

RALibras: Um Aplicativo para Ensino Suportado Por Realidade Aumentada em Situações *Offline*

**Trabalho de Conclusão do Curso de
Tecnologia em Sistemas para Internet**

**Guilherme da Silva Mottin
Orientadora: Silvia de Castro Bertagnolli**

¹Instituto Federal de Educação, Ciência e Tecnologia do Rio Grande do Sul (IFRS)
Campus Porto Alegre
Av. Cel. Vicente, 281, Porto Alegre – RS – Brasil

guilhermedasilvamottin@gmail.com, silvia.bertagnolli@poa.ifrs.edu.br

Resumo. *Este artigo descreve o desenvolvimento do aplicativo RALibras para sistema operacional Android, que possui como propósito o suporte ao ensino da Língua Brasileira de Sinais (Libras) com o uso de Realidade Aumentada (RA). O foco é tornar o software o mais disponível possível e, para isso, ficou estabelecido que as experiências de RA devem ocorrer de maneira online e offline. Esse requisito não funcional exigiu a aplicação de diferentes tecnologias para que ele fosse viabilizado. O público alvo são alunos de todos os níveis que desejarem conhecer uma nova abordagem no estudo de Libras, bem como professores que buscam por novas ferramentas que possam ser utilizadas de forma complementar às atividades de sala de aula.*

1. Introdução

O Censo 2010 do Instituto Brasileiro de Geografia Estatística estimou que cerca de dez milhões de brasileiros são surdos, desses por volta de três milhões não conseguem ouvir (IBGE, 2010). A Língua Brasileira de Sinais (Libras) é a linguagem utilizada pela maioria dos deficientes auditivos.

O Decreto 5.626 de 2005 (BRASIL, 2005) regulamenta o ensino de Libras como sendo obrigatório nos cursos de formação de professores (Pedagogia, Licenciatura, Educação Especial) e cursos de Fonoaudiologia. Esse mesmo ato normativo também estabeleceu Libras como disciplina optativa nos demais cursos superiores e educação profissional. Em reconhecimento à importância da Libras para os surdos e seu impacto na educação, ela foi inserida no rol de competências gerais da Base Nacional Comum Curricular (BNCC):

“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.”(BNCC, 2017, p. 9, grifo nosso)

O presente trabalho descreve o projeto do aplicativo RALibras, que tem como finalidade auxiliar no ensino da Libras em sala de aula, apoiando-se no uso de Realidade Aumentada (RA) ainda que em situações *offline*. Para isso, utiliza-se a capacidade da RA em representar, com melhor efetividade que as mídias tradicionais como o papel, informações no formato tridimensional (3D) e ampliar o envolvimento do estudante durante o aprendizado.

O RALibras tem como público alvo alunos de qualquer nível educacional que possuam interesse em estudar Libras e professores que identifiquem o aplicativo como um ferramenta complementar a outras estratégias pedagógicas. Estão disponíveis funcionalidades *online*, tais como *upload*, *download* de sinais em Libras (e o seu compartilhamento com outros usuários do RALibras), bem como o agrupamento deles em *tags* para posterior pesquisa; e *offline*, como utilizar os modelos 3D armazenados no *smartphone* em experiências de RA.

As próximas seções do artigo apresentam a fundamentação teórica na seção 2, os trabalhos relacionados na seção 3, o desenvolvimento do sistema na seção 4, modelagem e funcionamento do sistema proposto na seção 5, as plataformas *mobile* e *web* desenvolvidas nas seções 6 e 7 e conclusões na seção 8.

2. Fundamentação Teórica

Nos anos 1990, foi publicado um artigo relatando o projeto de um *Heads-Up Display* (HUD) criado pela empresa Boeing com o objetivo de auxiliar os funcionários da complexa linha de montagem de aviões. Para isso, o HUD funcionou como óculos por onde as instruções de montagem de uma ferramenta de Desenho Assistido por Computador (CAD) poderiam ser vistas ou sobrepostas ao ambiente de trabalho (por exemplo, indicação do próximo rebite a ser soldado). Sem essa tecnologia, o funcionário precisava deslocar-se até um terminal que possuísse CAD, memorizar a instrução recebida e retornar ao seu posto - o que poderia ocasionar lentidão e falhas. A inclusão de instruções virtuais ao mundo real motivou o autor a cunhar o termo Realidade Aumentada (RA) (CAUDELL; MIZELL, 1992).

Já em 1997, Azuma (1997) definiu a RA como sendo uma derivação da Realidade Virtual (RV) que não possui o objetivo de substituir o mundo real, e sim complementá-lo com itens virtuais que, idealmente, aparentam pertencê-lo. Em sua pesquisa, Azuma (1997) afirmou que não era preciso ater-se ao uso de HUD em aplicações, optando por relatar outros métodos que também mesclavam o mundo real com objetos virtuais, proporcionavam interação em tempo real e tridimensionalidade. Segundo o autor, a primeira decisão a se tomar ao projetar para RA era considerar qual a natureza do meio que dará suporte para que se mescle o real com o virtual - se óptico (um HUD, por exemplo) ou vídeo (sistema de tela e câmera). Em sua época, o meio existente de alcançar um sistema de realidade aumentada móvel com uso de vídeo consistia no acoplamento de câmeras a robôs que enviavam a imagem capturada e a posição do dispositivo para um computador. Hoje, tem-se dispositivos amplamente disponíveis que permitem utilizar a RA móvel: os *smartphones*.

Atualmente, um pouco mais de 228 milhões de telefones móveis estão registrados nos bancos de dados das concessionárias de telefonia atuantes no Brasil. Isso significa que há quase 1 *smartphone* por habitante, demonstrando assim a facilidade de acesso que a população tem a essa tecnologia (ANATEL, 2020). Sendo que, o sistema operacional Android possui quase 85% da quota de mercado brasileiro de dispositivos móveis (STATCOUNTER, 2020). Logo, ao projetar-se um *software* para uma plataforma específica, optar pelo sistema operacional Android é uma estratégia sólida para alcançar o maior número de pessoas no país.

Embora a densidade de *smartphones* demonstre que há quase um deles por brasileiro, mesmo assumindo que a distribuição seja perfeita e que todos possuam algum dispositivo, permanece ainda a barreira do acesso à *internet*. Um levantamento realizado pelo Instituto de Pesquisa Econômica Aplicada (IPEA) demonstra que 6 milhões de estudantes brasileiros não possuem acesso à rede mundial de computadores em seu domicílio, quer seja por banda larga, 3G ou 4G (IPEA, 2020). Desses, apenas 192 mil estão matriculados no Ensino Superior, indicando que os alunos da Educação Básica estão notoriamente mais sujeitos a não conseguir acessar recursos *online*. Outro estudo feito pelo IPEA indica que 73% dos estabelecimentos de ensino fundamental e médio apresentam conexão com a *internet* (IPEA, 2020). Os estudantes do Ensino Básico que não possuem acesso em seu domicílio têm a chance de encontrar em suas escolas oportunidade de acesso à web, mas dependem de recurso para estudo enquanto *offline*.

