o módulo. O item 7 indica quando a animação do sinal foi finalizada e o item 8 é um botão para o usuário voltar ao menu inicial do software (SILVA, 2018).

Os outros módulos, Jogo Associativo e Jogo de Raciocínio Rápido, são dedicados ao treinamento dos conhecimentos de Libras em formatos de jogos. O Jogo Associativo utiliza a biblioteca de Realidade Aumentada Vuforia, para que o usuário aproxime o sinal ao algarismo ou letra apresentados. O Jogo de Raciocínio Rápido utiliza a biblioteca Realidade Virtual Google VR para criar um ambiente virtual. Neste jogo, o usuário deve escolher o sinal que representa o algarismo ou letra apresentado antes que o tempo acabe. Para integração das bibliotecas presentes nos Jogos e nas animações do módulo Aprender os Sinais foi utilizado o Unity como ferramenta de desenvolvimento.

Figura 4 – LibrAR Jogo Associativo



Fonte: Silva (2018).

Na Figura 4 (item 1) é apresentado o Jogo Associativo que exibe ao usuário todos os sinais que ele precisa associar para ganhar o jogo. Neste mesmo item é mostrado também quais já foram associados de forma correta, por exemplo, o número dois é o único sinal que já foi associado, por isto está marcado com um sinal de correto ao lado dele. O item 2 exibe ao usuário a quantidade de erros cometidos e o total de erros que podem acontecer durante o jogo. O item 3 é o marcador responsável por exibir o sinal de Libras. O item 4 é o marcador responsável por exibir o significado do sinal. Por fim, o item 5 é um botão para o usuário voltar a tela inicial do software.

2.4 TRABALHOS CORRELATOS

Nesta seção serão apresentados três trabalhos correlatos que possuem características relacionadas ao software desenvolvido. O primeiro é exibido no Quadro 1 e trata do Protótipo de um tradutor de língua portuguesa para Libras (RODRIGUES, 2015). O segundo, Libras translator via web for mobile devices (LIMA et al, 2012), é apresentado no Quadro 2 e consiste em auxiliar na comunicação por texto digitado ou por sinais. Signa (BATISTA, 2015) é o terceiro, que consiste em um software para cadastrar e reconhecer sinais de Libras, é explanado no Quadro 3.

Quadro 1– Protótipo de um tradutor de língua portuguesa para Libras

Referência	Rodrigues (2015).
Objetivos	Realizar a tradução de sentenças escritas em língua portuguesa para Libras.
Principais funcionalidades	Apresenta a transcrição da frase textual utilizando a gramática de Libras, apresenta os vídeos dos sinais em Libras da frase na gramática de Libras.
Ferramentas de desenvolvimento	Foi desenvolvido em Java para o ambiente desktop com as bibliotecas Cogroo e Xuggle, que permitiram o reconhecimento da sentença.
Resultados e conclusões	Obteve resultados satisfatórios, porém é recomendado sua utilização como apoio de um profissional da área, para determinar se a apresentação da tradução está apropriada.

Fonte: adaptado de Rodrigues (2015).

Rodrigues (2015) criou um protótipo que tem como objetivo realizar a tradução de sentenças escritas em língua portuguesa para Libras. O protótipo conta com um pré-processamento que transcreve a frase ajustando as palavras que não contêm símbolos na Libras, utilizando palavras sinônimas que tenham símbolos respectivos para representar a palavra a ser substituída. Após o protótipo realiza a conversão da transcrição para vídeos que representam os sinais de Libras. Segundo Rodrigues (2015), os resultados foram satisfatórios, porém recomenda-se utilizar com apoio de um profissional da área. Pois o dicionário da Libras possui muitos sinais e algumas palavras possuem mais de um sinal.

Quadro 2 – Libras translator via web for mobile devices

Referência	Lima et al. (2012).
Objetivos	Auxiliar na comunicação por texto digitado ou por sinais.
Principais funcionalidades	Um chat que permite que a comunicação seja tanto por texto quanto por sinais, caso seja texto será realizado a tradução para sinais em Libras, caso sinal será realizado a tradução para texto.
Ferramentas de desenvolvimento	Foi desenvolvido em Java para dispositivos móveis na plataforma Linux, concebida em uma abordagem Cliente-Servidor.
Resultados e conclusões	Embora possua potencial educacional este ambiente não apresenta um ensino teórico sobre Libras. Também não possui uma forma de treino de forma evolutiva, mesmo permitindo ao usuário realiza um gesto e ver o resultado gerado. Este ambiente não tem a finalidade de ensinar a partir do treino, portanto é um software para conversação.

Fonte: adaptado de Lima et al. (2012).

Lima et al. (2012) construíram um chat que permite a comunicação tanto por texto quanto por sinais. O chat é capaz de traduzir sentenças em sinais de Libras e Sinais de Libras em textos. O aplicativo foi construído em camadas cliente servidor, sendo que o cliente apenas transmite as imagens do sinal e ela é classificada e traduzida no servidor. Essa abordagem foi necessária pois os dispositivos móveis não ofereciam capacidade suficiente devido aos recursos computacionais limitados.

Quadro 3 – Signa

Referência	Batista (2015).
Objetivos	Cadastrar e reconhecer sinais de Libras.
Principais funcionalidades	Reconhecimento e cadastro de sinais.
Ferramentas de desenvolvimento	Foi desenvolvido em C#, utilizando o equipamento Leap Motion e suas bibliotecas.
Resultados e conclusões	Não traz o ensino teórico dos conceitos básicos de Libras e uma forma de estudo, embora apresente os sinais previamente cadastrados a uma palavra, sendo considerado parcialmente didático, porém não permite um treino teórico.

Fonte: adaptado de Batista (2015).

Batista (2015) construiu um aplicativo para permitir o treinamento de sinais, utilizando de aprendizado de máquina para realizar a classificação do sinal, com um algoritmo que utiliza uma base de dados previamente cadastrados. Essa base de dados pode ser expandida pelo usuário para melhorar a classificação ou criar sinais. Segundo Batista (2015), o software apresenta limitações com sinais semelhantes e quando o sinal gera oclusão dos dedos para o Leap Motion.

3 DESCRIÇÃO DA SOFTWARE

Esta seção pretende apresentar os detalhes de especificação e implementação do software. Para tanto, são apresentados em três subseções. A subseção 3.1 apresenta a visão geral do software, explicando o comportamento dos módulos do software. A subseção 3.2 apresenta a funcionalidade destaques de animações, detalhando como utilizar e quais recursos se utilizou para o desenvolvimento. A subseção 3.3 explica como definir os critérios de aceitação, além de explorar o comportamento da comparação.

