

COMPONENTE DE ÁUDIO 3D NO FLUTTER PARA APLICAÇÕES INTERATIVAS

Ascênio Sanderson Bezerra Monteiro, Bruno de Sousa Monteiro

Resumo: Dado a competição acirrada do mercado digital, é cada vez mais difícil criar um produto que se destaque entre tantos outros. Para isso, faz-se necessário promover inovação. Para aplicativos, isso é traduzido no aumento da interatividade entre o usuário e o dispositivo, por exemplo, na forma de áudio. Visando produzir uma solução genérica que possa servir para os diversos mercados (turismo, marketing, cultura, entretenimento, etc) e também como ferramenta de acessibilidade, este trabalho propõe um componente de áudio 3D que encapsula o tratamento de áudios com características de distância, orientação e atenuação, simulando um ambiente virtual sonoro. Por ter sido desenvolvido para o Flutter¹, pode ser adaptado posteriormente para múltiplas plataformas e abstraído para fácil uso do cliente. Entretanto, para o escopo deste trabalho, foi implementado apenas para Android, devido às restrições da plataforma para iOS. Para validação, foi criado uma aplicação Flutter com lugares geolocalizados, de modo que os usuários pudessem estar consumindo suas respectivas descrições por meio de áudio 3D.

Palavras-chave: inovação; áudio 3D; interatividade; ambiente virtual; multiplataforma.

1. INTRODUÇÃO

Com o avanço da tecnologia e a popularização/acesso aos computadores, tornou-se comum o uso dos dispositivos para solução de problemas e mais diversos usos no dia a dia. Bancos digitais, compra de comida, comunicação em texto, ligações, entretenimento na forma de livros, filmes ou jogos são apenas algumas das muitas coisas presentes no cotidiano de um usuário. Seja por meio de computadores ou celulares, as atividades do dia a dia estão cada vez mais digitais.

Uma das formas que isso vem sendo posto em prática é por meio da tecnologia de áudio 3D, também conhecido como áudio espacial. O áudio 3D consiste basicamente na ideia de tocar sons por meio de fones de ouvidos, de forma que o ouvinte possua sentimentos de posição e orientação. Esse recurso pode ser usado nos mais diversos domínios. Em jogos, o visual normalmente recebe muita atenção e capricho, mas recentemente a parte do áudio também vem sendo mais refinada para que o usuário se sinta mais imerso em determinado mundo virtual [1]. Na educação vem sendo usado como uma ferramenta para potencializar a absorção do conhecimento visto em sala de aula [2]. No design, é usado como ferramenta de estudo de acessibilidade para avaliação da experiência do uso de aplicativos móveis por deficientes visuais, para melhorar adaptar esses apps para aqueles com deficiência [3]. O setor do turismo também vem se beneficiando dessas tecnologias, pois possibilitam visibilidade (de lugares, produtos e serviços), *feedback* rápido dos consumidores, rapidez na logística e melhoria na qualidade dos serviços prestados [4]. No Brasil, a indústria do turismo representa um importante setor, que só em 2019 movimentou R\$ 238,6 bilhões de reais (2,2% a mais do que em 2018) e gerou 35.692 novos postos de trabalho (alta de 164% comparado com 2018) [5]. Para ilustrar esse potencial, no Brasil, o turismo é responsável por 7,7% da economia, porém, a média mundial é de 10,3% e na Espanha, por exemplo, a participação é de 14,3% [6].

Ao combinar a solução do áudio com os dispositivos móveis encontra-se uma oportunidade de experiência de usuário que pode vir a beneficiar muitos tipos de usuários. Aplicativos, em especial, têm ganhado maior visibilidade nos últimos anos com a explosão de redes sociais e popularização dos celulares. De acordo com a pesquisa “State of Mobile 2022: Brazil”, publicada em 2022, usuários no Brasil passam 1/3 das horas em que estão acordados em dispositivos móveis [7]. Isso mostra que especialmente no Brasil, investir em *apps* de forma geral pode ser bastante interessante, ainda mais quando considerado os diversos domínios aos quais podem estar relacionados.

Portanto, neste trabalho será desenvolvido uma solução de áudio 3D para enriquecimento de experiência de uso em aplicativos móveis, visando produzir um componente reaproveitável de forma independente ao domínio da aplicação/problema. A fim de que seja acessível ao máximo número de usuários possível, a solução será desenvolvida com Flutter¹, um *framework* para desenvolvimento de aplicativos multiplataforma e terá como estudo

¹ Flutter: <https://flutter.dev>

de caso uma aplicação fictícia de turismo. Neste app, são emitidos sons contendo curiosidades ou informações históricas da cidade de Apodi/RN a partir de alguns pontos definidos no mapa. Estes serão fixos e servirão de insumo para executar um experimento. A aplicação servirá de objeto de estudo para definir metas e estudar limitações da solução encontrada.

1.1 Objetivo

Para o escopo deste trabalho, tem-se como objetivo desenvolver uma solução, facilmente portátil entre plataformas, para experiências interativas em dispositivos móveis na forma de áudio com características de direção e profundidade (áudio 3D), tendo como parâmetros a posição geográfica e a orientação do usuário.