Pesquisas sobre a aplicação da tecnologia de RA como ferramenta de ensino têm sido realizadas em todos os níveis do sistema de ensino. Na educação infantil, Bassani (2019) utilizou o *software QuiverVision* para desenvolver atividades de colorir com crianças e concluiu que a tecnologia promoveu a interação e o desenvolvimento cognitivo. No nível do Ensino Básico, Cerqueira (2019) desenvolveu o jogo educacional *FootMath* para ensino de funções matemáticas e validou-o com professores das áreas da matemática e da informática, que retornaram um feedback positivo por ele ser uma abordagem adicional com potencial de incentivar a participação dos alunos. No Ensino Superior, Pereira e colaboradores (2019) desenvolveram um *software* para auxiliar o ensino de cálculo em cursos de graduação, o 'EducAR - Quadrics'. Esse aplicativo foi testado por alunos que acreditam que seu uso facilita a apropriação do conteúdo durante o estudo. A tecnologia de RA é uma ferramenta que pode complementar estratégias de ensino, sendo muito atrativa para os alunos e eficiente por representar conteúdos complexos.

3. Trabalhos Relacionados

É possível encontrar alguns *softwares* em trabalhos acadêmicos que objetivam suporte ao ensino de Libras por meio da RA. No contexto deste trabalho, foram selecionados aqueles que mais se aproximam das funcionalidades do sistema desenvolvido.

3.1. Ensino de Libras usando Realidade Aumentada (ELRA)

O *software* ELRA (NAZARETH *et al.*, 2014), foi projetado como uma ferramenta de auxílio ao ensino do alfabeto de Libras para crianças Portadoras de Deficiências Auditivas (PDAs). Os recursos necessários para uso do sistema são um computador, um monitor e uma câmera - além de marcadores impressos para uso com Realidade Aumentada (RA). Para utilizar o sistema, a criança escolhe um marcador, mostra-o para a câmera, e então o ELRA interpreta a imagem da câmera para exibir na tela um modelo 3D de uma mão na posição do sinal correspondente.

No que diz respeito ao método de funcionamento (utilização de marcadores de RA pelos usuários) e ao tema (apoio ao ensino de LIBRAS), o RALibras é semelhante ao ELRA. Os principais diferenciais do RALibras comparado ao ELRA é que ele foi

desenvolvido para dispositivos móveis e possui suporte à animação 3D (3-Dimensões) - o que torna possível a representação de sinais mais elaborados e não apenas os estáticos. Outro aspecto é que o RALibras disponibiliza a funcionalidade de inclusão de conteúdo pelo usuário, ou seja, além dos sinais vigentes novos sinais poderão ser incluídos na sua base de dados.

3.2. Software para aprendizagem de LIBRAS utilizando realidade aumentada

No trabalho de Correia et. al. (2018) é detalhado um aplicativo para ensino de LIBRAS com RA, o qual é direcionado para a plataforma Android. No trabalho o autor relata que o desenvolvimento encontrava-se no estágio inicial: implementando a tradução da imagem do marcador de RA (captada pela câmera de um celular) para um modelo 3D na tela do dispositivo. As tecnologias utilizadas para tornar isso possível foram o *AR Core*, que possibilitou utilizá-lo em sistemas operacionais Android e iOS; e a *Unity3D* um motor gráfico, normalmente utilizado para o desenvolvimento de jogos, que permitiu a modelagem 3D e animação dos sinais em Libras.

O aplicativo implementado possui características comuns ao aplicativo de Correia et. al. (2018), visto que ambos podem ser usados em dispositivos Android e objetivam auxiliar o ensino de LIBRAS por meio da realidade aumentada. Entretanto, o RALibras provê a organização dos sinais por meio de *tags* e a possibilidade de inclusão de conteúdo pelos usuários.

3.3. Língua Portuguesa/Libras R.A.

No estudo elaborado por Carvalho et. al. (2017) avaliou-se a efetividade de um programa de ensino de LIBRAS implementado por meio do *software* Língua Portuguesa/Libras R.A. Nesse sistema, descrito por Carvalho (2011), o aprendizado é feito através do estabelecimento da relação entre a palavra em português, o respectivo sinal em LIBRAS e uma imagem representativa. Para cada programa de ensino (conjunto de sinais a serem aprendidos em uma aula), é possível cadastrar até quinze trios de palavra/sinal/imagem - uma vez que os marcadores de RA são pré-determinados. Outra funcionalidade é a edição e exclusão de trios já cadastrados.

O RALibras, assim como o sistema descrito no parágrafo anterior, é fundamentado na associação de palavras e sinais. Porém, ele é capaz de, com um único marcador, exibir qualquer sinal disponível no banco de dados e, por isso, é mais versátil. Ainda, o aplicativo possibilita compartilhamento do conteúdo criado pelos usuários.

3.4. Estudo Comparativo dos Trabalhos

A presente seção estabelece um comparativo das principais funcionalidades dos aplicativos analisados e o RALibras. Para tanto, foi elaborado o Quadro 1 que lista as características analisadas e compara cada uma delas com alguns dos trabalhos estudados e o trabalho proposto. O *software* de Nazareth (2014) não foi incluído por não possuir as funcionalidades relacionadas.

Quadro 1 - Comparação entre características o RALibras e sistemas similares

Características	Correia (2018)	Carvalho (2011)	RALibras
1. Uso por meio de dispositivo móvel	X		X
2.Criação, Busca, Atualização e Exclusão de sinais em Libras		X	X
3.Uso de <i>tags</i> para criar agrupamentos de sinais		X	X
5.Possibilidade de visualização do sinal usando animação 3D	X	X	X
6. Imagem complementar durante a RA			X
7.Publicação de sinais			X
8.Uso da RA em modo <i>online</i>			X
9.Uso da RA em modo <i>offline</i>			X

Fonte: do Autor, 2021.

4. Desenvolvimento do Sistema

Durante o desenvolvimento do sistema foram tomadas decisões em favor das tecnologias classificadas como *software* livre para remover a necessidade de pagamento de licenças para tornar o RALibras viável. Essas escolhas fizeram com que soluções para alguns desafios tivessem que ser encontradas.

4.1 Implementação da Plataforma *Mobile*

Tornar a RA disponível em dispositivos Android, a partir da versão 5, enquanto *offline* exigiu que fossem adotados métodos não convencionais que abrangem desde o armazenamento até a exibição do modelo 3D na tela do *smartphone*.

4.1.1 Implementação da Realidade Aumentada *Offline* para Android

O estudo sobre as tecnologias necessárias para viabilizar o RALibras caracterizou-se pela pesquisa por *frameworks* de criação de experiências de RA para dispositivos Android. Alternativas consolidadas no mercado e providas por empresas de renome foram analisadas, porém desconsideradas por não serem totalmente gratuitas. O Vuforia, por exemplo, oferece a possibilidade de desenvolver a RA utilizando sua tecnologia, entretanto exige a aquisição de uma das suas modalidades de licença (VUFORIA, 2018). Ao excluir os *frameworks* pagos, restaram o ARCore (desenvolvido pela Google) e o A-Frame (*software* livre).

O ARCore disponibiliza recursos para a implementação de experiências em RA baseadas no reconhecimento do ambiente (por exemplo, geolocalização e reconhecimento de superfícies); ou que necessitem oferecer compartilhamento entre usuários (ARCORE, 2019). Ainda que essas funcionalidades tenham grande potencial, o aplicativo produzido com o *framework* é acessível apenas por dispositivos certificados pela Google (ARCORE, 2020). A relação de *smartphones* aptos é composta por modelos dos segmentos de mercado médio e alto padrão, caracterizando elevado custo

de aquisição - o que desqualifica o ARCore por não atender a necessidade do RALibras de atender ao maior número possível de dispositivos.