3.1 VISÃO GERAL

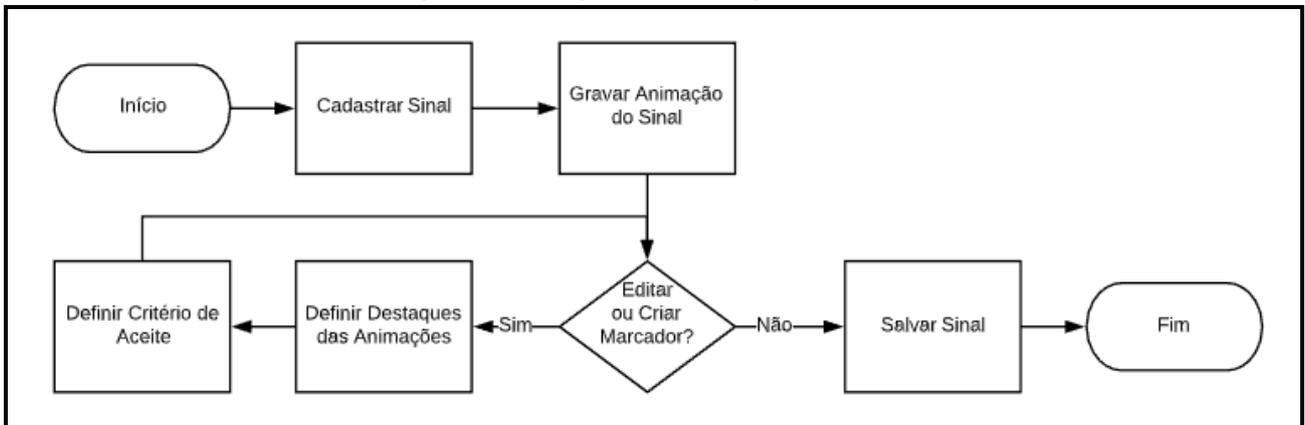
O software tem por objetivo auxiliar no aprendizado de Libras, possibilitando o profissional de Libras cadastrar sinais que serão usados para treino do aluno. Dessa forma, o software disponibiliza três módulos ao usuário: *Aprender os Sinais*, *Definições de Sinais* e *Treinamento*. O módulo *Aprender os Sinais* foi estendido do *LibRAR*, conforme seção 2.3, para permitir o aprendizado teórico dos sinais de Libras do alfabeto e números. Sinais esses que já passaram por processo de avaliação de um profissional da área (SILVA, 2018).

Os módulos *Definições de Sinais* e *Treinamento* foram criados usando outro modelo de mão, isso pois no trabalho anterior apenas trabalhava com a mão direita e nesse será possível interagir com as duas. Foi utilizado uma mão em formato de esqueleto para simular os movimentos das mãos, tornando visível o movimento das articulações. O módulo *Definições de Sinais* foi desenvolvido para permitir o profissional da área criar sinais, destaques nas animações dos sinais e definir critérios de aceitação para que o sinal seja aceito. Figura 5 apresenta o fluxograma, que demonstra o comportamento do módulo *Definições de Sinais*.

A primeira etapa do módulo é o cadastramento do sinal (Figura 6). O cadastramento permite ao usuário definir um ID para o sinal, esse que não pode ser um ID já utilizado por outro sinal e somente será usado para identificação interna, não sendo apresentado ao usuário. Para tal, deve ser cadastrado um nome ao sinal, que será apresentado ao usuário para identificação do sinal. Deve ser definido qual mão será utilizada para fazer o sinal, podendo optar-se por

usar as duas mãos ou apenas uma, conforme as configurações do sinal. Ao selecionar *Edita / Gravar Animação do Sinal* o usuário será direcionado a parte de gravação da animação do sinal (Figura 7).

Figura 5 – Fluxograma de Definições de Sinais



Fonte: elaborado pelo autor.

Figura 6 – Cadastramento de Sinais

Fonte: elaborado pelo autor.

Figura 7 – Gravação de Sinal

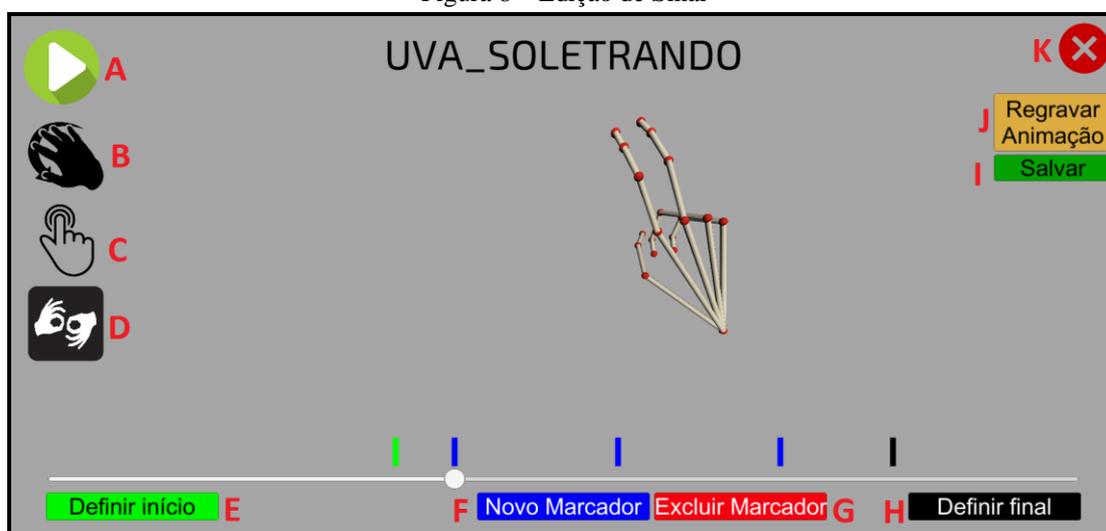
Fonte: elaborado pelo autor.

A gravação (Figura 7) utiliza o equipamento Leap Motion para fazer o rastreamento das mãos, e assim o usuário pode decidir o momento em que começa e termina a gravação, bem como é possível refazer a gravação. Utilizou-se o formato Unity Runtime Animation Recorder (NEWYELLOW, 2019) para realizar essa gravação, que é uma biblioteca Open Source para gravação de objetos no Unity. Com essa biblioteca é possível gravar os sinais em arquivos .anim que é utilizado pelo Unity para fazer as animações nos objetos. Para permitir cancelar a gravação se definiu que a gravação sempre utilizasse o arquivo lastRecorded.anim, assim ao utilizar o botão de salvar é copiado esse arquivo para o nome do sinal que deve ser gravado. Essa biblioteca matem todas as translações e rotações do objeto que está sendo gravado em *frames* por segundo.