1.2 Atividades

Para alcançar o objetivo estabelecido, foram definidas as seguintes atividades:

1. Pesquisar sobre soluções existentes (bibliotecas) que pudessem dar suporte à solução do problema, ou seja, processamento de áudio 3D.
2. Com base na análise de pontos positivos e negativos (descritos na seção seguinte), realizar um estudo mais aprofundado sobre a biblioteca OpenAL, para ser integrada ao Flutter.
3. Desenvolver um módulo portátil da biblioteca OpenAL em Dart², de modo a executar áudio 3D.
4. Desenvolver um aplicativo mobile para consumir a biblioteca anteriormente desenvolvida e cujo áudio fonte é produzido por texto sintetizado, proveniente dos pontos de interesse criados no mapa.
5. Testar o aplicativo desenvolvido por um grupo de usuários de diferentes faixas etárias, cujos relatos foram coletados por meio da técnica *design thinking* utilizada para definir sucesso/limitações da solução.

1.3. Fundamentação teórica

Em função do número crescente de dispositivos móveis [8] e devido à crescente complexidade de manter aplicações Android e iOS, surgiram *frameworks* como React Native³ e Flutter. Seu objetivo é, a partir de uma mesma base de código, produzir como saída aplicações para ambas plataformas. Com Flutter, a aplicação pode ser tão performática quanto uma aplicação nativa, pois a compilação emite binários nativos correspondentes à plataforma alvo [9].

Embora Flutter tenha crescido e esteja ajudando grandes empresas como Nubank, a entregar um banco escalável [10]; eBay motor, a vender carros para entusiastas automobilísticos [11], ainda é uma tecnologia relativamente nova e não possui muita bibliografia sobre seu uso. No que diz respeito ao desenvolvimento de aplicações voltadas para áudio 3D, a bibliografia demonstrou-se bastante limitada. Não existem tantas publicações voltadas especificamente para o *mobile* e aquelas que existem são focadas em uma plataforma específica.

No trabalho de LOPES, é continuado o desenvolvimento de um trabalho predecessor sobre aplicação de interatividade por meio de áudio 3D. O trabalho foi desenvolvido usando somente entradas e saídas disponíveis nos próprios dispositivos iOS, como GPS por exemplo. Foram programados algoritmos de processamento digital com Objective C++ e utilizando-se da Accelerate framework, uma solução nativa iOS para cálculo de álgebra linear de alta performance. O resultado foi uma biblioteca denominada AAVE que foi usada para simular uma sala virtual com fontes sonoras em locais estratégicos, como uma espécie de museu interativo. A solução conseguiu emitir sons com erro absoluto inferior a 16 cm em 95% das vezes [12].

TRINTA desenvolveu um trabalho com objetivo diferente, mas com a mesma tecnologia. Seu objetivo era investigar o uso do áudio 3D como forma de facilitar o uso de dispositivos móveis por meio de pessoas com deficiência (PCD) visuais. Um método de avaliação de usabilidade consiste em avaliar o humor dos usuários após utilizar um produto. Essa avaliação se dá por meio da seleção contínua de cartões visuais que possuem uma emoção escrita/denhada após cada etapa de um experimento, contabilizando e avaliando a experiência de forma positiva e negativa no geral. Para deficientes visuais a abordagem é diferente: esses cartões são substituídos por onomatopéias. Utilizando-se dessa técnica, a pesquisa concluiu que mesmo que existam ferramentas de acessibilidade prontas para uso nos dispositivos, a experiência que elas entregam não é satisfatória para o problema que atendem. Por meio do uso do áudio espacial (aquele que possui sentimentos de direção e sentido), foi possível entregar uma experiência que recebeu feedback positivo tanto de usuários PCD como não-PCD.

Dessa forma, considerando os benefícios do uso do áudio 3D nas mais diversas aplicações, seja por razões de acessibilidade ou para aumentar a imersão do usuário, e também considerando a escassez de trabalhos no que

² Dart: <https://dart.dev>

³ React Native: <https://reactnative.dev/>

diz respeito às soluções multiplataforma, este trabalho foi desenvolvido. Busca-se contribuir para a academia e também para o mercado, provendo uma solução que pode ser utilizada tanto no Android como no iOS.

2. DESENVOLVIMENTO

Nesta seção serão apresentados os detalhes de execução de cada uma das atividades realizadas nesse trabalho, tais como: artefatos de entrada, técnicas e ferramentas adotadas, tarefas, dificuldades e resultados alcançados.

2.1 Análise de bibliotecas para processamento de áudio 3D

Inicialmente foi analisado qual seria a biblioteca de processamento de áudio 3D que serviria de base para a solução. Foram consideradas principalmente características como portabilidade, preço e facilidade de uso. Ao final, a Tabela 1 sumariza as principais diferenças entre os projetos analisados.

2.1.1. Agora 3D Spatial Audio⁴

Agora 3D Spatial Audio é um produto fornecido pela empresa Agora, especializada em serviços de *streaming* de vídeo e áudio em tempo real. Disponível tanto para *mobile* como *web*. Entretanto, seu uso gratuito é limitado aos primeiros 10 mil minutos mensais, sendo acima disso cobrado o valor de 99 centavos de dólar por mil minutos. A dependência de terceiros e aumento de custos, além da existência de alternativas gratuitas, tornam esta opção menos interessante.

2.1.2. Surround Sound⁵

Este pacote foi feito para Flutter e funciona por meio de uma *WebView* dentro do app, ou seja, são usadas as APIs de áudio padrões do navegador. Dessa forma, é relativamente portátil, porém sofre de problemas de performance já que não é uma solução totalmente nativa.