O *software* livre A-Frame provê experiências de RV, com base nas tecnologias *Hypertext Markup Language* (HTML) e JavaScript (JS), acessadas por meio de um navegador. Ainda que a RV seja seu objetivo principal é possível desenvolver RA por meio da biblioteca 'AR.js' contida nele, oferecendo RA baseada em marcadores, ou reconhecimento de imagem ou geolocalização. As aplicações que o utilizam funcionam nos sistemas operacionais Windows, Android e iPhone, sendo o principal requisito o suporte às tecnologias WebGL e WebRTC, e têm bom desempenho mesmo em *smartphones* antigos (AFRAME, 2017). A WebGL é uma interface de programação de aplicação (*Application Programming Interface* - API) para renderização de modelos 3D com uso de HTML (WEBGL, 2020) e a WebRTC uma API para comunicação em tempo real para transmissão de informação adotada por navegadores e aplicativos móveis (WEBRTC, 2019). Após o estudo sobre as características desses *frameworks* determinou-se o uso do A-Frame na implementação do RALibras.

Escolheu-se para o desenvolvimento do sistema a RA baseada em marcadores impressos, com intuito de obter maior consistência durante a exibição dos objetos 3D. O marcador a ser utilizado é o Hiro (Apêndice A - Figura A.1), um dos modelos oferecidos pela 'AR.js' garantindo a capacidade do aplicativo detectá-lo facilmente. Ele é bitonal, em preto e branco, para tornar menos provável que as configurações da câmera ou a luz do ambiente interfiram no uso. Para que 'AR.js' detecte o Hiro, ele interpreta a imagem captada pela câmera e, caso suponha que há um candidato a marcador na imagem, utiliza os cantos da borda interna (preta) para gerar uma imagem do marcador captado com a distorção causada pela perspectiva da câmera corrigida. Essa imagem é comparada com o Hiro e o modelo 3D é exibido se o limite mínimo de confiança for contemplado (HIRZER, 2008).

Para que funcione, o A-Frame necessita que um arquivo *html* descrevendo o funcionamento da RA seja acessado por meio de um navegador. A primeira opção estudada foi hospedá-lo em um servidor externo, porém isso impossibilitaria o uso da RA estando o dispositivo *offline* - um requisito importante para viabilizar o RALibras em sala de aula. Visando 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 livre que provê servidores web para que sejam embarcados em aplicações Java. Possui requisitos que o A-Frame exige do servidor, tais como *HyperText Transfer Protocol Secure* (HTTPS) e *cross-origin resource sharing* (CORS, indispensável porque o RALibras carregará os modelos 3D diretamente da *cache* do *smartphone* e precisará desse mecanismo para que o navegador permita a operação). Oferece também suporte a requisições do tipo GET (que são utilizadas para informar qual modelo 3D foi solicitado) e provimento dinâmico de conteúdo (meio para que o documento *html* descrevendo a RA seja gerado de acordo com a requisição GET recebida) (NANOHTTPD, 2019).

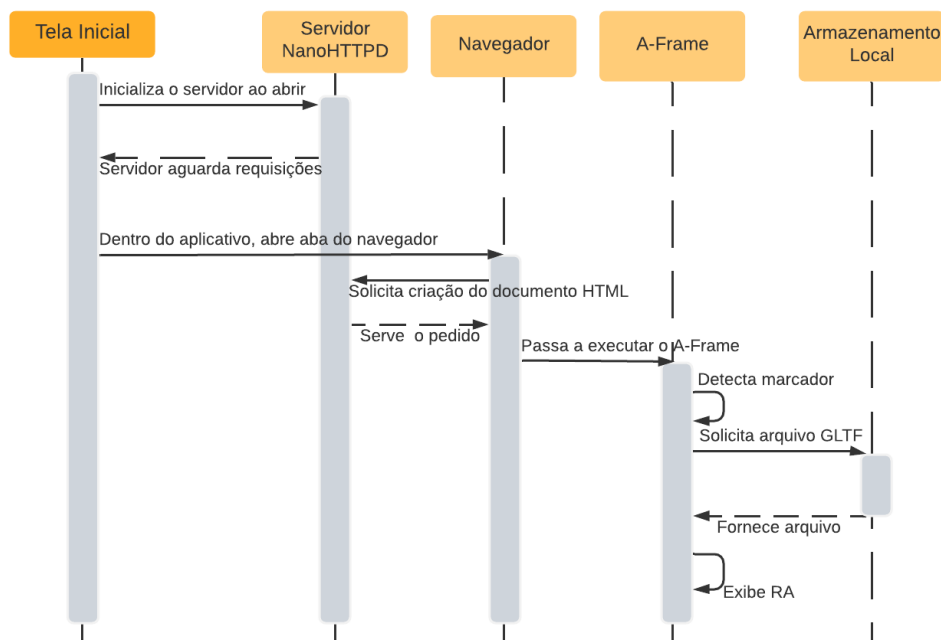
Quando o usuário abrir o RALibras, um servidor NanoHTTPD é instanciado e permanecerá escutando as requisições enviadas ao endereço *localhost*. Uma aba do navegador padrão do dispositivo é aberta após o usuário selecionar um sinal para visualização em RA e, então, fará uma requisição GET ao servidor, que responderá ao

navegador um arquivo html. Esse documento descreve ao A-Frame os recursos necessários para exibição do modelo 3D correspondente ao sinal selecionado. Por fim, o navegador acessa a câmera do *smartphone* e inicia a experiência de RA.

Durante a RA, com todos os recursos carregados pelo navegador, o usuário deverá apontar a câmera do dispositivo para o marcador. A seguir, o ‘AR.js’ processará a imagem e detectará o Hiro, renderizando um objeto 3D representando o sinal em Libras escolhido logo acima dele. O *software* gratuito Blender foi usado para criar o modelo 3D representando uma mão virtual e programar os seus movimentos. Arquivos binários *glb*, do formato *GL Transmission Format* (glTF), foram escolhidos para armazenar os modelos a serem utilizados no RALibras pela capacidade em diminuir o tamanho em memória e o processamento necessário ao usar (GLTF, 2020).

Com as tecnologias listadas criou-se uma “mão virtual”, para disponibilizar o alfabeto em Libras junto com a instalação e oferecimento do uso de RA em situações *offline*. Todo esse processo encontra-se esquematizado na Figura 1, que ilustra, através de um diagrama de sequência, todo o processo de reconhecimento do sinal de Libras.

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



Fonte: Mottin e Bertagnolli (2020).

4.1.2 Persistência de Dados

Para que a A-Frame proveja a RA com o *smartphone* estando *offline*, é necessário que sejam armazenadas as informações a serem exibidas: o nome do sinal, o nome da animação a ser reproduzida, o arquivo que contém o modelo 3D, o arquivo da imagem que será vista dentro da RA (e referências para eles).

Por já vir com a instalação do aplicativo e ser um conjunto que não será expandido, o alfabeto em Libras possui essas informações já codificadas em uma lista que é inicializada quando da abertura da funcionalidade “Alfabeto Libras”. O modelo 3D está descrito no arquivo “*sinaisAlfabeto.glb*” e é armazenado como um *asset* do RALibras, não podendo ser excluído. Contudo, mesmo que sejam representados com ele 29 sinais, ele ocupa um pouco mais de 2MB de espaço de armazenamento.

A funcionalidade de *download* de sinais exige que os dados sejam persistidos no *smartphone* de um modo diferente. O modelo 3D e a imagem (a ser exibida durante a RA) são escritos no espaço cache de memória secundária reservado ao aplicativo. As referências a esses arquivos e os demais dados são guardados utilizando a biblioteca Android de persistência de dados *Room*, que os organiza em uma tabela (Apêndice B - Figura B.1) por meio de *Server Query Language* (SQL). O usuário poderá remover sinais adicionados ao *smartphone*, o que apagará os dados persistidos na tabela e excluirá os arquivos relacionados (do modelo e da imagem) liberando, assim, espaço em memória.