Para poder utilizar essa biblioteca algumas alterações foram necessárias, pois ela gravava muitos *frames* por segundo deixando os arquivos muito grande. Dessa forma, foi reduzido para cinco *frames* por segundo, fazendo com que o tamanho do arquivo gerado ficasse muito menor do que o original. Embora não grave exatamente todos os movimentos, esses são supridos pela execução do Unity sobre os arquivos .anim, pois quando executado, o Unity realiza os movimentos entre um *frame* até o outro modificando as translações e rotações de forma gradual. Dessa forma, o resultado desta transição gradual entre dois *frames* passa uma sensação de uma animação contínua, mesmo que se utilize apenas cinco *frames* por segundo.

Após o sinal gravado o usuário vai para a parte de edição do sinal (Figura 8). Nesse momento o usuário pode criar ou editar marcadores em qualquer parte da gravação do sinal, utilizando o botão deslizante. Na Figura 8 pode-se observar os botões da tela de edição do sinal. O botão A) serve para rodar a animação podendo ver como ficou os movimentos. Os botões B), C) e D) apenas são apresentados quando o botão deslizante estiver sobre um marcador, pois esses itens são para configuração do marcador (respectivamente definir rastros, pontos de atenção e critérios de aceite). Já os botões E) e H) serve para definir onde começa e termina o sinal, e esses serão aplicados onde o botão deslizante estiver posicionado. Os botões F) e G) servem para criar ou excluir um marcador que estiver na posição do botão deslizante. O botão I) é para salvar as alterações feitas até o momento. Já o botão J) é para redirecionar para a gravação de sinal, onde poderá sobrescrever o sinal gravado anteriormente por um novo. O botão K) serve para voltar para a tela de cadastramento do sinal.

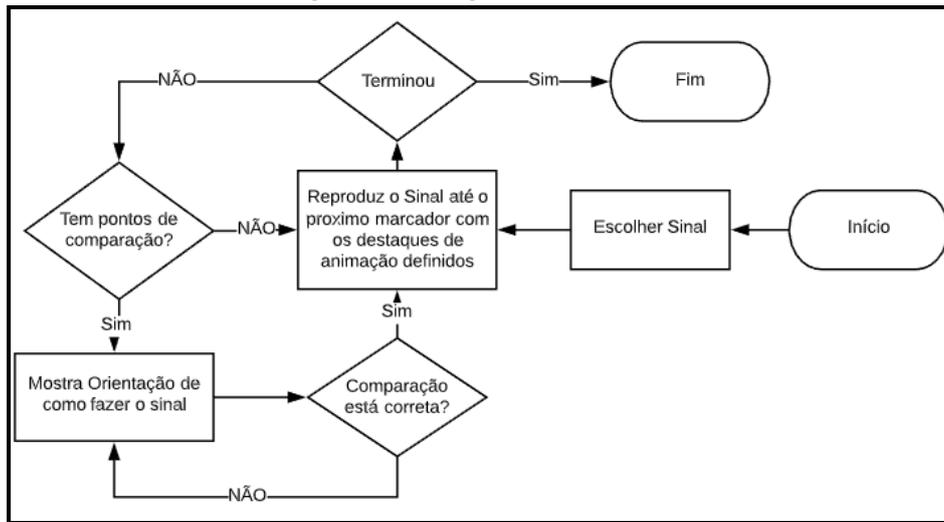
Figura 8 – Edição de Sinal



Fonte: elaborado pelo autor.

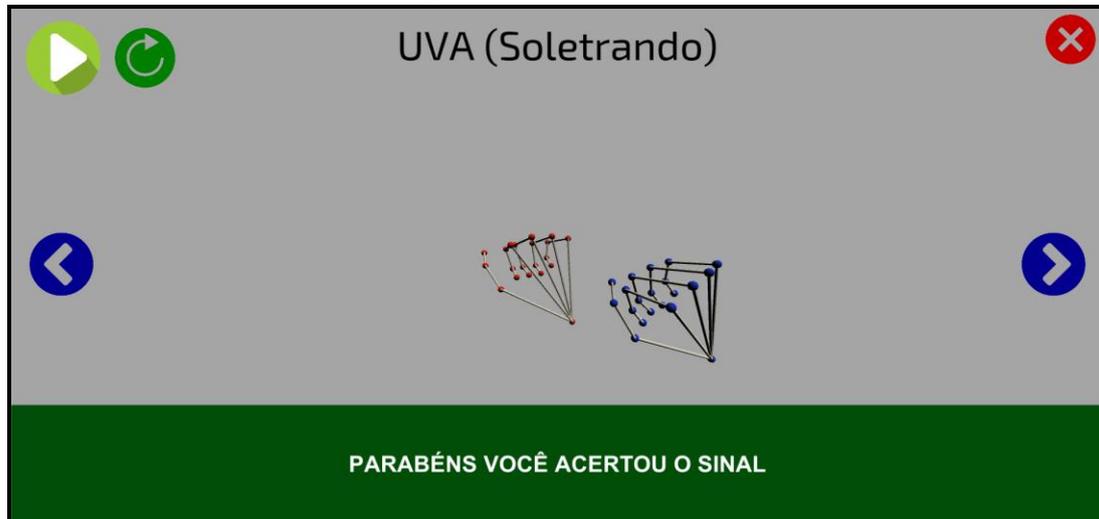
Finalizando a definição do sinal, ele já ficará disponível no módulo Treinamento, que permite ao usuário treinar o sinal previamente cadastrado, apresentando a mão gravada no módulo Definições de Sinal e a mão do usuário a partir do equipamento Leap Motion. A Figura 9 apresenta o comportamento do módulo e as etapas que se passam desde a escolha do sinal até quando o usuário completar o sinal. Esse fluxograma apresenta o comportamento da funcionalidade de comparação de sinais. Após escolher o sinal, ele começa a ser reproduzido, aplicando os destaques de animações definidos durante o módulo Definições de Sinais. O sinal é reproduzido até que ache outro marcador, caso seja o marcador de fim, apresenta uma mensagem ao usuário indicando que finalizou o sinal (Figura 10). Enquanto não encontrar o marcador de fim será aplicado os destaques de animação definidos no marcador, caso o marcador tenha critérios de aceitação definidos, será apresentado ao usuário as orientações (Figura 11) de como realizar o sinal e enquanto não for atendido os critérios definidos o sinal não continua a reprodução.