2.1.3. OpenAL⁶

OpenAL é uma API de áudio 3D multiplataforma. Funciona de forma similar ao OpenGL: do ponto de vista do usuário existe uma máquina de estados, métodos que alteram e métodos que usam os estados. Da mesma forma como OpenGL é uma API e cada placa de vídeo tem sua própria implementação, OpenAL depende de implementação de terceiros, sendo OpenAL Soft⁷ uma das mais comumente utilizadas. No Android, esta faz uso de bibliotecas de áudio de baixo nível, como AAudio e OpenSL, abstraindo detalhes de implementação e tornando transparente para o cliente. Em geral, é bem documentada com exemplos e explicação de sua API [13].

2.1.4. Spatial Audio Framework⁸

Este framework contém um conjunto de módulos focados em diversas subáreas do áudio espacial: encoding/decoding, simulação de sala, etc. Criada inicialmente para o campo da pesquisa e posteriormente publicada para uso geral por Leo McCormack, especialista na área e atualmente candidato a PhD, esta ferramenta possui uma gama de recursos avançados disponíveis bastante diversificada e também expansível por plugins. Desenvolvida na linguagem C, usando bibliotecas de álgebra linear famosas (como OpenBLAS) e SIMD, entrega um bom desempenho. Entretanto, como a maioria dos seus recursos não são necessários para a solução do problema e sua complexidade de implementação é extremamente alta, tanto pela abrangência de recursos como pela falta de exemplos online, não é uma boa ferramenta para o problema.

2.1.5. Comparativo

⁴ Agora 3D Spatial Audio: <https://www.agora.io/en/products/3d-spatial-audio>

⁵ Surround Sound: https://pub.dev/packages/surround_sound

⁶ OpenAL: <https://www.openal.org>

⁷ OpenAL Soft: <https://github.com/kcat/openal-soft>

⁸ Spatial Audio Framework: https://github.com/leomccormack/Spatial_Audio_Framework

A seguir, na Tabela 1, é apresentado um resumo do comparativo entre essas quatro bibliotecas, descritas acima, sob cinco parâmetros: gratuidade, performance, complexidade, necessidade de ponte e estado do projeto.

Tabela 1. Comparativo entre principais recursos das bibliotecas (autoria própria)

Biblioteca	Gratuito	Boa performance	Complexidade de uso	Exige ponte (FFI)	Projeto descontinuado
Agora 3D Spatial Audio	NÃO	SIM	MÉDIA	SIM	NÃO
Surround Sound	SIM	NÃO	BAIXA	NÃO	SIM
OpenAL	SIM	SIM	MÉDIA	SIM	NÃO
Spatial Audio Framework	SIM	SIM	ALTA	SIM	NÃO

A coluna de performance foi avaliada a partir do conjunto de dependências de software para uma determinada biblioteca. Não houve medição empírica da real performance pois seria necessário compreender o funcionamento, desenvolver um exemplo, medir e comparar cada uma das opções. Percebeu-se que como Surround Sound é executado dentro de um navegador, possui mais limitações e fará uso de uma abstração desnecessária para o problema, enquanto que as demais acessam diretamente os recursos disponíveis na plataforma.

A complexidade foi definida por alguns fatores: nível de abstração e necessidade de ponte. O surround sound e OpenAL possuem primitivas de simples entendimento e uso: basta passar a localização do usuário e das fontes sonoras, a biblioteca trata do restante. Por essa razão, o surround recebeu a classificação de “baixa” complexidade. Isso servirá como métrica de comparação com as demais. Com relação ao OpenAL, apesar de ser semelhante ao surround, exige ponte, por isso sua complexidade é superior e categorizada como mediana. Já o Agora é uma solução de alto nível, mas com ponte, por isso recebeu a mesma categoria que o OpenAL. Já o Spatial Audio é uma solução de baixo nível, com poucas camadas de abstrações. O usuário possui maior controle do que deseja, possui maior flexibilidade na medida que se torna mais difícil de ser usada. Por essa razão, recebeu a classificação de maior complexidade.

Dadas as opções acima, todas as que possuem performance satisfatória exigem ponte. Além disso, o Surround não recebe atualizações há cerca de 19 meses, portanto a solução teria que ser nativa. Assim, tratando-se de uma biblioteca de software livre e, dados os pontos positivos e negativos das demais opções, a OpenAL foi escolhida para ser adotada na implementação deste trabalho.

2.2 Estudo sobre a biblioteca OpenAL

O OpenAL funciona por meio de alguns conceitos fundamentais, são eles: *buffer*, *source* e *listener*. Os *buffers* armazenam os áudios, os *sources* (ou simplesmente fontes) representam fontes de som posicionadas no mundo e o *listener* (ouvinte) é o próprio usuário. A depender da posição do usuário e sua orientação no mundo virtual, projeta-se o som do lado correto ao fone de ouvido e são aplicados efeitos de atenuação e direção.