4.2 Implementação da Plataforma Web

De acordo com o requisito de utilizar *software* livre, foi evitada a plataforma facilitadora paga mais usual, *Google Firebase*, na infraestrutura remota. Como alternativa, hospedou-se em um servidor externo um sistema composto por um banco de dados SQL, pelo *back-end* e pelo *front-end*.

Para a persistência dos dados foi selecionado o Sistema Gerenciador de Banco de Dados (SGBD) *MariaDB Server* por ser *software* livre e um dos SGBD mais utilizados atualmente (MARIADB, 2021). Nele são mantidos os dados sobre os usuários (apenas o mínimo necessário para autenticação), as *tags* e os sinais em Libras. No Apêndice B (Figura B.2) é possível encontrar o esquema detalhando as tabelas que foram criadas para a plataforma *Web*.

O Laravel foi escolhido pela produtividade que provê ao implementar uma aplicação, oferecendo: funcionalidades que são comuns em sistemas *Web*; *login* e controle de sessão; controle das dependências do projeto; suporte a REST API (*Application Programming Interface*); e o *Eloquent ORM* (*Object-Relational Mapping*) - uma ferramenta que utiliza o padrão de projeto *Active Record* para realizar o mapeamento entre a classe descrita em *php* e os dados armazenados no banco. A interface principal das operações sobre sinais do tipo *create*, *retrieve*, *update*, *delete* (CRUD) é a que faz maior uso do Laravel e foi projetada para ser usada em computadores, devido a melhor usabilidade para o preenchimento de formulários, especialmente aqueles que exigem *upload* de arquivos.

A comunicação entre as Plataformas *Web* e do *App* foi realizada através da *Retrofit*, visto que ela viabiliza a comunicação do aplicativo *mobile* com o *software web*. Essa biblioteca Java provê comunicação HTTP e o mapeamento da resposta de requisições para objetos (RETROFIT, 2021). Utilizando-a, serviços foram implementados para iniciarem requisições HTTP destinadas à API REST implementada no *back-end web*. Ao receber uma requisição, o servidor realiza o processamento e encaminha a resposta - que será mapeada pelo *Retrofit* para o objeto esperado, possibilitando a sequência do processamento.

A autenticação do usuário foi desenvolvida com a combinação dessa tecnologia com a *Laravel Sanctum*, um pacote destinado a autenticação de *Single Page Applications* (SPA) e aplicativos móveis. Uma requisição POST é feita ao enviar o formulário de *login* para a *url* de autenticação contida na plataforma *web*, que verificará os dados e responderá com um *token*. O aplicativo persiste como um *cookie* local, que manterá o usuário autenticado, ainda que ele feche o RALibras, ao verificar se os *tokens* remoto e local são idênticos. A única funcionalidade da plataforma móvel que exige o

login é a que permite ao usuário consultar os sinais que cadastrou no sistema, a segurança dessa função é garantida pelo *Laravel Sanctum* que verifica o *token* quando ela é acessada. Ao efetuar o *logout*, os *tokens* remoto e local são destruídos.

5. Modelagem e Funcionamento do Sistema Proposto

Com a subdivisão do RALibras em plataforma *web* e plataforma *mobile*, o sistema foi implementado de acordo com as funcionalidades expostas no Quadro 1 - originando dois modelos diferentes.

5.1 RALibras: plataforma *mobile*

O RALibras foi desenvolvido visando *smartphones* que possuam o sistema operacional Android (escolhido porque é o mais propagado no Brasil, com potencial para chegar ao maior número de dispositivos móveis possível) a partir da versão 5 e provê suporte ao ensino de Libras por meio da tecnologia de RA (*online* e *offline*). O aplicativo permite que o usuário imprima o marcador e visualize sinais já armazenados ainda que o dispositivo encontra-se *offline*, possibilitando o uso em ambientes onde o acesso à *internet* é precário ou inexistente. Entretanto, a conexão com a *web* será necessária para realizar: *downloads*; buscas por sinais publicados por todos os usuários; busca dos sinais pelas suas *tags*; busca por sinal por meio do seu código (alfanumérico de 6 caracteres, gerado aleatoriamente quando o sinal é cadastrado); e visualização dos sinais cadastrados pelo usuário. Visando melhorar a experiência dos diversos perfis de usuário que se busca atender, a autenticação do usuário é necessária apenas para acessar a listagem dos sinais que ele cadastrou. As funcionalidades da plataforma *mobile* encontram-se ilustradas no Apêndice C - Figura C.1.

Com a instalação do RALibras é disponibilizado por padrão, dentro do aplicativo, o alfabeto em Libras para que o usuário aprenda a “soletrar” palavras enquanto não realizar o *download* de novos sinais em Libras para o dispositivo. Antes de iniciar a experiência de RA, é necessário imprimir o marcador Hiro para que o sistema utilize-o na orientação da projeção do modelo 3D a ser exibido. De posse do marcador, basta posicioná-lo em uma superfície plana e escolher uma palavra para visualizar, com a RA sobreposta ao marcador (Apêndice A - Figura A.2), a sua representação em Libras. Na tela em que esse sinal é visualizado, é exposta (se houver) a imagem associada ao sinal.

5.2 RALibras: plataforma *web*

A plataforma *web* do RALibras tem como principal objetivo oferecer as funcionalidades CRUD para sinais Libras. O cadastro de um sinal é realizado por meio do *upload* de arquivo contendo o modelo 3D dele (no formato *glb*) ao banco de dados remoto do sistema. Ao cadastrar, o usuário obrigatoriamente define: o nome do sinal; se o gesto é privado ou de acesso público (possibilitando que outros usuários do RALibras visualizem-no na busca por sinais publicados e por *tags*, permitindo o *download* se quiserem mantê-lo em seus dispositivos); a imagem a ser exibida durante a RA (que estar no formato *png* e *jpg*); pelo menos uma *tag* que o categorize (ou por escolha dentre as existentes no sistema, ou pela criação de uma nova).

Além da opção de tornar público um sinal, outra opção de disponibilizá-lo é por

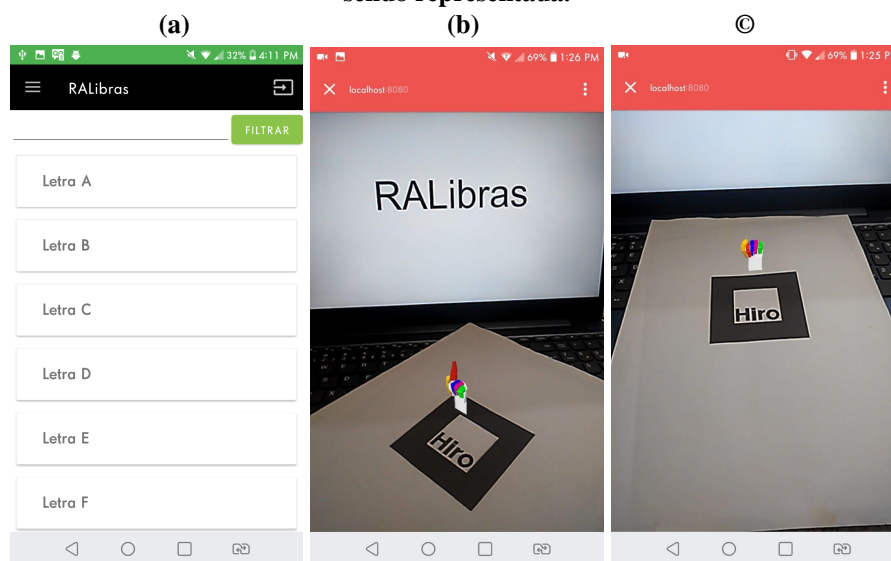
meio do seu código de compartilhamento alfanumérico, que pode ser informado para outro usuário, que buscará por ele através do aplicativo no *smartphone*. Esse código é atribuído ao concluir o cadastro de um novo sinal e é composto por 6 caracteres alfanuméricos aleatórios, sendo feita uma verificação para evitar colisões com códigos já existentes no RALibras. Os sinais que não foram transformados em públicos podem ser distribuídos por meio dessa funcionalidade.