Figura 9 – Fluxograma Treinamento



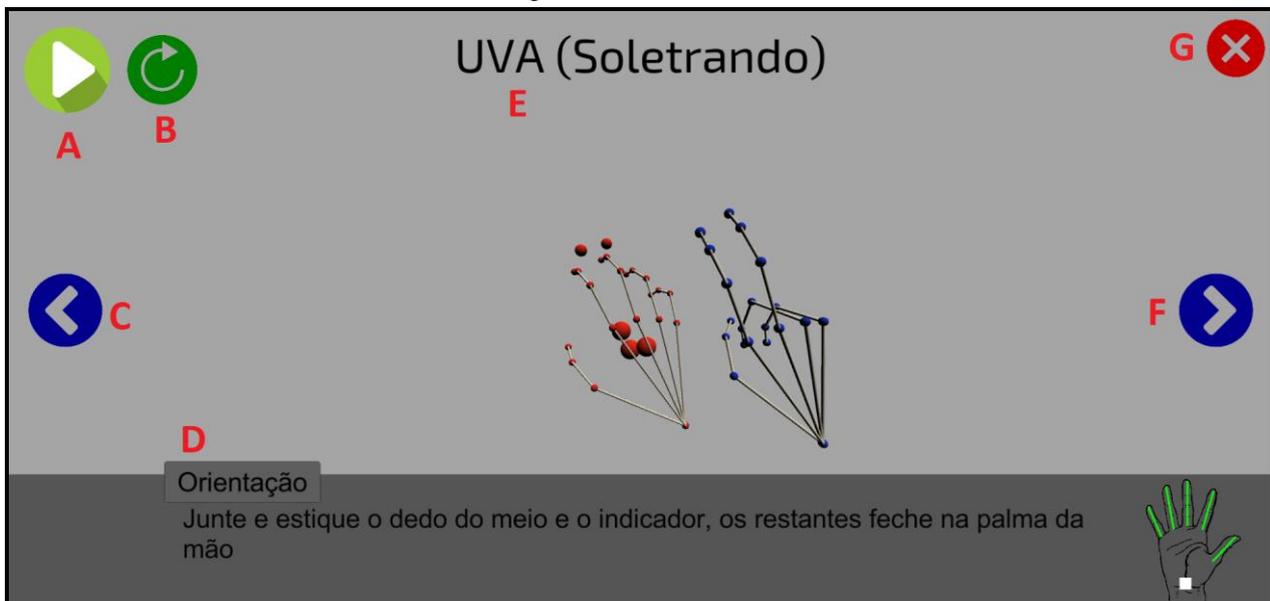
Fonte: elaborado pelo autor.

Figura 10 – Término do sinal



Fonte: elaborado pelo autor.

Figura 11 – Treinamento



Fonte: elaborado pelo autor.

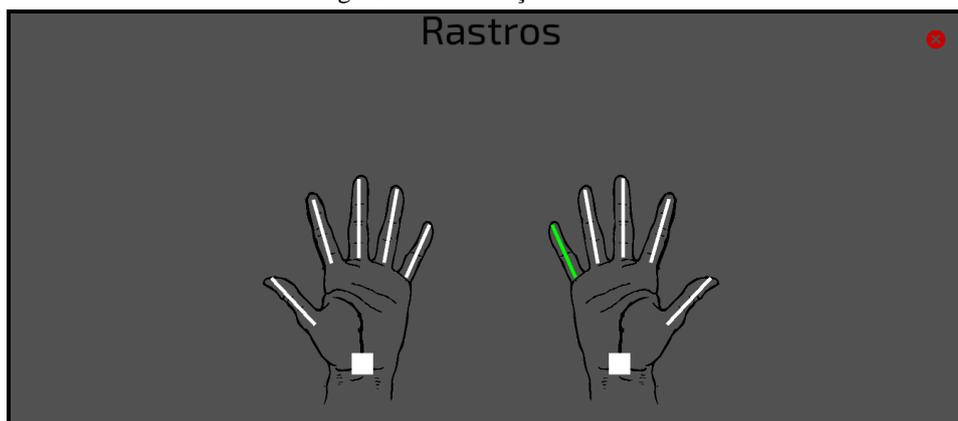
A Figura 11 demonstra o momento de comparação do módulo *Treinamento*, os botões C) e F) são para navegação entre os sinais (anterior e próximo respectivamente). O botão A) é utilizado para parar ou retomar a animação do sinal. Já o B) para reiniciar a animação do sinal. O último botão é o G) que é utilizado para voltar ao menu principal. A marcação E) é onde apresenta o nome dado ao sinal durante a *Definições de Sinais*. Por último na parte inferior a marcação D) é apresentada uma orientação textual para auxiliar na realização do sinal. Orientação que é cadastrada pelo módulo *Definições de Sinais* durante a definição do critério de aceite. Também no centro da tela é apresentada as mãos que são utilizadas no sinal e quais pontos estão sendo comparados.

A visualização da mão gravada pelo software sempre terá as juntas na cor azul, de ambas as mãos (direita ou esquerda), quando os sinais as utilizarem. Para as mãos do usuário que está sendo captada pelo Leap Motion, essas são representadas de vermelho e amarelo, respectivamente a mão direita e esquerda. Para sinalizar onde os dedos deveriam estar no momento de comparar se os dedos estão na posição correta (quando um dedo não se enquadrar nos critérios definidos pelo profissional), será apresentado um círculo indicando a área aceitável onde o dedo deveria estar (o círculo estará na mesma cor que as juntas da mão a qual se refere).

3.2 DESTAQUES DE ANIMAÇÕES

Os destaques de animações são utilizados chamar atenção para alguma parte específica da mão do usuário, podendo ser adicionado rastros ou pontos de atenção. Esses destaques serão apresentados ao usuário no módulo *Treinamento* e definidos no módulo *Definições de Sinais*. Durante o módulo *Definições de Sinais* o usuário pode definir rastros utilizando o botão B) da Figura 8, sendo direcionado a tela de rastros (Figura 12), que lhe permite definir rastros para qualquer dedo da mão ou para o pulso. Ao selecionar algum dedo para utilizar rastros ele fica na cor verde.

