

RALibras: um protótipo *offline* para introduzir o ensino de Libras

Guilherme da Silva Mottin¹, Sílvia de Castro Bertagnoli¹

¹Instituto Federal de Educação, Ciência e Tecnologia do Rio Grande do Sul – Campus Porto Alegre (IFRS)
90.030-041 – Porto Alegre – RS – Brasil

Resumo. *Este trabalho relata o desenvolvimento do protótipo de um aplicativo que tem como objetivo auxiliar no ensino da Língua Brasileira de Sinais (Libras), por meio da tecnologia de Realidade Aumentada (RA) – o RALibras. Para o seu desenvolvimento, foi realizada uma pesquisa bibliográfica e exploratória. O protótipo, atualmente, exibe o alfabeto em Libras com o auxílio de um marcador, sem a necessidade de conexão com a internet. A ideia original era validar o protótipo na disciplina de Libras, ofertada a diversos cursos do IFRS, no início de 2020. Porém, devido à suspensão das atividades letivas com a pandemia de Covid-19, o aplicativo ainda não pôde ser avaliado pelos usuários reais.*

1. Cenário de Uso

Conforme dados do censo do Instituto Brasileiro de Geografia e Estatística (IBGE) de 2010, aproximadamente 5% dos brasileiros possuem algum nível de surdez, dentre os quais 27% possuem a “surdez profunda”, ou seja, não conseguem ouvir nada. A única forma de comunicação que muitas pessoas com deficiência auditiva possuem é o uso da Língua Brasileira de Sinais (Libras).

Preocupando-se com a inclusão da Libras no contexto educacional, o Decreto 5.626 de 2005 [Brasil 2005] instituiu que o ensino de Libras deve ser realizado através de “disciplina curricular obrigatória nos cursos de formação de professores para o exercício do magistério ... e nos cursos de Fonoaudiologia”. Por cursos de formação de professores, compreendem-se cursos de licenciatura, o curso normal médio e o superior, o curso de Pedagogia e o curso de Educação Especial. Esse decreto ainda estabelece que “A Libras constituir-se-á em disciplina curricular optativa nos demais cursos de educação superior e na educação profissional”.

Em 2017, a Libras passou a integrar uma das competências gerais da BNCC (Base Nacional Comum Curricular) [Mec 2017]: “Utilizar diferentes linguagens – verbal (oral ou visual-motora, como Libras, e escrita), corporal, visual, sonora e digital –, bem como conhecimentos das linguagens artística, matemática e científica, para se expressar e partilhar informações, experiências, ideias e sentimentos em diferentes contextos e produzir sentidos que levem ao entendimento mútuo.” (grifo nosso). Assim, percebe-se a importância que essa língua tem para a comunidade surda, o que gera reflexos na educação.

Nesse sentido, o presente trabalho descreve o *software* RALibras, que foi planejado para oferecer suporte ao ensino da Libras por meio da tecnologia de Realidade Aumentada (RA). Espera-se que, com ele, seja possível vencer as limitações que mídias comuns no ensino (por exemplo, o papel) têm ao representarem sinais complexos, que exigem uma

visualização em três dimensões (3D); e proporcionar aos estudantes um maior interesse no aprendizado [Lopes 2019]. Visando a ampliar o acesso ao sistema, o RALibras está sendo desenvolvido para a plataforma Android, visto que ela é a mais difundida no Brasil.

Quando finalizado, o público que o aplicativo atenderá são alunos de todos os níveis que tiverem interesse em aprender a linguagem de sinais por meio de uma tecnologia diferenciada, usando um *smartphone* com sistema operacional Android (na versão 5, no mínimo). O sistema foi pensado para uso de modo *online* e *offline*: no primeiro, preveem-se as funcionalidades de *upload* e *download* de sinais pelos usuários, criação de conjuntos de sinais e compartilhamento desses conjuntos entre os usuários; no segundo, projeta-se a capacidade de prover a experiência de RA com os modelos tridimensionais armazenados no celular. No contexto deste artigo, será apresentado o protótipo inicial do RALibras, além de se abordarem algumas decisões tecnológicas que viabilizaram o uso da RA na plataforma Android.