Essas funções e a disponibilização do marcador para impressão pretendem fornecer adaptabilidade ao sistema ao viabilizar que um professor organize e personalize o conteúdo a fim de melhor desenvolver as suas atividades de aprendizagem com seus estudantes. As funcionalidades da plataforma *web* encontram-se ilustradas no Apêndice C - Figura C.2.

6 A Plataforma *Mobile*

O aplicativo para Android do RALibras permite, assim que instalado, a visualização de todo o alfabeto em Libras por meio da RA (Figura 2). Isso é feito utilizando o arquivo *glb* contido no instalador do *software*, disponibilizando o alfabeto em Libras por padrão.

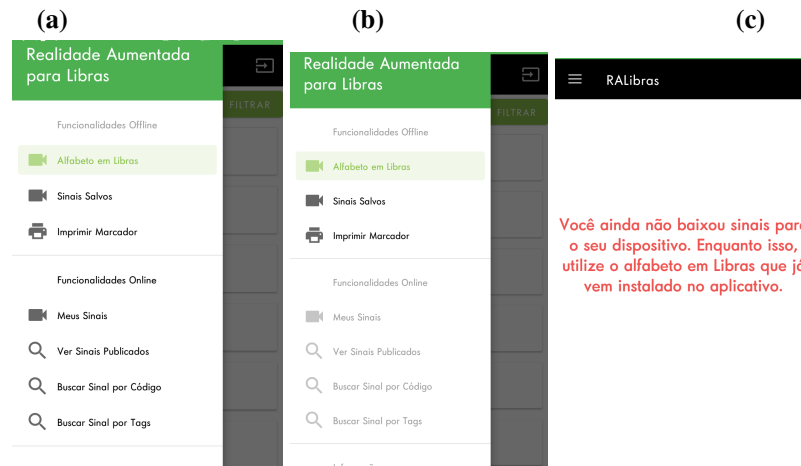
Figura 2 - (a) Tela de exibição dos sinais contidos na instalação do RALibras ;(b) Tela de Realidade Aumentada com a letra ‘G’ sendo representada; (c) Tela de Realidade Aumentada com a letra ‘A’ sendo representada.



Fonte: do Autor, 2021.

Se houver conexão com a *internet* disponível, as funcionalidades *online* poderão ser acessadas para que o usuário busque por mais conteúdos e expanda a quantidade de sinais disponíveis no dispositivo para uso de RA *offline* (Figura 3).

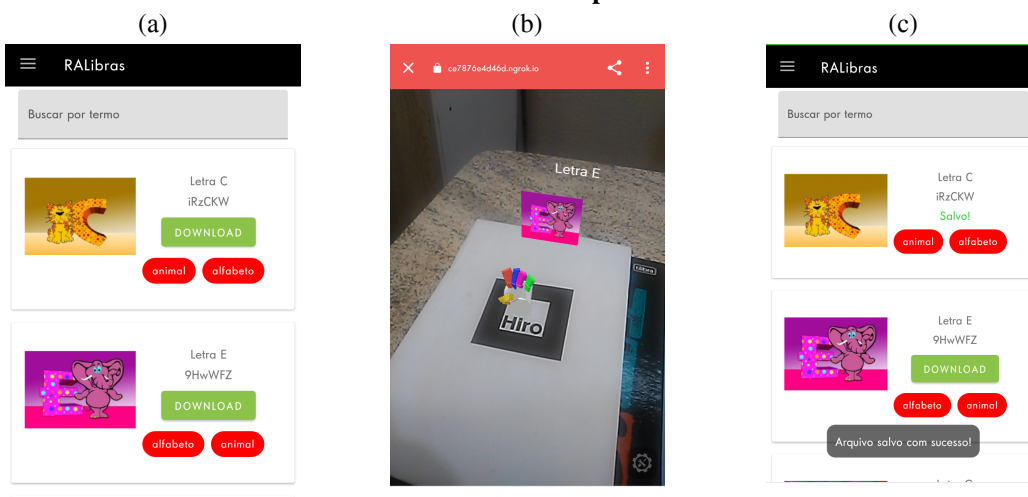
Figura 3 - (a) Configuração do menu com o dispositivo estando *online*; (b) Configuração do menu com o dispositivo estando *offline*; (c) Tela de exibição dos sinais armazenados no dispositivo quando não foi salvo conteúdo adicional.



Fonte: do Autor, 2021.

Os sinais publicados no sistema são listados no formato de *cards*, exibindo a imagem que aparecerá durante a RA, seu nome, seu código de compartilhamento e o botão para *download* (Figura 4a). É possível verificar a experiência de RA antes de salvar o sinal: ao tocar no *card*, o RALibras abre uma aba do navegador e solicita para a plataforma *web* que encaminhe o *html* correspondente e, posteriormente, os arquivos da imagem e modelo 3D. Com todos os recursos carregados, o A-Frame inicia a captura de imagens da câmera que deve estar apontada para o marcador Hiro (Figura 4b).

Figura 4 - (a) Listagem de todos os sinais publicados no sistema; (b) Visualização da RA *online* sem que se armazene conteúdo no *smartphone* (c) Notificação no *card* de que o sinal já está armazenado no dispositivo



Fonte: do Autor, 2021.

A plataforma *mobile* possui três tipos de busca: por *tags*, por código de compartilhamento e por termo. A primeira exibe ao usuário todas as *tags* cadastradas para que ele selecione pelo menos uma para que sirva como parâmetro de busca (Figura 5a). A segunda retorna ao usuário o sinal de acordo com o código de compartilhamento informado, mesmo que ele não esteja marcado como publicado na plataforma *web*

(Figura 5b). A terceira está presente nas listagens do alfabeto, dos sinais salvos no dispositivo e de todos os sinais publicados, ela funciona usando os nomes descritos nos *cards* (Figura 5c).

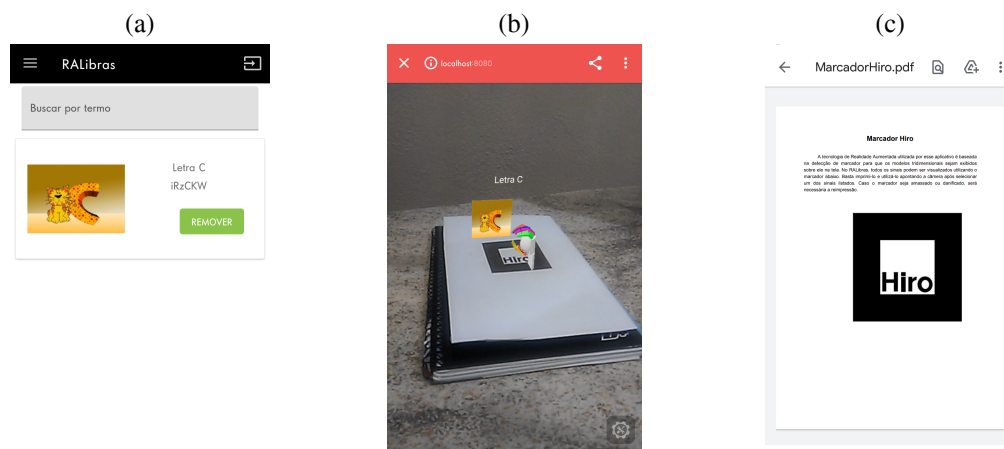
Figura 5 - (a) Exemplo de busca por tags; (b) Exemplo de busca por código de compartilhamento (c) Exemplo de filtragem pelo nome do sinal na listagem de sinais publicados.