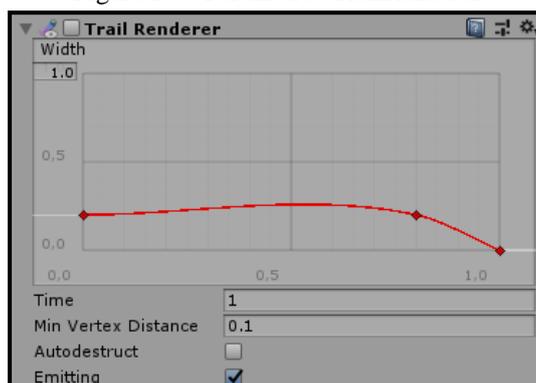
Figura 12 – Definição de rastros



Fonte: elaborado pelo autor.

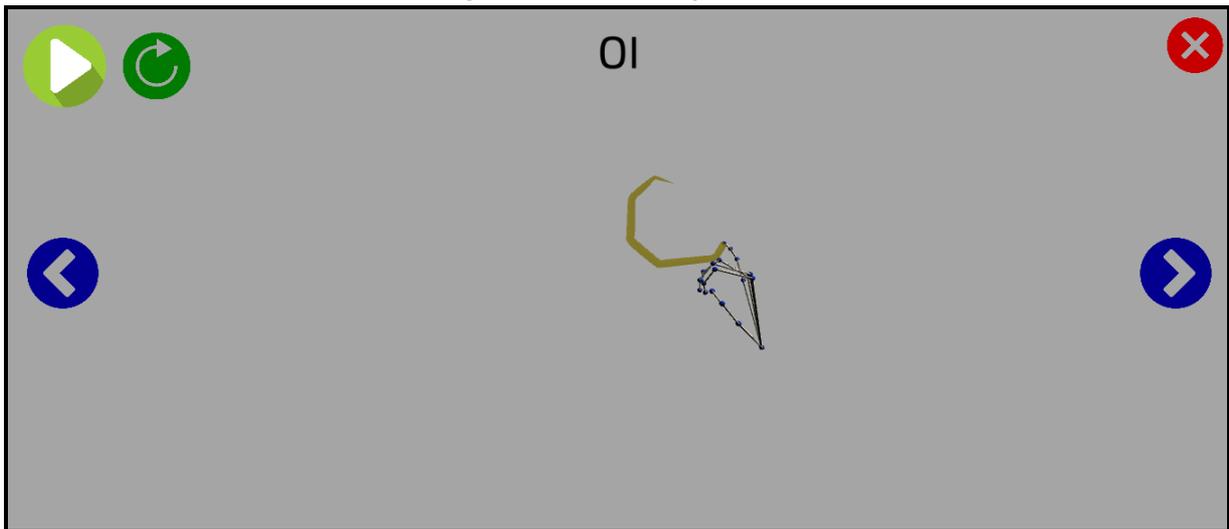
Para os rastros utilizou-se o componente *TrailRenderer* (UNITY, 2019) que é utilizado para fazer um rastro no espaço 3D. Esse componente recebe uma configuração de curva para realizar o rastro. Dessa forma é aplicado a mesma curva a todos os pontos da mão, porém somente são ativados quando a reprodução do sinal passar por um marcador que tiver definição de rastros. O parâmetro de curva utilizada pode ser visualizada na Figura 13, utilizando 1 segundo para o desaparecimento do rastro, iniciando com uma espessura de 0.2, no segundo 0.8 a espessura é mantida em 0.2 e ao chegar no final do segundo a espessura do traço é 0. O resultado obtido ao utilizar-se o rastro pode ser observada na Figura 14, que é marcado por onde o dedo passou no espaço 3D.

Figura 13 – Parâmetros TrailRenderer



Fonte: elaborado pelo autor.

Figura 14 – Demonstração rastro



Fonte: elaborado pelo autor.

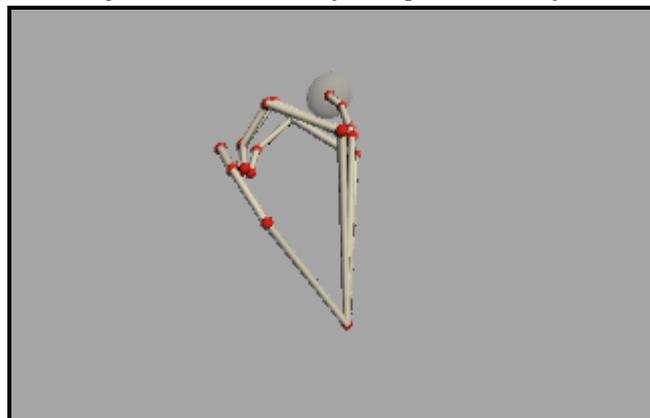
Os pontos de atenção são criados desenhando círculos na ponta do dedo selecionado (Figura 15) durante edição da animação ao pressionar o botão C) da Figura 8. Para essa animação é utilizada um círculo de raio zero com opacidade e incrementado 0,0005 a cada milissegundo, até atingir 300 milissegundos recebendo novamente raio zero. O círculo desenhado para o ponto de atenção pode ser observado na Figura 16.

Figura 15 – Definição de pontos de atenção



Fonte: elaborado pelo autor.

Figura 16 – Demonstração de ponto de atenção



Fonte: elaborado pelo autor.

3.3 COMPARAÇÃO

Para comparação se utilizou parâmetros cadastrados durante a edição do sinal no módulo definição de sinais. O botão D) da Figura 8 acessa a tela de definição de critérios de aceitação (Figura 17). Na tela o usuário pode informar uma orientação textual, essa será demonstrada ao usuário enquanto em uma comparação a mão do usuário não atender os critérios de aceitação definidos, a definição pode ser feita selecionando qualquer dedo ou o pulso para realizar a comparação, ao selecionar, o usuário pode escolher ativar a comparação e ao ativar também consegue definir um valor para o critério de aceitação. O valor definido pelo usuário é demonstrado uma previa do espaço permitido no círculo que é desenhado na ponta do dedo, que demonstra previamente a margem de aceitação.

Figura 17 – Definição de pontos de atenção



Fonte: elaborado pelo autor.