2. Desenvolvimento

O início do desenvolvimento do RALibras caracterizou-se pela pesquisa bibliográfica e exploratória dos *frameworks* de suporte à RA disponíveis para o sistema operacional Android. Durante esse processo, foram encontradas opções que vêm sendo aprimoradas há alguns anos por grandes empresas que são referência para a implementação de *softwares* nessa área, porque oferecem ferramentas testadas e documentadas. Entretanto, a grande maioria dessas soluções não é totalmente gratuita – exemplo disso é o Vuforia, o qual permite que se desenvolva a aplicação com a tecnologia; mas, para que ocorra o lançamento do sistema, é cobrada uma taxa anual de licenciamento [Vuforia 2018]. As opções restantes após a exclusão daquelas que exigem aporte financeiro em algum momento do ciclo de desenvolvimento foram o ARCore (desenvolvido pela Google) e o A-Frame (de código aberto).

O ARCore possui ferramentas fundamentais para o desenvolvimento de aplicações de RA que dependem do reconhecimento do ambiente ao redor do usuário, ou que necessitem oferecer RA compartilhada entre dispositivos [Arcore 2019]. Apesar de todas as características positivas, esse *framework* não atende ao requisito do RALibras de ser acessível ao maior número possível de dispositivos Android, porque apenas uma restrita lista de *smartphones* de médio e alto desempenho certificados pela Google pode instalar aplicativos com essa tecnologia [Arcore 2020].

Por sua vez, o A-Frame é um software livre com o objetivo de fornecer a experiência de Realidade Virtual (RV) a partir de um navegador, utilizando as tecnologias *Hypertext Markup Language* (HTML) e JavaScript. Ainda que o enfoque inicial tenha sido a RV, seu uso foi estendido com a inclusão da realidade aumentada, através da ‘AR.js’, que é uma biblioteca leve projetada para possibilitar RA por detecção de imagens, por localização do usuário ou por marcadores. Esse *framework* funciona em qualquer dispositivo (seja ele Windows, Android, iPhone) que suporte as tecnologias *WebGL* e *WebRTC*, e propõe bom desempenho do aplicativo mesmo em *smartphones* antigos [Aframe 2017]. A *WebGL* é uma interface de programação de aplicação (*Application Programming Interface* – API) que permite a renderização de objetos 3D a partir da HTML [Webgl 2020]; e a *WebRTC* constitui-se em uma API que torna possível os navegadores e aplicativos móveis estabelecerem comunicação em tempo real para transmissão de vídeo, áudio ou dados [Webrtc 2019]. Assim, após analisar as características dessas tecnologias (ARCore x A-

Frame), decidiu-se por desenvolver o RALibras usando o A-Frame, em conjunto com a biblioteca 'AR.js', a qual atua como provedora da tecnologia de RA.

Dentre as possibilidades disponibilizadas pelo A-Frame, foi escolhida a RA baseada em marcadores impressos, porque proporciona maior estabilidade na exibição do sinal de Libras durante a experiência de aprendizagem. Um marcador consiste, geralmente, em uma figura quadrada, com duas bordas: uma interna, preta e larga; outra externa, branca e mais estreita. Ao apontar a câmera para o sinal, utilizando o RALibras, o aplicativo o reconhecerá e exibirá acima dele, na tela, o conteúdo desejado. O marcador selecionado foi o 'Hiro', pois ele é um dos marcadores padrão da 'AR.js', fornecendo mais segurança sobre a sua capacidade de detecção. Esse marcador é bitonal (preto e branco) para diminuir a interferência da luz e das configurações da câmera durante o uso. A detecção ocorre após o 'AR.js' inferir que há um candidato a marcador na imagem, então os cantos da borda interna (preta) auxiliam na correção da distorção da perspectiva da câmera. O resultado do cálculo é comparado com o marcador Hiro; e o modelo 3D será exibido se o resultado da comparação satisfizer um limite mínimo de confiança [Hirzer 2008].