Fonte: do Autor, 2021.

A funcionalidade ‘Sinais salvos’, presente no menu, exhibe o conteúdo adquirido por *download* da plataforma *web* (Figura 6a). Os sinais listados nessa tela possuem todos os recursos necessários (arquivos contendo imagem, modelo 3D e dados armazenados no banco de dados local) para RA *offline* disponíveis para uso. A RA (tanto *offline* quanto *online*) dependem do uso conjunto com o marcador Hiro (Figura 6b), que pode ser impresso pelo aplicativo *mobile* ou pela plataforma *web* (Figura 6c). Esse marcador já está dimensionado em um tamanho adequado e é disponibilizado para que o usuário não se preocupe com esse ajuste ao imprimi-lo.

Figura 6 - (a) Tela contendo os sinais adquiridos por *download*; (b) RA *offline* de sinal armazenado no *smartphone*; (c) Documento pdf que contém o marcador Hiro pronto para impressão.

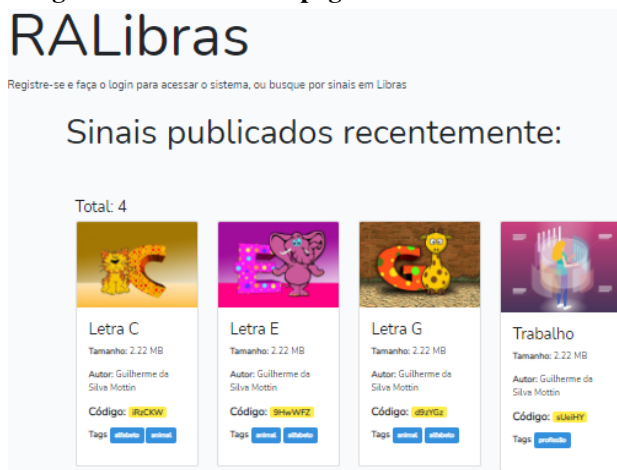


Fonte: do Autor, 2021.

7 A Plataforma Web

Ao acessar o RALibras o usuário será recepcionado pela página inicial que exibe os quatro últimos sinais publicados, assim como orientações sobre as ações possíveis: *login* e busca na base de dados (Figura 7).

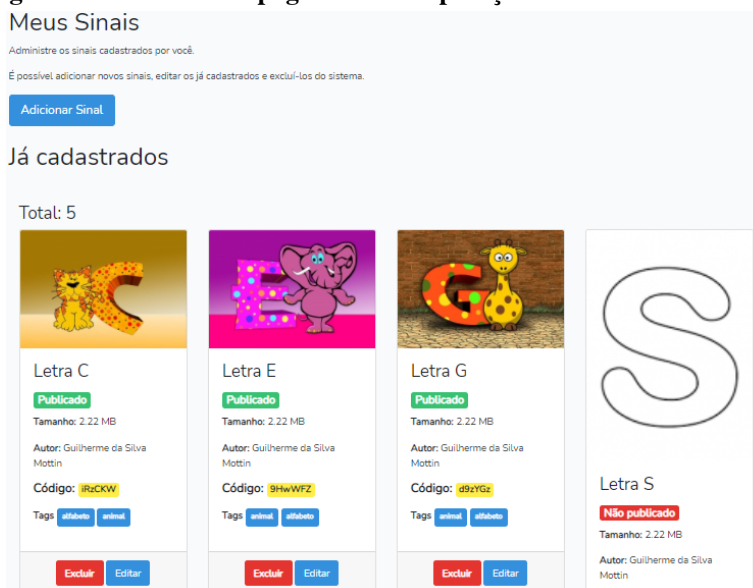
Figura 7 - Conteúdo da página inicial do RALibras



Fonte: do Autor, 2021.

O usuário ganha acesso a novas funcionalidades após autenticar-se no sistema, que permitem visualizar o conteúdo cadastrado por ele mesmo, assim como administrá-lo. Além disso, viabiliza a busca por outros sinais compartilhados pelos demais usuários. Ao clicar em 'Meus Sinais', será exibida a tela que permite acessar e modificar informações relativas aos sinais de Libras que estão disponíveis (Figura 8).

Figura 8 - Conteúdo da página com as operações CRUD sobre os sinais



Fonte: do Autor, 2021.

O cadastro de sinais é realizado em duas etapas: uma para informar dados

básicos e fazer *upload* do arquivo da imagem a ser exibida nas buscas (e dentro da RA) e do arquivo do modelo 3D; e, outra, para confirmar se o sinal é exibido corretamente e informar qual animação contida no arquivo *glb* deve ser associada ao sinal (Figura 9) e quais *tags* serão associadas. Essa divisão em etapas é necessária, porque o *framework A-frame* é utilizado para renderizar a prévia do sinal a ser cadastrado. O *A-frame* deve ter acesso ao arquivo e, por isso, o *upload* do *glb* para o servidor é o ponto de transição para um estado em que é possível carregar os dados do arquivo para prover uma visualização de seu modelo 3D.

Figura 9 - Segunda etapa do cadastro, após *upload* do arquivo *glb*, onde o usuário confere se a exibição é a esperada e seleciona uma das animações do arquivo



Fonte: do Autor, 2021

A funcionalidade de edição de sinal conduz o usuário a um formulário similar ao utilizado na segunda etapa do cadastro, já preenchido com a informação encontrada no banco de dados. A funcionalidade de exclusão remove o sinal e os arquivos dele do RALibras, porém ele permanecerá disponível nos *smartphones* em que já estiver salvo.

A funcionalidade de busca está disponível para qualquer usuário, independente de autenticação. Ao clicar na barra de menu em 'Buscar Sinal', o navegador será redirecionado para a tela em que é possível realizar pesquisas sobre os sinais armazenados no sistema (que inicia exibindo todos aqueles já publicados). É possível realizar a busca por termos (a serem procurados no nome dos sinais) ou por *tags*, bastando selecionar a respectiva aba para isso.

8. Conclusões

Apesar da RA ser um conceito com quase trinta anos de existência, na última década intensificou-se sua aplicação na educação. Os avanços recentes na tecnologia de dispositivos móveis e sua difusão entre a população brasileira criaram um potencial para que a RA evolua e impacte positivamente a área da Educação. Porém, para que o uso da RA ocorra de forma isonômica, as aplicações devem incluir os estudantes e instituições de ensino que não possuem acesso à *internet*; e visarem *smartphones* de baixo custo.

Isso pode ser feito ao tornar a funcionalidade principal do sistema disponível *offline* e ao considerar, durante o projeto, as restrições tecnológicas dos dispositivos móveis mais antigos. As técnicas adotadas e funcionalidades desenvolvidas no RALibras foram consequência da escolha de atender da maneira mais ampla possível não só o público alvo da plataforma mobile, mas também possibilitar que a plataforma

web seja hospedada por instituições sem custos extras advindos de tecnologia de terceiros que exigem licenciamento.