Quadro 4 – Funções de comparação entre a mão gravada e a mão do usuário

```
internal static bool Compare(MarkerAnimation marker, DrawHands handRecorded, DrawHands handTrack)
{
    bool result = true;
    foreach (var pointCompare in marker.Compare)
    {
        var pointCompareRecorded = handRecorded.GetPoint(pointCompare.part);
        var pointCompareTrack = handTrack.GetPoint(pointCompare.part);

        var matrixTracked = handTrack.GetLocalToWorldMatrixReference(pointCompare.part.Substring(0, 1));
        var matrixRecorded = handRecorded.GetWorldToLocalMatrixReference(pointCompare.part.Substring(0, 1));

        var pointProjectionTrack = matrixTracked.MultiplyPoint(matrixRecorded.MultiplyPoint(pointCompareRecorded.transform.position));

        var distance = (pointProjectionTrack - pointCompareTrack.transform.position).magnitude;
        // devido a escala da mão é necessario dividir o valor aceitavel em 5 vezes;
        var distanceAcceptable = pointCompare.value / 5;

        if (distance > distanceAcceptable || distance < -distanceAcceptable)
        {
            result = false;
            if (handTrack.DrawHand(pointCompare.part.Substring(0, 1)))
            {
                drawSphere(pointProjectionTrack, distanceAcceptable, handTrack, pointCompare.part.Substring(0, 1));
            }
        }
    }
    return result;
}
```

Fonte: Schubert (2003, p. 63).

O usuário pode definir quantos dedos ele quiser comparar sobre o marcador. A comparação é realizada utilizando a matriz de transformação das palmas das mãos, caso utilize apenas a mão direita ou ambas as mãos, é utilizado a matriz de transformação da mão direita, e caso utilize apenas a mão esquerda é utilizado a matriz de comparação da mão esquerda. Dessa forma quando o usuário utiliza a comparação conforme Quadro 4, é multiplicado o ponto da mão gravada com a matriz de transformação do universo para a palma da mão, após é multiplicado a matriz de transformação da mão rastreada da palma da mão para o universo, assim é projetado o ponto no universo que deveria estar o dedo do usuário, se a distância entre esse ponto e a ponta do dedo do usuário for igual ou inferior a margem de

aceitação definida, a comparação daquele ponto é aceita. Enquanto não for aceita é desenhado um círculo no local onde o dedo deveria estar posicionado, com o raio da margem de aceitação definida. Esse processo de comparação acontece para todos os dedos que contiverem definição de comparação.

Ao cadastrar um sinal com movimento da mão pode ser utilizado a comparação com o pulso, quando comparado o pulso pela primeira vez é guardado a matriz de transformação das mãos (do usuário e gravada). Durante as próximas comparações é utilizado a mesma matriz de transformação utilizada na primeira comparação do pulso. Assim o usuário precisará realizar o movimento relativo à posição em que a mão dele.

4 RESULTADOS

Esta seção apresenta os testes realizados com o software. Na subseção 4.1 é apresentado os testes realizados a fim de validar os objetivos do trabalho. Na subseção 4.2 apresenta as entrevistas realizadas com os profissionais de Libras que auxiliaram no desenvolvimento do software.

4.1 TESTES DE FUNCIONALIDADE

Para validação dos objetivos foi separado os testes em características dos sinais de Libras. As características avaliadas foram Configuração de mãos (diferenciar configurações de mãos parecidas), movimento (exigir que o usuário realize a movimentação com a mão que o sinal faz), orientação (exigir que o usuário mantenha a orientação do sinal conforme o sinal gravado) e soletração (permitir testar todas as letras das palavras).

Para testar as configurações de mãos foram utilizados os sinais U e V , ambos são parecidos, distinguindo-os apenas na distância entre o dedo do meio e o indicador, sendo que, para o sinal U deve obrigatoriamente estar juntos o dedo do meio e o indicador, enquanto o sinal V deve apenas estar separados. Durante os testes ao definir margem de aceitação menores para o U , o módulo `Treinamento` força o usuário a manter os dedos juntos, e quando definido para o V , deixando uma margem de aceitação que não permita manter os dedos juntos, faz com que o usuário tenha que afastar o dedo do meio do indicador, realizando o sinal V corretamente. Obtendo resultado para a configuração de mão OK.

Para testar o movimento foi cadastrado o sinal `uva`. Esse sinal não é alterado a configuração de mãos, porém há um movimento da mão no sinal. Ao definir critérios de aceitação que utilizem o pulso, no módulo `Treinamento` a comparação força o usuário a movimentar a mão de forma relativa. Obtendo resultado para o movimento OK. Para testar a orientação foi utilizado a soletração da palavra `uva`, foi definido os critérios de aceitação no sinal. Ao testar no módulo `Treinamento` foi aceito sinais das letras com orientação incorreta. Obtendo resultado para a orientação NÃO OK.

Para testar a soletração dos sinais, foram utilizados os sinais `uva` e `casa`. Ambos foram cadastrados como soletrações das palavras e definido critérios de aceitação para cada letra. Ao Utilizar no módulo `Treinamento` o software respeitou as comparações forçando o usuário a reproduzir corretamente cada letra que foi definido critério de aceitação. Obtendo resultado para soletração OK.

Durante os testes das funcionalidades algumas configurações de mãos não foram reconhecidas corretamente, ao testar a soletração da palavra FURB observou-se que as letras F e R não foram rastreadas pelo Leap Motion de forma correta. O problema é conforme relatado por Batista (2015, p. 86) “De forma geral, os sinais que obtiveram o resultado NÃO OK apresentam um nível de oclusão em relação ao Leap Motion”. Para obter-se uma amostragem foi conduzido um teste com as letras do alfabeto, das vinte e seis letras apenas treze foram reconhecidas corretamente, cinquenta por cento. Entre as letras que não foram reconhecidas corretamente quatro delas, equivalente a quinze por cento, o reconhecimento foi intermitente, em que na maioria das vezes não reconhecia corretamente. Nove letras do alfabeto não foram reconhecidas pelo equipamento em qualquer momento dos testes, resultando em aproximadamente trinta e cinco por cento que não foram reconhecidas em nenhum momento.

4.2 ENTREVISTAS COM PROFISSIONAL DE LIBRAS

Durante o desenvolvimento se buscou orientação de profissionais da área para avaliar como melhor realizar comparações nos treinamentos de Libras, pois não se tinha muito conhecimento de qual a acuracidade necessária em cada dedo no momento de realizar um sinal de Libra. Realizou-se quatro entrevistas com os profissionais da área, a primeira (SANTOS, 2019a) com participação da Marisa introduziu-se o Leap Motion, demonstrado problemas de oclusão dos dedos do Leap Motion e avaliou-se algumas configurações de mãos que o Leap Motion reconhece de forma mais assertiva. Nessa foi destacado como sinal de exemplo a comparação entre os sinais U e V , ambos são parecidos diferenciando a distancia entre o dedo do meio e o indicador, que necessariamente o U deve estar ambos os dedos juntos, sem espaçamento, enquanto o V basta os dedos não estarem juntos, mas podendo ter uma distância variada, permitindo uma margem maior para o critério de aceitação do sinal.