Ao se selecionar o A-Frame, adicionou-se um nível de complexidade ao *software*, uma vez que ele precisa estar descrito em arquivo *.html* e ser acessado por meio de um navegador para ser utilizado. Por isso, primeiramente, hospedou-se em um servidor externo um arquivo *.html* com o código imprescindível para a experiência em RA. Ele foi acessado pelo aplicativo e obteve sucesso em exibir, como teste, um cubo sobre o marcador. Porém, essa arquitetura de sistema impossibilitaria que o aplicativo funcionasse estando o *smartphone* inteiramente *offline*, um requisito importante para garantir a disponibilidade do conteúdo do RALibras no ambiente da sala de aula. Visando a contornar essa limitação, verificou-se que é possível inicializar (a partir do aplicativo) um servidor interno ao dispositivo por meio do NanoHTTPD [NanoHttpd 2019].

A NanoHTTPD é uma biblioteca Java de código-aberto projetada para implementar um servidor web leve passível de ser embarcado em outras aplicações. Ela foi escolhida porque possui recursos essenciais para o protótipo, tais como: (i) protocolo *HyperText Transfer Protocol Secure* (HTTPS), necessário devido ao uso do WebXR (padrão para renderização de RA, que utiliza WebGL) pela A-Frame – mas, como endereços de *loopback* são considerados seguros pelos navegadores, ele não foi utilizado no aplicativo –; (ii) suporte a requisições do tipo GET, usado para que o servidor receba os dados descrevendo qual sinal em Libras deve exibir; (iii) provimento dinâmico de conteúdo, um meio prático para que o documento *.html* seja criado considerando os parâmetros passados pela requisição GET; (iv) *Cross-Origin Resource Sharing* (CORS) [NanoHttpd 2019], que é indispensável para que o navegador permita que o aplicativo carregue os arquivos para serem renderizados de uma origem distinta daquela que se está acessando – o RALibras viabiliza que os modelos 3D sejam carregados a partir da *cache* do *smartphone*.

Um servidor NanoHTTPD é instanciado assim que o aplicativo inicia, permanecendo conectado e recebendo requisições pelo endereço *localhost*. Quando um sinal em Libras é selecionado pelo usuário, o aplicativo abre uma aba do navegador padrão do dispositivo que fará a requisição GET, desencadeando a criação de um documento *.html* pelo servidor. Esse documento contém os parâmetros necessários para a exibição do modelo 3D correspondente àquele selecionado. O navegador é capaz de exibir a visão da câmera do *smartphone* e iniciar a experiência de RA, quando tiver recebido todos os recursos solicitados.

Na tela de RA e com todos os recursos carregados pelo navegador, assim que o usuário direcionar a câmera para o marcador Hiro, o A-Frame o detectará e exibirá uma mão acima dele na tela – que estará na posição do sinal selecionado. Para a criação do modelo exibido e dos seus movimentos, foi utilizado o *software* gratuito Blender, e o formato de arquivo empregado na extração foi o *gl Transmission Format* (glTF). O glTF é um formato de licença livre que busca diminuir o tamanho em memória de objetos 3D e o processamento necessário para seu uso [Gltf 2020]. Com essas tecnologias, foi possível criar uma “mão virtual” e armazenar o alfabeto em Libras do protótipo em um único arquivo, tornando-o um aplicativo leve, com menos de 12 megabytes. Essas etapas encontram-se esquematizadas na Figura 1, que ilustra, através de um diagrama de seqüências, todo o processo de reconhecimento do sinal de Libras.