A imagem acima representa o modelo padrão responsável pelo efeito de atenuação, isto é, a perda de volume à medida que a distância entre o ouvinte e uma fonte sonora aumenta e, analogamente, o ganho de volume à medida que se aproximam. O modelo, conhecido por *Inverse Distance Model* (Figura 1), é configurável diante dos parâmetros *AL_REFERENCE_DISTANCE* e *AL_ROLLOFF_FACTOR*. O primeiro define a distância mínima para qual o ganho é máximo (*clamped*), enquanto o segundo define a taxa de queda do ganho conforme o aumento da distância.

Existem outros modelos como *Exponential Distance Model* e *Linear Distance Model* que podem ser mais interessantes a depender do efeito desejado. Cada combinação modelo-parâmetros pode entregar resultados completamente diferentes, mas como o problema não requer uma percepção específica de áudio foram utilizados modelo e valores padrão.

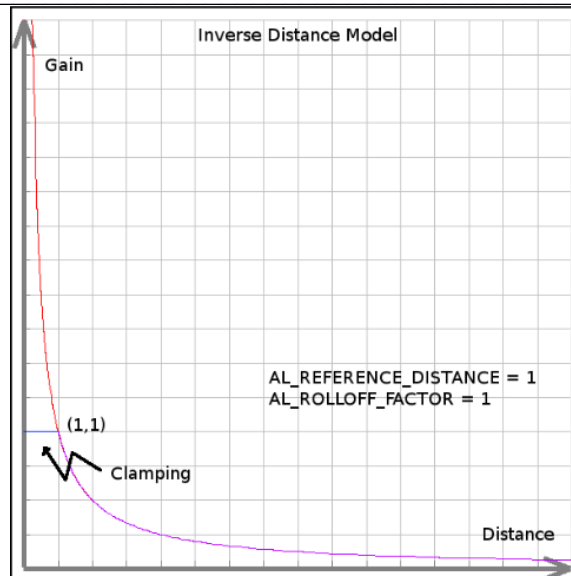


Figura 1. Gráfico do Inverse Distance Model. (OpenAL)

Na Figura 2 é possível observar a intensidade sonora percebida a depender da posição e direção entre os objetos. O usuário 1 encontra-se suficientemente perto de duas fontes sonoras, tal que consegue ouvir ambas simultaneamente, porém o ouvido de lado oposto à fonte sonora sofre perda. Naturalmente, se estiver virado diretamente para a fonte, ambos ouvidos ouvirão com clareza. Para o usuário 2 o ganho em ambos ouvidos seria próximo de 1.0, enquanto que para o usuário 1 seria, por exemplo, 0.75 para fonte de lado correspondente ao ouvido e 0.5 para o outro.

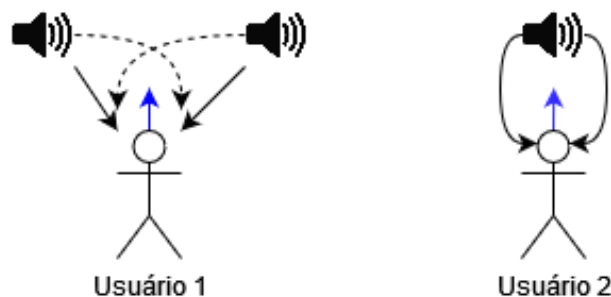


Figura 2. Percepção da direção. (Autoria própria)

2.3 Implementação do módulo portátil da biblioteca OpenAL para Dart

Inicialmente, precisava-se determinar a estabilidade do OpenAL em relação às demais opções discutidas anteriormente. Como não há documentação oficial específica para o mobile e não foram encontradas bibliografias que fizessem o mesmo que este trabalho, qualquer impasse implicaria no reinício do desenvolvimento. Portanto, foi feita uma aplicação protótipo para desktop para aplicar os conhecimentos sobre as abstrações da biblioteca sendo executada em sua plataforma mais estável. O protótipo exercitou dois cenários: o toque de um áudio com o ouvinte parado e em movimento.

Para tocar o áudio em posição fixa foi utilizado a documentação oficial e exemplos online para atingir um exemplo mínimo funcional. Foi necessário usar ALUT⁹, a biblioteca de utilitários do OpenAL, para fazer o carregamento de arquivos WAV e obter amostras de áudio em memória. Em seguida, para testar posições dinâmicas foi suficiente rotacionar a fonte sonora ao redor da origem, posição padrão do ouvinte. Assim, foi validado que a biblioteca entrega todas as funcionalidades necessárias ao trabalho e que o comportamento do app deveria ser o mesmo, desde que as funções fossem chamadas na mesma ordem.

⁹ ALUT: <http://distro.ibiblio.org/rootlinux/rootlinux-ports/more/freealut/freealut-1.1.0/doc/alut.html>

Adaptar o exemplo para Flutter foi um problema inicialmente por falta de documentação. Pielotd escreve em seu *blog* como fazer a compilação da biblioteca utilizando Make¹⁰ como sistema de *build* [14]. Entretanto, esse método implica em listar manualmente todos os arquivos C do projeto que devem ser compilados, exigindo potencial manutenção extra conforme novas versões são lançadas. Além disso, o projeto OpenAL Soft faz uso do CMake¹¹ que é oficialmente suportado pela plataforma, então usá-lo seria a opção menos custosa.

Embora os experimentos tenham sido feitos em C++, o OpenAL é exportado por meio de uma interface C e nenhum dos dois pode ser chamado diretamente pelo Dart (linguagem de programação da framework Flutter). Por isso, após adaptar o processo de build da aplicação para incluir código nativo, foi necessário desenvolver uma ponte por meio de *foreign function interface* (FFI). Essa ponte (também conhecida como *binding*) é necessária para permitir que uma linguagem (neste caso Dart) possa chamar rotinas em outra linguagem (neste caso funções nativas).