Durante a segunda entrevista (SANTOS, 2019b) foi demonstrado um protótipo do módulo *Definições de Sinais*. Nessa entrevista ainda não se tinha desenvolvido o critério de aceite, e se discutiu sobre pontos importantes e como a professora de Libras pensava sobre a mão. Dessa forma, inicialmente a possibilidade de definir pontos de atenção eram em qualquer junta da mão, pois se acreditava ser mais assertiva, porém a Marisa (SANTOS, 2019b) explicou que geralmente a referência aos pontos das mãos se dão nos ossos dos dedos e não nas juntas. Então se alterou a implementação para que a configuração passasse a ser por ossos dos dedos ao invés das juntas. Nessa entrevista também se decidiu dois sinais que serviriam como exemplos para demonstração das funcionalidades, no caso do sinal “uva” e “casa”. E para permitir variações também foi feito os mesmos sinais soletrando-os. Também se solicitou uma forma de explicar como deve ser reproduzido o sinal, para tal introduziu-se a orientação textual, que é configurado pelo profissional da área durante a *Definições de Sinais* e apresentada durante o *Treinamento*.

Na terceira entrevista (SANTOS; GARRAO, 2019) o professor Patrício participou da reunião. O professor não conhecia o dispositivo Leap Motion, então se apresentou o dispositivo juntamente com os módulos *Definições de Sinais* e *Treinamento* que estavam parcialmente desenvolvidos. Como sugestões dessa reunião se melhorou a definição do critério de aceite do módulo *Definições de Sinais*, entregando um retorno visual quando é definido a distância permitida para cada dedo que será comparado. Também se adicionou uma mensagem sinalizando que o usuário completou o sinal. Outra alteração sugerida foi em relação a orientação do Leap Motion no espaço 3D. Com essa alteração o desenho das mãos no software não ficara mais espelhadas, mas acompanham o movimento do usuário no mundo real, assim ao mover a mão para frente no mundo real a mão desenhada no software também é movida na mesma direção.

Na última reunião (SANTOS, 2019c) feita somente com a Marisa se demonstrando a versão final do software. Nesta demonstração foram apresentadas algumas novas funcionalidades, como: orientações no treinamento e treinamento de sinais com movimento. O sinal de exemplo utilizado para teste da funcionalidade de sinais de movimento foi o sinal de uva. Que permitiu validar além da posição relativa entre as duas mãos, o movimento que umas das mãos deveriam fazer para que o sinal estivesse correto.

5 CONCLUSÕES

Devido a extensão do *Libras* o software atende ao objetivo específico disponibilizar ao usuário animações que explicam os conceitos básicos da Libras, que permite ao usuário visualizar como se realiza todos os números e letras em Libras. O módulo *Definições de Sinais* atende ao objetivo específico disponibilizar uma ferramenta ao profissional de Libras que permitirá cadastrar e definir os critérios de aceitação do sinal, que entrega em editor de sinal com os requisitos de cadastramento e definições dos critérios de aceitação. O módulo *Treinamento* atende o terceiro objetivo específico, permitir que os usuários treinem suas configurações de mão na reprodução do sinal de Libras de acordo com os critérios estabelecidos pelo profissional de Libras, que permite ao usuário treinar suas configurações de mãos. Assim como permite qualquer usuário seja surdo ou não treinar sinais de Libras, contribuindo socialmente para a comunicação dos surdos. Este fator está associado com a inclusão, uma vez que aumenta a predisposição do aprendizado de Libras.

O software apresenta potencial no treinamento, encontrou-se limitações que, caso resolvidas, podem ampliar a sua utilização. Ao se utilizar o dispositivo Leap Motion com o rastreamento de mãos do dispositivo várias configurações de mãos não são representadas de forma correta no mundo virtual. O dispositivo, por exemplo, não foi capaz de rastrear cinquenta por cento do alfabeto de forma correta. Como existem muitas configurações de mãos definidas por orientação ou por movimentação, a oclusão dos dedos decorrente destes movimentos não permitiu o Leap Motion ter um reconhecimento efetivo.

Outro fator de limitação do software é que durante a animação se realiza a comparação de forma relativa na mão do usuário. Porém da mesma forma que isso dá mais liberdade ao usuário para não ficar preso a mão gravada e ser capaz de realizar ao lado da gravação, isso permite ao usuário realizar o sinal com a orientação da palma da mão de forma incorreta, poderá aceitar um sinal reproduzido errado pelo usuário devido a orientação utilizada na reprodução do sinal. Por fim, as possíveis extensões mapeadas durante o desenvolvimento e testes do software são:

- a) utilizar outro dispositivo ou algoritmo de rastreamento das mãos que possibilitem uma melhor precisão em casos de oclusão de dedos da mão;
- b) utilizar outro dispositivo ou algoritmo para rastreamento do corpo e cabeça que possibilitem o reconhecimento de sinais ancorados ao corpo do usuário;
- c) alterar a forma de comparação entre a mão gravada e do usuário de forma que possibilite a comparação correta da orientação das mãos;
- d) utilizar técnicas de educação para melhorar o aprendizado de Libras;
- e) realizar mais testes do aplicativo com uma quantidade maior de alunos para identificar mais possibilidades de melhorias e ampliar a quantidade de pessoas que conseguem utilizar o aplicativo.