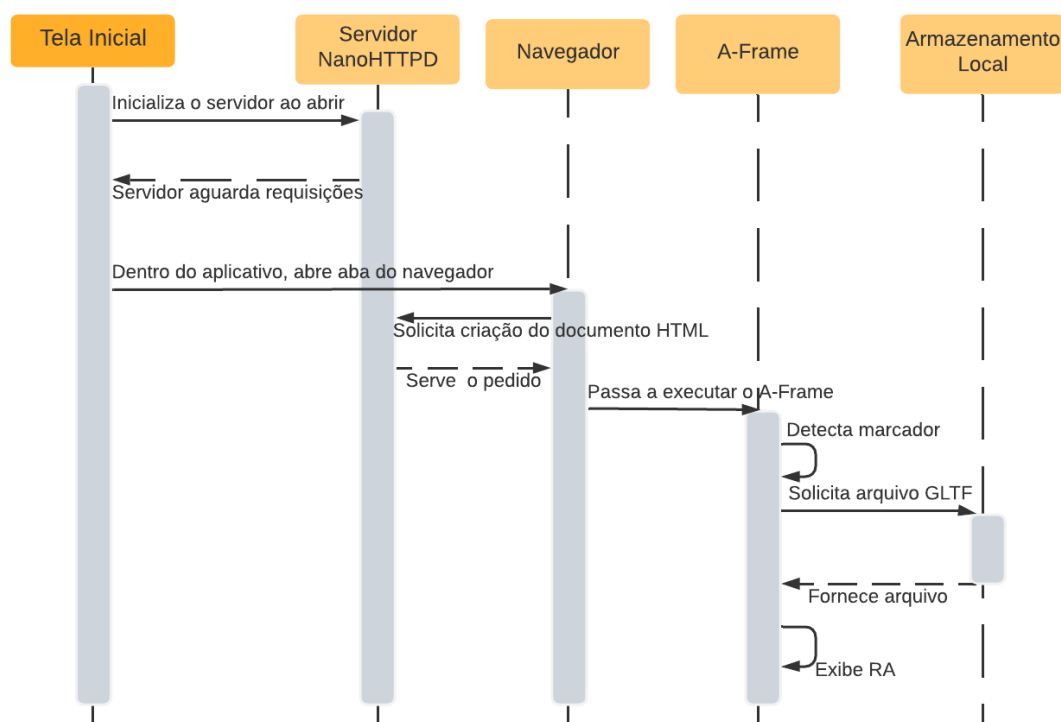


Figura 1. Abstração do processo necessário para exibição da RA no aplicativo

3. Apresentação do *Software*

Conforme mencionado previamente, o RALibras é um aplicativo para o apoio ao ensino de Libras com uso de Realidade Aumentada. Ele encontra-se em desenvolvimento para smartphones que utilizem o sistema operacional Android, desde a API nível 21 (Android versão 5) até a API nível 29 (Android versão 10). O protótipo aqui exposto refere-se à funcionalidade *offline* do sistema, sendo implementado e testado em dispositivos Android nas versões 6 e 7.1. Nas próximas seções encontram-se detalhes da modelagem e algumas das telas da interface gráfica com o usuário do RALibras.

3.1. A Modelagem

O aplicativo foi modelado usando-se a UML (*Unified Modeling Language*), através do diagrama de casos de uso, o qual tem como objetivo modelar o comportamento de um sistema computacional e mapear, de forma visual, seus principais requisitos [Guedes 2018]. No

diagrama ilustrado pela Figura 2, tem-se como ator o usuário, que pode ser um aluno ou um docente, e os casos de uso são: (i) visualizar sinal em Libras por RA, com a possibilidade de ler um conteúdo associado a ele por quem o cadastrou; (ii) imprimir marcador de RA; (iii) adicionar sinal em Libras, sendo que ele pode ser compartilhado com outros usuários.

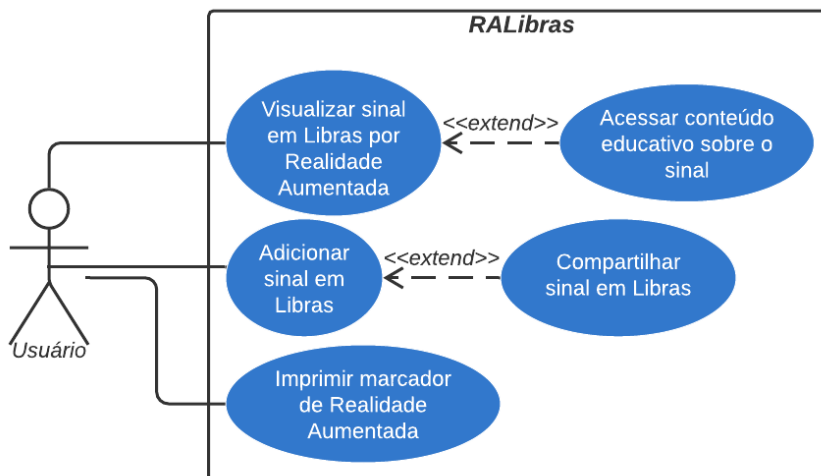


Figura 2. Diagrama de Casos de Uso do RALibras

Destaca-se ainda que, como principal requisito não funcional estabelecido para esse aplicativo, tem-se a possibilidade do seu uso de modo *offline*. Como muitas das salas de aula das escolas brasileiras não possuem acesso à internet, pensou-se em viabilizar o uso do aplicativo mesmo sem acesso à rede. Assim, após realizar o *download*, o usuário tem acesso aos sinais de Libras, mesmo que o seu celular não esteja conectado à internet. Convém mencionar que o objetivo do aplicativo é ser uma ferramenta para complementar a aprendizagem de Libras, podendo ser usado em disciplinas optativas e obrigatórias dos diferentes cursos.