Na Figura 3 está representada a arquitetura da aplicação. A parte verde representa o código Dart e a parte azul, código C. Como foi dito acima, há necessidade da ponte para fazer a comunicação entre os dois lados, e na prática existe um mapeamento de 1 para 1 entre as duas linguagens. Significa que para cada método em C que se pretende chamar, deve haver um código Dart correspondente. Além disso, é necessário que as assinaturas das funções sejam equivalentes, isto é: a tipagem de parâmetros e retorno devem ser os mesmos. Além disso, para que dada função seja chamada, é necessário encontrar o símbolo de mesmo nome no *shared object* do OpenAL. Por esses motivos, foi utilizado o Pigeon¹², uma ferramenta oficial do Flutter para automatizar a geração desse mapeamento a partir de cabeçalhos C/C++.

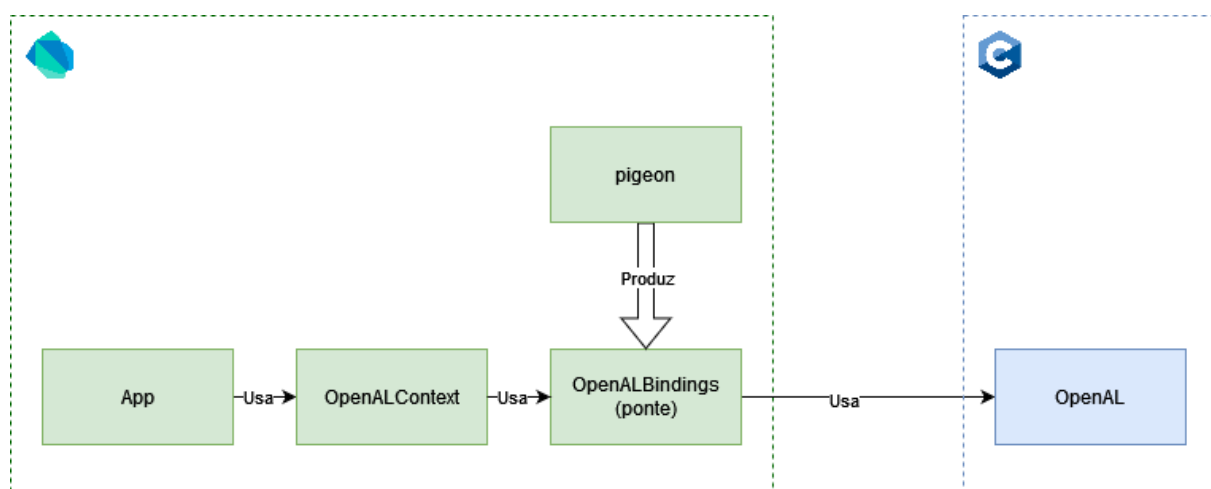


Figura 3. Arquitetura da aplicação. (Autoria própria)

A partir da ponte é possível chamar quaisquer funções do OpenAL conforme se faça necessário, porém dado uma certa necessidade há uma sequência de chamadas que devem ser seguidas, exemplo: para inicializar a biblioteca, é necessário:

1. Abrir o arquivo da biblioteca
2. Carregar o dispositivo
3. Criar um contexto a partir do dispositivo encontrado
4. Definir a posição e orientação padrão do ouvinte
5. Desalocar memória

Em alguns casos, se a ordem não for obedecida o áudio apenas não será tocado. Porém, na situação acima, se o usuário tentar executar a etapa 4 antes da etapa 3 é possível que a aplicação sofra um *crash* por tentar chamar métodos antes mesmo que a inicialização tenha finalizado. É importante deixar essas chamadas próximas umas das outras no código de forma que possam ser facilmente verificadas. Além disso, seria conveniente que casos genéricos como inicialização, emitir áudio, remoção de áudio e liberação de memória fossem abstraídos por meio de um terceiro, responsável por garantir a ordem e encapsulamento. Por esses motivos foi introduzido o OpenALContext (consulte Figura 3). Cada aplicação pode criar o seu próprio para resolver seus problemas específicos, mas também pode ser genérico e assim facilitar o reuso entre diferentes aplicações.

¹⁰ Make: <https://www.gnu.org/software/make>

¹¹ CMake: <https://github.com/kcat/openal-soft>

¹² Pigeon: <https://pub.dev/packages/pigeon>

Neste ponto, era possível tocar áudios 3D no Android sem problemas, mas nenhum protótipo havia sido testado no iOS. Investigou-se a possibilidade de suportar a plataforma, porém, devido à falta de equipamento para desenvolvimento (Mac), carência de tempo e complexidade, a ideia não foi levada adiante.

2.4 Implementação do aplicativo Flutter

Na seção 2.3 foi implementado o módulo necessário para função básica: tocar áudios 3D. Nesta seção, será abordada a construção do app que faz uso desse módulo e outras integrações.