REFERÊNCIAS

- ARAÚJO, Ana Paula de. **Língua Brasileira de Sinais (LIBRAS)**. [S.l.], [2009]. Disponível em: <http://www.infoescola.com/portugues/lingua-brasileira-de-sinais-libras/> Acesso em: 06 nov. 2018.
- BATISTA, Júlio César. **SIGNA: uma aplicação para ensino-aprendizagem da Libras**. 2015. 115 f. Monografia (Especialização) - Curso de Bacharelado em Ciência da Computação, Centro de Ciências Exatas e Naturais, Universidade Regional de Blumenau, Blumenau, 2015.
- BENTO, Nanci A. **Os parâmetros fonológicos: configuração de mãos, ponto de articulação e movimento na aquisição da Língua Brasileira de Sinais – um estudo de caso**. 2010. 143f. Dissertação (Mestrado em Letras e Linguística) – Programa de Pós-Graduação em Letras e Linguística, Universidade Federal da Bahia, Salvador.
- BRITO, Lucinda. **Por uma gramática de língua de sinais**. Rio de Janeiro: Tempo Brasileiro, 1995. 273 p.
- CARVALHO, T. S. S.; TAVARES, I. M. S. Inclusão escolar e a formação de professores para o ensino de Libras (Língua Brasileira de Sinais): do texto oficial ao contexto. In: **Pesquisa em Educação: Desenvolvimento, ética e responsabilidade social**, 2010. Maceió. V EPEAL.
- FERREIRA, Adir L. et al. **Aprendendo Libras: Módulo 2**. Natal: EDUFERN, 2011. Disponível em: http://sedis.ufrn.br/bibliotecadigital/site/pdf/TICS/Livro_MOD2_LIBRAS_Z_WEB.pdf Acesso em: 06 nov. 2018.
- IBGE. **Censo demográfico 2010: Características gerais da população, religião e pessoas com deficiência**. Rio de Janeiro, 2012. Disponível em: http://biblioteca.ibge.gov.br/visualizacao/periodicos/94/cd_2010_religiao_deficiencia.pdf Acesso em: 20 set. 2018.
- LACERDA, Cristina Broglia Feitosa de. A inclusão escolar de alunos surdos: o que dizem alunos, professores e intérpretes sobre esta experiência. **Cad. Cedes: educação, surdez e inclusão social**, Campinas, v. 26, n. 69, p.163-184, 2006. Bimestral.
- LEHMANN, Harold P. et al. An Ethnographic, Controlled Study of the Use of a ComputerBased Histology Atlas during a Laboratory Course. **Journal of the American Medical Informatics Association : JAMIA**, [Oxford], v. 6, n. 1, p. 38–52 jan/fev. 1999.
- LIMA, M. A. S. et al. LIBRAS translator via web for mobile devices. **Proceedings Of The 6th Euro American Conference On Telematics And Information Systems - Eatis '12**, [s.l.], p.1-4, 2012. ACM Press. <http://dx.doi.org/10.1145/2261605.2261670>.
- LUZ, Renato Dente. **Cenas Surdas: os surdos terão lugar no coração do mundo?**. São Paulo: Parábola, 2013. 192p.
- MELO, Alain Rosemberg Lívio Linhares de. **Um estudo sobre o mapeamento de gestos do leap motion para a língua brasileira de sinais (Libras)**. 2015. 97 f. Dissertação (Mestrado) - Curso de Pós-graduação em Ciência da Computação, Centro de Ciência e Informática, Universidade Federal de Pernambuco, Recife, 2015.
- NEWYELLOW. **GitHub - newyellow/Unity-Runtime-Animation-Recorder: Record animations in Unity runtime. Can save to .anim, maya, or FBX ASCII format**. [S.l.], 2019. Disponível em <https://github.com/newyellow/Unity-Runtime-Animation-Recorder> Acesso em: 15 nov. 2019.
- QUADROS, Ronice Müller de; KARNOPP, Lodenir Becker. **Língua de sinais brasileira: estudos lingüísticos**. Porto Alegre : Artmed, 2004. 221 p.
- RODRIGUES, Jonathan. **Protótipo de um tradutor de textos de língua portuguesa para Libras**. 2015. 43 f. TCC (Graduação) - Curso de Ciência da Computação – Bacharelado, Centro de Ciências Exatas e Naturais, Universidade Regional de Blumenau, Blumenau, 2015.
- SANTOS, Marisa G.B. dos. **Entrevista explicando o Leap Motion e entendendo as características de Libras**. Entrevistadores: Matheus Adriano Pereira e Dalton Solano dos Reis. Blumenau. 2019a. Entrevista feita através de conversação – não publicada.
- SANTOS, Marisa G.B. dos. **Entrevista apresentando o protótipo do módulo Definir Sinais**. Entrevistadores: Matheus Adriano Pereira e Dalton Solano dos Reis. Blumenau. 2019b. Entrevista feita através de conversação – não publicada.
- SANTOS, Marisa G.B. dos. **Entrevista apresentado o software finalizado**. Entrevistadores: Matheus Adriano Pereira e Dalton Solano dos Reis. Blumenau. 2019c. Entrevista feita através de conversação – não publicada.
- SANTOS, Marisa G.B. dos; GARRAO, Patrício V. **Entrevista explicando o Leap Motion e módulos parcialmente desenvolvidos**. Entrevistadores: Matheus Adriano Pereira e Dalton Solano dos Reis. Blumenau. 2019. Entrevista feita através de conversação – não publicada.
- SILVA, Luan Ribeiro da. **Librar: conceitos básicos de libras usando realidade aumentada e realidade virtual**. 2018. 74 f. Monografia (Especialização) - Curso de Bacharelado Ciências da Computação, Centro de Ciências Exatas e Naturais, Universidade Regional de Blumenau, Blumenau, 2018.
- SILVA, Lúcia Marta Giunna da et al. Comunicação não-verbal: reflexões acerca da linguagem corporal. **Revista Latino-Americana de Enfermagem**, Ribeirão Preto, v. 8, n. 4, p. 52-58, ago. 2000. Disponível em http://www.scielo.br/scielo.php?script=sci_arttext&pid=S0104-1169200000400008&lng=en&nrm=iso. Acesso em 19 Nov 2019.
- ULTRALEAP LTD. **Lançamentos Ultraleap, prontos para redefinir a interação - Leap Motion**. [S.l.], 2019a. Disponível em: <https://www.leapmotion.com/news/ultraleap-launches-ready-to-redefine-interaction/> Acesso em: 26 nov. 2019.

ULTRALEAP LTD. **Visão geral da API - documentação do Leap Motion C # SDK v3.2 Beta.** [S.l.], 2019b. Disponível em: https://developer-archive.leapmotion.com/documentation/csharp/devguide/Leap_Overview.html Acesso em: 26 nov. 2019.

ULTRALEAP LTD. **Comunicação WebSocket - documentação do Leap Motion JavaScript SDK v3.2 Beta.** [S.l.], 2019c. Disponível em: https://developer-archive.leapmotion.com/documentation/javascript/supplements/Leap_JSON.html Acesso em: 26 nov. 2019.

UNITY. **Unity - Manual: Trail Renderer.** [S.l.], 2019. Disponível em <https://docs.unity3d.com/Manual/class-TrailRenderer.html> Acesso em: 15 nov. 2019.

XU, Deyou. A Neural Network Approach for Hand Gesture Recognition in Virtual Reality Driving Training System of SPG. **18th International Conference On Pattern Recognition (icpr'06)**, [s.l.], p.519-522, 2006. IEEE. <http://dx.doi.org/10.1109/icpr.2006.109>.

WEICHERT, Frank et al. Analysis of the Accuracy and Robustness of the Leap Motion Controller. **Sensors**, [s.l.], v. 13, n. 5, p.6380-6393, 14 maio 2013. MDPI AG. <http://dx.doi.org/10.3390/s130506380>.