3.2. RALibras: Possibilidades de Interação

Ao abrir o aplicativo, é exibida uma tela listando os sinais disponíveis para a experiência de aprendizagem. No protótipo, optou-se por, inicialmente, disponibilizar as letras do alfabeto em Libras pela sua utilidade – uma vez que torna possível, quando aprendidas, representar a palavra desejada para uma pessoa surda. Outra motivação importante foi demonstrar a capacidade do conjunto de tecnologias utilizadas para exibir modelos 3D em movimento.

O menu principal do RALibras pode ser aberto a partir das telas ‘Início’, ‘Ajuda’ e ‘Sobre’. Sua função é permitir a navegação entre as funcionalidades do aplicativo, por isso a maioria delas pode ser acessada por meio dele. Quando aberto, o menu expandirá da esquerda para a direita, por cima do conteúdo da tela (Figura 3b).

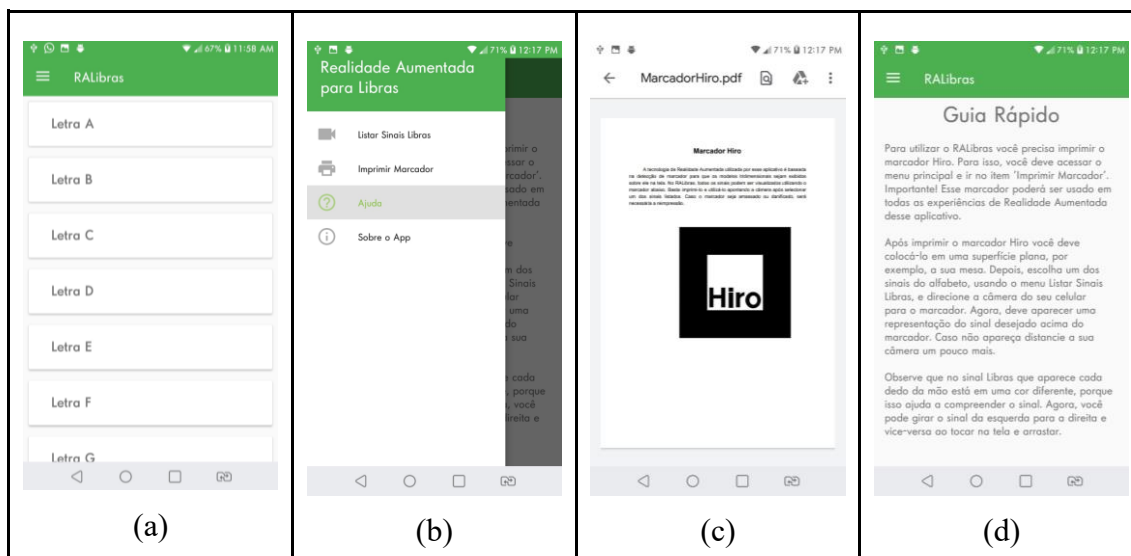


Figura 3. Tela Inicial e Menu Principal do RALibras

Devido às opções realizadas durante o projeto do protótipo, é necessária a impressão do marcador Hiro (Figura 3c) – podendo ser em papel branco comum – para que a experiência de RA aconteça. Facilitando o uso do RALibras, é possível ter acesso (por meio do visualizador de PDF padrão do *smartphone*) a um arquivo PDF contendo o marcador já dimensionado para impressão e instruções sobre o uso dele. Todos os sinais de Libras podem ser visualizados utilizando-o. Para retornar ao aplicativo, basta fechar o visualizador de PDF.

O RALibras possui uma tela dedicada (Figura 3d) à exposição de texto de ajuda ao usuário. Nela estão descritos, resumidamente, os passos necessários para usar o aplicativo, desde a impressão do marcador; passando pela utilização da câmera durante a experiência de RA; até a interação com o sinal Libras exibido na tela.