A primeira funcionalidade desenvolvida foi a visualização do mapa. Nele são posicionados os POIs (pontos de interesse) e é exibido a localização do usuário em tempo real. Esses dois recursos quando usados em conjunto auxiliam na validação do funcionamento da atenuação e, para o usuário final, facilita a descoberta de pontos próximos. Para a visualização foi utilizado o pacote *google_maps_flutter*¹³ e para obter a localização utilizou-se do pacote *geolocator*¹⁴.

Nos primeiros testes, verificou-se que ao se distanciar de uma fonte de som, a intensidade sonora diminui, e ao se aproximar, aumenta, como esperado. Isso foi possível devido ao mapeamento da latitude e longitude como coordenadas dentro do mundo virtual do OpenAL. Mas isso causou um problema: uma distância de vários metros é responsável por uma mudança pequena nas coordenadas, não chega a 1° de diferença e portanto o usuário escuta sempre o áudio com o mesmo volume. Fazia-se necessário encontrar uma transformação das coordenadas terrestres para algum sistema de coordenadas onde distâncias em metros na vida real correspondem também a várias unidades de medida. Para essa tarefa foi escolhido usar UTM. UTM é um sistema de projeção baseado no plano cartesiano que faz uso de metros como unidade de medida. “O sistema UTM foi recomendado pela IUGG (International Union of Geodesy and Geophysics) para a cartografia em pequenas e médias escalas” [15]. Assim, trata-se de uma projeção interessante para este problema, uma vez que o usuário se locomove em baixa velocidade. “Para trabalhos cartográficos urbanos, analógicos ou digitais, cujas escalas variam de 1/10.000 até 1/500, a projeção mais utilizada no Brasil é a U.T.M.” [16].

Por algum tempo foi usado um arquivo de áudio genérico e fixo, porém para obter áudios dinâmicos a partir de textos foi necessário sintetizar áudios por meio do *text to speech* do Android. O áudio produzido é um arquivo WAV, porém nenhum som estava sendo emitido. Ao investigar o protótipo nativo foi identificado que o áudio usado como exemplo possui amostras representadas como inteiros de 16 *bits*, enquanto que aqueles produzidos pelo sintetizador tinham como saída amostras de ponto flutuante. Portanto, foi necessário realizar esse mapeamento para que a aplicação pudesse voltar a funcionar novamente.

Por fim, bastava definir a orientação do usuário com base no compasso. Assim, se a fonte sonora estiver exatamente à esquerda do ouvinte, deve sentir que o som vem da esquerda; se a fonte estiver à direita, deve ouvir somente do lado direito. Entretanto, no Android em específico não existe um método padrão para obter o compasso diretamente, seria necessário usar algum método alternativo. Inicialmente considerou-se usar a orientação proveniente do GPS, porém suas leituras não são frequentes e precisas o suficiente para captar mudanças sutis de direção. Para tanto, seria necessário fazer manualmente alguns cálculos com base na leitura dos sensores acelerômetro e magnetômetro, mas graças ao pacote *flutter_compass*¹⁵ não foi necessário.

Na Figura 4 é possível observar uma *screenshot* da aplicação rodando em um dispositivo físico. O usuário encontra-se próximo à biblioteca e está olhando levemente para a direita do POI do parque. O esperado é que nesse caso consiga ouvir o texto em ambos os lados, porém com maior intensidade do lado esquerdo.

¹³ Package *google_maps_flutter*: https://pub.dev/packages/google_maps_flutter

¹⁴ Package *geolocator*: <https://pub.dev/packages/geolocator>

¹⁵ Package Flutter: *flutter_compass*: https://pub.dev/packages/flutter_compass



Figura 4. Tela do aplicativo com os pontos de interesse, cujas descrições são apresentadas aos usuários em áudio 3D quando chegam perto. (Autoria própria)

Na imagem acima, percebe-se que a localização do testador é representada por dois elementos: o círculo azul e um POI sob ele. Enquanto o azul representa o que é visível para o Google Maps¹⁶, o vermelho representa o que é visível para a aplicação. Idealmente, ambos estariam sincronizados e posicionados nas mesmas coordenadas, porém na prática existe uma divergência entre eles.

Há uma reta que se inicia na posição do usuário (POI) e termina na circunferência vermelha. Essa reta demonstra a direção assumida para qual o usuário está direcionado, com base nos sensores acelerômetro e magnetômetro do dispositivo. No entanto, o comprimento da reta não possui nenhum significado especial, apenas possui um tamanho fixo por simplicidade.

2.5 Validação do aplicativo

Para validar a aplicação, foi feito um experimento na qual quatro fontes sonoras foram definidas em determinados pontos ao redor da praça Getúlio Vargas da cidade de Apodi. Cada um dos pontos emite um som contendo alguns fatos históricos da cidade. Foram selecionados três participantes, cuja tarefa foi caminhar ao redor desses pontos e descrever suas impressões sobre a experiência. Usando a técnica do *thinking out loud* (pensar em voz alta) [17] os usuários deram sua opinião no momento no qual ocorre a interação, provendo um *feedback* mais autêntico e facilitando a identificação de problemas comuns. Foi justificado para cada um deles a necessidade de gravar e todos estiveram de acordo.