Devido ao curto espaço de tempo para desenvolvimento, não foi possível adicionar algumas ideias que surgiram ao longo da execução deste trabalho. Uma delas consiste em aumentar o acesso ao aplicativo em situações *offline* (tanto ao instalador quanto aos sinais), possibilitando o compartilhamento de sinais e do *Android Package* (APK), que instala o RALibras, por *bluetooth*. Outra funcionalidade identificada compreende a possibilidade de captura do movimento das mãos do usuário, para posterior aplicação dele a um modelo 3D, facilitando a criação de sinais. A inclusão dos perfis professor e aluno seria interessante, pois com isso seria possível implementar funcionalidades específicas, como, por exemplo, a inclusão de lições que um docente pode propor para uma turma ou para um aluno específico. Por fim, acredita-se que diversas áreas da educação se beneficiam do uso da RA. Assim, adaptar o RALibras para que se torne uma plataforma genérica que não contemple apenas Libras é algo possível, tendo em vista o modo como ele foi implementado.

Referências

AFRAME. **ARCore**. Creating augmented reality with ar.js and a-frame. 2017. Disponível em: <https://aframe.io/blog/arjs/>. Acesso em: 09 nov. 2020.

ANATEL - Agência Nacional de Telecomunicações, 2020. **Telefonia móvel**. Disponível em: <https://www.anatel.gov.br/paineis/aceessos/telefonia-movel> Acesso em: 09 nov. 2020.

ARCORE, **ARCore**. ARCore overview. 2019. Página Web. Disponível em: <https://developers.google.com/ar/discover>. Acesso em: 09 nov. 2020.

ARCORE, **ARCore**. ARCore supported devices. 2020. Página Web. Disponível em: <https://developers.google.com/ar/discover/supported-devices>. Acesso em: 09 nov. 2020.

AZUMA, R. T. A Survey of Augmented Reality. **Presence: Teleoperators and Virtual Environments**, [s. l.], p. 355-385, Agosto 1997. Disponível em: <https://doi.org/10.1162/pres.1997.6.4.355>. Acesso em: 08 maio 2020.

BASSANI, P. S. Realidade aumentada na escola: experiências de aprendizagem em espaços híbridos. **Revista Diálogo Educacional**, v. 19, n.62, p. 1174-1198, 2019. Disponível em: <http://dx.doi.org/10.7213/1981-416X.19.062.DS13>. Acesso em: 10 out. 2020.

BRASIL. **Base Nacional Comum Curricular**. Brasília, 2017. Disponível em: http://basenacionalcomum.mec.gov.br/images/BNCC_EI_EF_110518_versaofinal_site.pdf. Acesso em: 09 set. 2021.

BRASIL. **Decreto nº 5.626, de 22 de dezembro de 2005**. Disponível em: http://www.planalto.gov.br/ccivil_03/_ato2004-2006/2005/decreto/d5626.htm. Acesso em: 30 set. 2020.

CARVALHO, D. **Software em língua portuguesa/libras com tecnologia de realidade aumentada**: ensinando palavras para alunos com surdez. 2011. 144 p. Tese (Doutorado em Educação) - Universidade Estadual Paulista, [S. l.], 2011. Disponível em: https://repositorio.unesp.br/bitstream/handle/11449/102184/carvalho_d_dr_mar.pdf?sequence=1&isAllowed=y. Acesso em: 3 abr. 2020.

CARVALHO, D. et al. Application of a Learning Program of Words in Libras Using Virtual Reality Technology. **Revista Brasileira de Educação Especial**, [s. l.], v. 23, n. 2, p. 215-232,

jun., 2017. Disponível em: <https://doi.org/10.1590/s1413-65382317000200005> . Acesso em: 3 abr. 2020.

CAUDELL, T. P.; MIZELL, D. W. Augmented Reality: an application of heads-up display technology to manual manufacturing processes. **IEEE Xplore**, [s. l.], 1992. Disponível em: <https://doi.org/10.1109/HICSS.1992.183317>. Acesso em: 15 maio 2020.

CERQUEIRA, J. M. **O jogo footmath – funções básicas da matemática com realidade aumentada**. 2019. Dissertação (Mestrado) - Instituto Politécnico do Cávado e do Ave, Mestrado em Engenharia em Desenvolvimento em Jogos Digitais, 2019. Disponível em: <http://ciencipca.ipca.pt/handle/11110/1886> Acesso em: 10 set. 2020.

CORREIA, D. et al. Desenvolvimento de um software educacional para o ensino aprendizagem de LIBRAS utilizando realidade aumentada. *In: Simpósio Internacional de Inovação em Mídias Interativas*, 2018, Goiânia. **Anais [...]**. Goiânia: [s. n.], 2018. 12 p. Disponível em: https://files.cercomp.ufg.br/weby/up/777/o/2_-_Derek_Correia.pdf . Acesso em: 3 abr. 2020.

GLTF. **Gltf**. glTF Overview. Khronos Group, 2020. Página Web. Disponível em: <https://www.khronos.org/gltf/>. Acesso em: 09 nov. 2020.

HIRZER, M. Marker Detection for Augmented Reality Applications. 2008. **Technical Report**. Disponível em: https://www.researchgate.net/publication/321253407_Marker_Detection_for_Augmented_Reality_Applications. Acesso em: 9 nov. 2020.

IBGE - Instituto Brasileiro de Geografia e Estatística. **Censo Demográfico 2010: características gerais da população, religião e pessoas com deficiência**. 2010. Disponível em: https://biblioteca.ibge.gov.br/visualizacao/periodicos/94/cd_2010_religiao_deficiencia.pdf. Acesso em: 30 de set. 2020.

IPEA - Instituto de Pesquisas Econômica Aplicada. **A infraestrutura sanitária e tecnológica das escolas e a retomada das aulas em tempos de covid-19**. 2020. Disponível em: https://www.ipea.gov.br/portal/images/stories/PDFs/nota_tecnica/200715_nt_diset_n_70_web.pdf. Acesso em: 08 de nov. 2020.

IPEA - Instituto de Pesquisas Econômica Aplicada. **Acesso domiciliar à internet e ensino remoto durante a pandemia**. 2020. Disponível em: http://repositorio.ipea.gov.br/bitstream/11058/10228/1/NT_88_Disoc_AcesDomInternEnsinoRemoPandemia.pdf . Acesso em: 08 de nov. 2020.

LARAVEL, **Laravel**. Eloquent: Getting Started. 2020. Página Web. Disponível em: <https://laravel.com/docs/7.x/eloquent#introduction>. Acesso em: 21 fev. 2021.

LOPES, L. M. D. et al . Inovações educacionais com o uso da realidade aumentada: uma revisão sistemática. **Educ. rev., Belo Horizonte** , v. 35, 2019 . Disponível em: <https://doi.org/10.1590/0102-4698197403>. Acesso em: 09 Nov. 2020.

MARIADB, **MariaDB**. About MariaDB Server. 2021. Página Web. Disponível em: <https://mariadb.org/about/#entry-header>. Acesso em: 21 fev. 2021.

MEC. **Ministério da Educação. Base Nacional Comum Curricular**. 2017. Disponível em: <http://basenacionalcomum.mec.gov.br/wp-content/uploads/2018/02/bncc-20dez-site.pdf>. Acesso em: 09 nov. 2020.

MOTTIN, Guilherme da Silva; BERTAGNOLLI, Silvia de Castro. RALibras: um protótipo *offline* para introduzir o ensino de Libras. *In: Workshops do Congresso Brasileiro de Informática na Educação (WCBIE)*, 9. , 2020, Online. **Anais [...]**. Porto Alegre: Sociedade Brasileira de Computação, 2020 . p. 171-178. DOI: <https://doi.org/10.5753/cbie.wcbie.2020.171>.