3. 3. Realidade Aumentada

O usuário pode selecionar uma letra do alfabeto para ver a sua representação em Libras. Quando isso ocorre, o aplicativo inicia uma aba do navegador já com o endereço e parâmetros corretos para requisitar todos os recursos necessários para a funcionalidade. Basta fechar a aba do navegador para retornar à tela inicial do RALibras. A Figura 4 ilustra o uso do RALibras: a Figura 4a possibilita exibir o sinal correspondente à letra “A”, enquanto a Figura 4b esquematiza a letra “G”. É importante notar que, no modelo produzido, cada dedo da mão possui uma cor, pois isso possibilita que o estudante possa identificar como representar o sinal desejado.

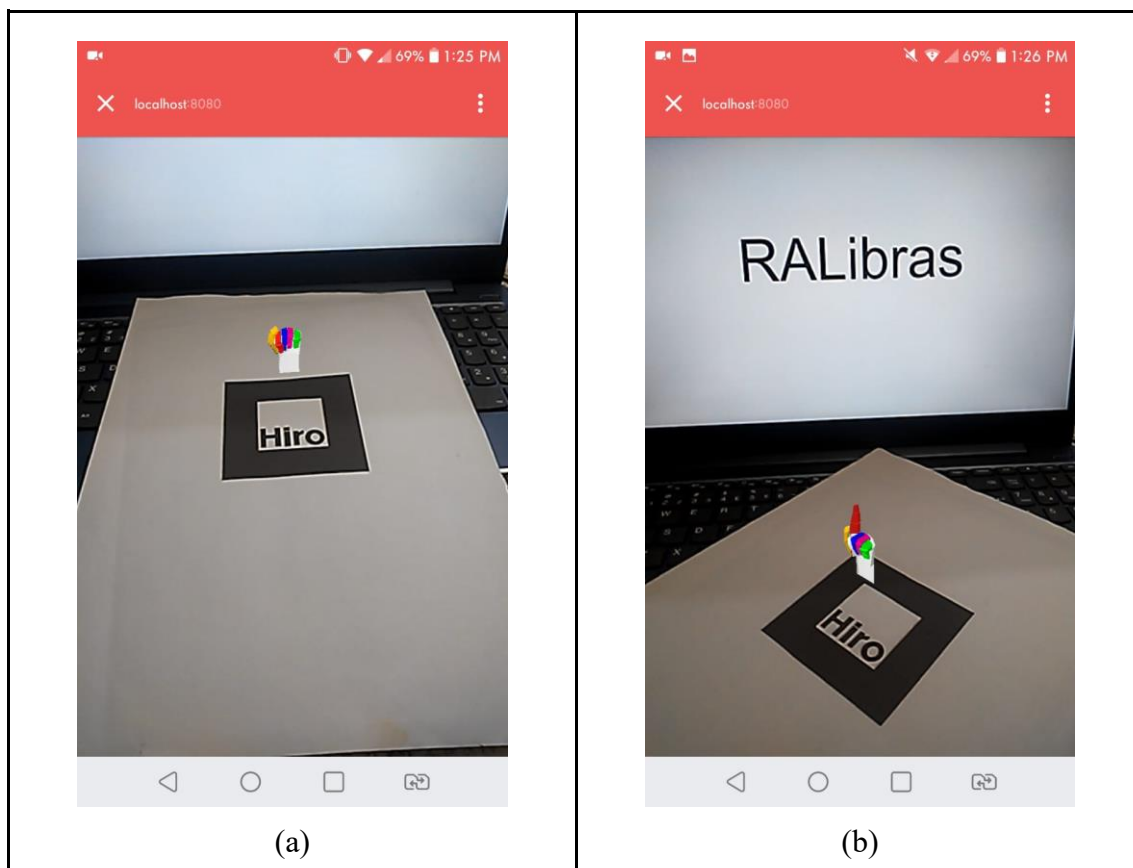


Figura 4. Visualização em RA das Letras A e G no RALibras

Como o navegador Chrome foi escolhido para acessar o arquivo *.html* no protótipo, não foi possível eliminar da aba o endereço acessado e o menu padrão situado no canto direito. Isso se deve porque a Google não permite a remoção dessas informações sem que seja implementada a comprovação de que aplicativo e o domínio que estão sendo acessados pertencem à mesma organização [Trusted 2020].