O primeiro voluntário foi um homem adulto de 22 anos. Utilizou a aplicação desenvolvida no seu próprio celular, um Xiaomi Note 7. Percebeu-se que neste modelo de celular o cálculo da orientação se mostrou muitas vezes impreciso. Mesmo permanecendo parado, o compasso do dispositivo apontava muitas vezes no lado direito, quando o som deveria estar vindo da esquerda. Além disso, devido à lentidão do GPS em atualizar a posição, às vezes o áudio estava muito baixo mesmo que o indivíduo estivesse próximo de um ponto, fazendo com que o usuário aumentasse o volume. Porém quando a localização era atualizada para mais próximo o volume aumentava significativamente de forma súbita, devido à diferença entre as leituras. Durante seu teste, foram comparadas as telas do seu celular e o M53. Havia divergências significativas na precisão e tempo de resposta deles, sendo o M53 mais preciso e rápido.

¹⁶Google Maps: <https://www.google.com.br/maps>

O segundo participante tinha 42 anos e era do sexo feminino. Utilizou um Samsung Galaxy M53 e, de acordo com ela, obteve bons resultados: “... se você girar à direita dele, escuto mais a direita. Bem nítido mesmo...”. Além disso, afirmou que os pontos de teste apresentaram sons como esperado, sem atrasos. Durante o experimento, dirigiu-se para cada um dos pontos e apontou um dos ouvidos para eles, em seguida invertendo de lado e partindo para o ponto seguinte. Por ter usado um celular mais responsivo e ficado praticamente parada, não houve muita margem de erro no seu teste. O atraso mais evidente seria aquele que acontece ao fazer um giro de 180°, porém a mesma relatou não o ter presenciado.

A terceira participante, uma idosa de 69 anos, utilizando também o M53, parecia um pouco confusa sobre o que deveria ser feito e não estava confortável para fazer o teste sozinha. Por isso, foi acompanhada, recebendo sugestões de para onde ir, para qual lado virar, etc, mas evitando dar ordens e perguntando o que estava achando. “teve alguns cantos que deu atraso... O último estava escutando à direita o que era para ser escutado à esquerda.”, diz ela sobre os atrasos. No geral, ela achou o teste interessante mesmo diante de certas dificuldades técnicas. Um fator que pode ter influenciado neste caso é a forma de segurar o celular. Como foi dito anteriormente, o compasso utiliza a orientação do próprio celular no seu cálculo, porém a participante estava segurando o dispositivo na vertical, enquanto os demais seguraram na horizontal. Um resumo da avaliação dos três participantes pode ser resumido na Tabela 2.

Tabela 2. Resumo dos relatos do experimento (autoria própria)

Participante	Funcional	Possui atrasos	Possui bugs	Celular usado
Participante 1	SIM	SIM	SIM	Xiaomi Note 7
Participante 2	SIM	NÃO	NÃO	Samsung Galaxy M53
Participante 3	SIM	SIM	NÃO	Samsung Galaxy M53

A partir dos relatos é possível concluir que a solução estava de fato usando a posição e orientação para alterar a percepção sonora, porém o erro e latência entre as leituras do GPS e dos sensores demonstrou-se insuficiente para uma experiência totalmente fluida. Caso as origens dessas leituras não viessem do próprio dispositivo, em um jogo, por exemplo, acredita-se que não haveria tanto atraso.

3. CONCLUSÕES

Durante a pesquisa para o desenvolvimento deste trabalho foi possível compilar a biblioteca OpenAL Soft para Android e fazer uso das abstrações do OpenAL mesmo com ausência de documentação. O uso dessa biblioteca permitiu desenvolver uma solução multiplataforma que, embora tenha sido implementada somente para Android, pode ser adaptada para iOS ou até mesmo *desktop*.

Diversos mercados podem se aproveitar dessa solução para desenvolver aplicativos interativos. Para o turismo, foi entregue uma oportunidade de aumentar a descoberta e exploração dos ambientes. Museus podem definir pontos dentro de seus espaços e contar a história de objetos ou eventos históricos sem a presença de um guia. Também dispensa gastos com infraestrutura (caixas de som e energia). Todo o controle está no próprio dispositivo, não sendo necessário poluir o ambiente com ruído sonoro. O usuário poderia selecionar e ouvir somente aquilo que lhe interessa.

Para Computação, este trabalho serve como guia para implementar qualquer biblioteca nativa de forma *cross-platform* com Flutter. O mesmo processo poderia ser aplicado em outras plataformas e os problemas encontrados seriam nuances específicas de cada uma, mas a ideia geral se mantém. Isso é bastante favorável para aproveitar-se do ecossistema de bibliotecas C e C++ que possuem décadas de desenvolvimento e desenvolver soluções desde a fundação, quando necessário. Desenvolver jogos, *game engines*, etc. Soluções que precisam de performance e são problemas conhecidos.

O experimento revelou as limitações dessa solução para problemas que interagem com o mundo real. A atualização de localização e orientação devem ser atualizadas suficientemente rápido para que não haja latência perceptível pelo usuário. Por outro lado, essa limitação é esperada para qualquer trabalho que interage com o ambiente. Leitura de sensores sempre apresenta ruído e pode ser mitigado, mas não eliminado totalmente. Algumas ideias para melhorar isso seria fazer uso de filtros digitais como média móvel, mas pode tornar a leitura menos responsiva à mudanças. No geral, esses problemas são causados pelo domínio do problema, não pela solução em si. Além disso, a solução é capaz de tocar especificamente áudios WAV de 16 bits, qualquer outra entrada deveria ser convertido previamente.