- NANOHTTPD. **NanoHttpd**. NanoHTTPD - a tiny web server in Java. 2020. Página Web. Disponível em: <https://github.com/NanoHttpd/nanohttpd>. Acesso em: 09 nov. 2020.
- NAZARETH, Débora Rabelo et. al. ELRA - Ensino de Libras usando Realidade Aumentada. **ISDEA '14: Proceedings of the 2014 Fifth International Conference on Intelligent Systems Design and Engineering Application**, [s. l.], p. 110-113, maio 2014. Disponível em: <https://doi.org/10.1590/0102-4698197403>. Acesso em: 3 abr. 2020
- PEREIRA, L. T; et al. EducAR – Quadrics: Uma Ferramenta de Apoio ao Ensino de Cálculo com Realidade Aumentada. **Revista Brasileira de Informática na Educação – RBIE**, v.27, n.2, p. 198-217, 2019. Disponível em: <https://www.doi.org/10.5753/RBIE.2019.27.02.198>. Acesso em: 9 nov. 2020
- RETROFIT, Square Inc. A type-safe HTTP client for Android and Java. 2021. Disponível em: <https://square.github.io/retrofit/>. Acesso em: 24 fev. 2021.
- STATCOUNTER. **StatCounter**. Mobile Operating System Market Share Brazil. 2018. Disponível em: <https://gs.statcounter.com/os-market-share/mobile/brazil>. Acesso em: 08 nov. 2020.
- VUFORIA. **Vuforia**. Vuforia Engine - Developer Portal. 2018. Página Web. Disponível em: <https://developer.vuforia.com/pricing>. Acesso em: 09 nov. 2020.
- WEBGL. **Webgl**. WebGL 2.0 Specification. Khronos Group, 2020. Página Web. Disponível em: <https://www.khronos.org/registry/webgl/specs/latest/2.0/#1>. Acesso em: 09 nov. 2020.
- WEBRTC. **Webrtc**. Real-time Communication - WebRTC. 2019. Página Web. Disponível: <https://webrtc.org/>. Acesso em: 09 nov. 2020.

APÊNDICE A

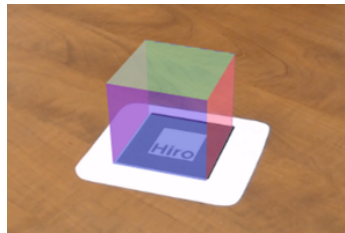
MARCADOR HIRO

Figura A.1 - Marcador bitonal em preto e branco, padrão Hiro



Fonte: <https://aframe.io/blog/arjs/> Acesso em novembro de 2020.

Figura A.2 - Exemplo do uso do marcador Hiro para projeção de um cubo em RA



Fonte: <https://stemkoski.github.io/AR-Examples/> Acesso em novembro de 2020.

APÊNDICE B

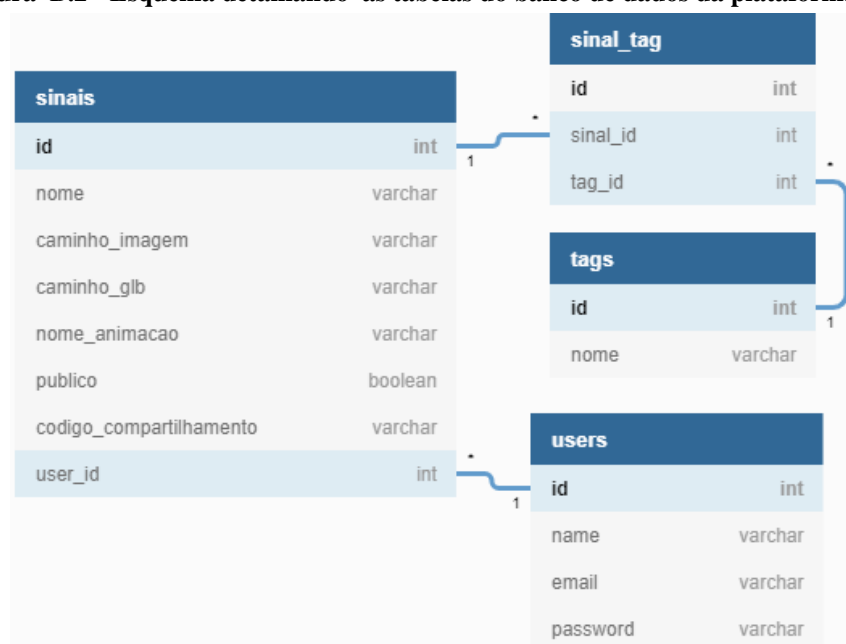
ESQUEMA DAS TABELAS DO BANCO DE DADOS DA PLATAFORMA WEB

Figura B.1 - Esquema detalhando a tabela que armazena os sinais no aplicativo *mobile*

sinaisLibra	
codigo_compartilhamento	varchar
nome	varchar
caminho_imagem	varchar
caminho_glb	varchar
nome_animacao	varchar

Fonte: do Autor, 2021.

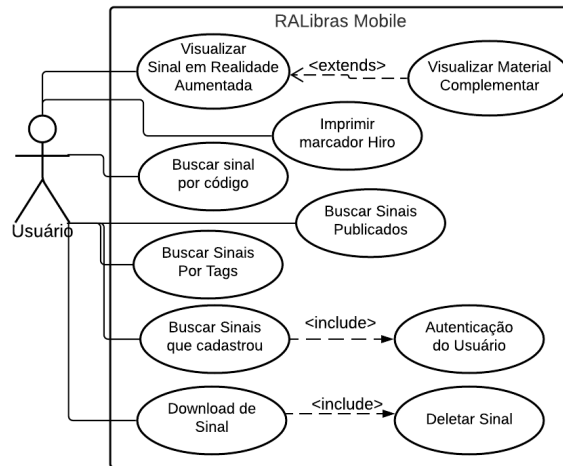
Figura B.2 - Esquema detalhando as tabelas do banco de dados da plataforma *web*



Fonte: do Autor, 2021

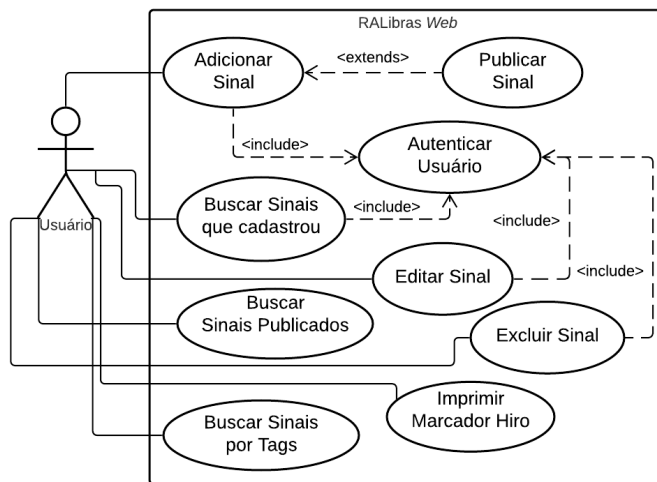
APÊNDICE C
DIAGRAMA DE CASOS DE USO

Figura C.1 - Diagrama de Casos de Uso da parte *mobile* da aplicação



Fonte: do Autor, 2021.

Figura C.2 - Diagrama de Casos de Uso da parte *web* da aplicação



Fonte: do Autor, 2021.