4. Considerações Finais

De acordo com Lopes (2019), existem barreiras a serem superadas para que a RA seja utilizada com sucesso. O protótipo do RALibras propõe soluções para algumas dessas barreiras: (i) ele foi projetado para ser compatível com o sistema operacional Android a partir da versão 5 (o protótipo foi testado em smartphones nas versões 6 e 7); (ii) o fato de o protótipo não precisar de conexão com a internet e nem de sinal de GPS auxilia na redução do consumo de energia – futuramente, pode ser estudada a viabilidade de aumentar a eficiência do uso da bateria por meio da personalização do A-Frame –; (iii) a experiência de RA no aplicativo pode ocorrer de modo *offline*, conforme implementado no protótipo. A barreira de “necessidade de conexão para acessar objetos virtuais” não ocorrerá após a implementação das funcionalidades *online*, porque o usuário poderá fazer o *download* de sinais em Libras para uso enquanto fora da rede; (iv) com relação a segurança e privacidade, o protótipo não utiliza dados de localização do *smartphone*, e os *softwares* de terceiros utilizados foram embarcados no aplicativo. Quando as funcionalidades *online* do RALibras forem implementadas, serão aplicadas técnicas adequadas à proteção de dados dos usuários.

Considerando o ponto de vista do ensino e da aprendizagem, o propósito do aplicativo é viabilizar que estudantes possam conhecer os sinais de Libras, cabendo ao docente contextualizar onde e como o sinal pode ser usado. Atualmente, o uso da Libras tem se concentrado, basicamente, na educação de nível superior; a intenção do RALibras é possibilitar, também, que estudantes da educação básica possam conhecer os sinais de um modo diferenciado e dinâmico. Pretende-se, futuramente, possibilitar a adição de sinais em Libras pelos usuários e o seu compartilhamento dentro do aplicativo, visando a incentivar a integração e a colaboração entre os usuários durante o uso do *software*.

Referências

- Aframe. (2017) “Creating Augmented Reality with AR.js and A-Frame. A-Frame”, <https://aframe.io/blog/arjs/>.
- Arcore. (2019) “ARCore overview”, Google, <https://developers.google.com/ar/discover>.
- Arcore. (2020) “ARCore supported devices”, Google, <https://developers.google.com/ar/discover/supported-devices>.
- Brasil. (2005) “Decreto nº 5.626, de 22 de dezembro de 2005”, http://www.planalto.gov.br/ccivil_03/_ato2004-2006/2005/decreto/d5626.htm.
- Gltf. (2020) “glTF Overview” - <https://www.khronos.org/gltf/>.
- Guedes, G. T. A. (2018) “UML 2 - uma abordagem prática”, Novatec.
- Hirzer, M. (2008) “Marker Detection for Augmented Reality Applications”, Technical Report, https://www.researchgate.net/publication/321253407_Marker_Detection_for_Augmented_Reality_Applications
- Lopes, L. M. D.; Vidotto, K. N. S.; Pozzebon, E.; Ferenhof, H. A. (2019) “Inovações educacionais com o uso da Realidade Aumentada: uma revisão sistemática”, *Educ. rev.*, v. 35.
- Mec. Ministério da Educação. (2017) “Base Nacional Comum Curricular”, <http://basenacionalcomum.mec.gov.br/wp-content/uploads/2018/02/bncc-20dez-site.pdf>.
- NanoHttpd. (2019) “NanoHTTPD - a tiny web server in Java”, <https://github.com/NanoHttpd/nanohttpd>.
- Webgl. (2020) “WebGL 2.0 Specification”, Khronos Group, <https://www.khronos.org/registry/webgl/specs/latest/2.0/#1>.
- Webrtc. (2019) “Real-time Communication - WebRTC”, <https://webrtc.org/>.
- Trusted. (2020) “Trusted Web Activity - Web on Android”, <https://developers.google.com/web/android/trusted-web-activity>.
- Vuforia. (2018) “Vuforia Engine - Developer Portal”, <https://developer.vuforia.com/pricing>.