3.1 Trabalhos Futuros

Como sugestão de trabalhos futuros, podem ser destacadas as seguintes pesquisas e desenvolvimentos:

- Empacotamento da solução Android na forma de um *package*, uma vez que não foi possível fazê-lo durante o desenvolvimento deste trabalho.
- Replicar o procedimento realizado aqui em Android para iOS e publicar ambos os trabalhos como um pacote no pub.dev¹⁷, repositório oficial de pacotes Flutter. Atualmente, para cada projeto que desejar fazer uso da solução é necessário replicar alguns passos feitos neste trabalho, mas o ideal seria que adicionar a dependência do pacote fosse suficiente para seu funcionamento em qualquer plataforma.
- Integrar o componente de áudio 3D desenvolvido neste trabalho com outro módulo de realidade aumentada, de modo a ampliar a imersividade do usuário.
- Avaliar o componente aqui desenvolvido em um aplicativo de turismo, já em produção, de modo que sua eficácia e eficiência possam ser testadas em cenários reais.

4. REFERÊNCIAS BIBLIOGRÁFICAS

- [1] JUNIOR, David, FORTE, Cleberson. ÁUDIO 3D EM JOGOS. Rev. Tec.FatecAM, v. 2, n. 1, p. 35-36, Mar. 2014.
- [2] COSTA, Bruna et. al. O ÁUDIO 3D COMO ELEMENTO PARA PRÁTICAS EDUCACIONAIS. In: Simpósio Internacional Interdisciplinar em Cultura e Sociedade do PGCult, 2., 2017, São Luís. Anais eletrônicos [...] São Luís: Salute, 2018. p. 583 - 599. Disponível em: https://www.academia.edu/43306869/O_%C3%81UDIO_3D_COMO_ELEMENTO_PARA_PR%C3%81TICAS_EDUCACIONAIS. Acesso em: 26 mai. 2023.
- [3] TRINTA, Ana. FERRAMENTA AUDITIVA 3D (AUDIO BINAURAL): desenvolvimento de um protótipo para o estudo da satisfação das pessoas com deficiências visuais na interação com interfaces digitais (smartphones). 2020. 172p. Dissertação (Mestrado) - Universidade Federal do Maranhão, São Luís.
- [4] GONDIM, Ruan dos Santos et al. Designing future scenarios in e-tourism. In: Proceedings of the 11th Euro American Conference on Telematics and Information Systems. 2022. p. 1-9.
- [5] Anuário Estatístico de Turismo 2020. Ano Base 2019. Disponível em: <<http://dadosefatos.turismo.gov.br/2016-02-04-11-53-05.html>> Acesso em: 7 de Maio de 2023.
- [6] WTTC. World Travel & Tourism Council. Disponível em: <<https://wtcc.org>> Acesso em: 7 mai. 2023.
- [7] State of Mobile 2022: Brazil. Data AI. 2022. Disponível em: <<https://www.data.ai/en/go/state-of-mobile-2022-brazil/>>. Acesso em: 7 de Maio de 2023.
- [8] DAHL, Ola. Exploring End User's Perception of Flutter Mobile Apps. 2019. 38p. Dissertação (Mestrado) - Universidade de Malmö, Malmö.
- [9] TASHILDAR, Aakanksha et. al. APPLICATION DEVELOPMENT USING FLUTTER. Rev. Irjmets, v. 2, n. 8, p. 1262-1266, Ago. 2020.
- [10] Scaling Nubank's mobile development with Flutter. Nubank, 2021. Disponível em: <<https://building.nubank.com.br/scaling-with-flutter/>>. Acesso em: 26 mai. 2023.
- [11] SPRAGUE, Corey, MCKENZIE, Larry. eBay Motors: Accelerating With Flutter™. Ebay, 2020. Disponível em: <<https://tech.ebayinc.com/product/ebay-motors-accelerating-with-fluttertm/>>. Acesso em: 26 mai. 2023.
- [12] LOPES, Sérgio et. al. Sistema de Realidade Aumentada 'Áudio 3D para Dispositivos iOS. In: Congresso de Engenharia de Áudio, 12., 2014, São Paulo. Anais [...] São Paulo: Salute, Sociedade de Engenharia de Áudio, 2014. p. 100 - 104.
- [13] HIEBERT, Garin et. al. OpenAL Programmer's Guide. Versions 1.0 and 1.1. 2007. 142p.
- [14] PIELOT, Martin. OpenAL on Android. Martin's Blog. 2010. Disponível em: <<https://pielot.org/2010/12/openal-on-android/>>. Acesso em: 7 mai. 2023.
- [15] SILVA, Daniel, FREDERICO, Lilian, COSTA, Glauber. Cálculo aproximado de fator de escala UTM para uso direto em estações totais, Rev. Bras. Geom, v. 4, n. 1, p. 59-67, Abr. 2016.

¹⁷ pub.dev: <https://pub.dev/>

[16] ROCHA, Ronaldo. Proposta de Definição de Uma Projeção Cartográfica para Mapeamento Sistemático em Grande Escala para o Estado do Rio Grande do Sul. 1994. 61p. Dissertação (Mestrado) - Universidade Federal do Paraná, Curitiba.

[17] VAN SOMEREN, Maarten; BARNARD, Yvonne F.; SANDBERG, J. The think aloud method: a practical approach to modelling cognitive. London: AcademicPress, v. 11, p. 29-41, 1994.