aplicativo móvel de orientação de objetos para pessoas com deficiÊncia visual

Guilherme Barth, Dalton Solano dos Reis – Orientador

Curso de Bacharel em Ciência da Computação

Departamento de Sistemas e Computação

Universidade Regional de Blumenau (FURB) – Blumenau, SC – Brasil

guibarth@furb.br, dalton@furb.br

**Resumo:** Este artigo apresenta um aplicativo iOS que utiliza o sensor LiDAR e o ML Kit da Google para fazer a identificação dos objetos de um ambiente interno e orientar o usuário de onde eles estão localizados. Foi utilizado um iPhone 13 Pro como dispositivo móvel. O projeto foi desenvolvido em Swift, utilizando o XCode v14.5 para suportar as últimas bibliotecas do LiDAR disponibilizadas pela própria Apple. Os testes mostraram que é possível fazer essa identificação de objetos com algumas limitações e o usuário não consegue se comunicar na língua nativa, apenas no inglês. Mais ...

**Palavras-chave**: Sensor LiDAR. ML Kit. Deficiência Visual.

# Introdução

Nos tempos atuais, diversos estudos e implementações vem sendo desenvolvidas a partir de uma tecnologia bastante conhecida: o laser. Um exemplo é o Light Detection and Ranging (LiDAR), que está se tornando cada vez mais popular e utilizado em diversos estudos, principalmente na área da topografia, como por exemplo, em planejamentos costeiros, florestas, urbanos e de automóveis (PEREIRA, 2022). O LiDAR é um dos termos utilizados para designar essa nova tecnologia de sensoriamento remoto, que também pode ser encontrada como Sistema de Varredura a Laser ou Perfilamento a Laser (PEREIRA, 2022). “O princípio de funcionamento do sistema de varredura laser consiste na emissão de um pulso laser de uma plataforma (aérea, terrestre ou orbital) com uma elevada frequência de repetição. O tempo de retorno dos pulsos laser entre a plataforma e os alvos é medido pelo sensor, permitindo a estimativa destas distâncias.” (PEREIRA, 2022, p. 232).

Conforme Silva (2019, p.1) “Para as pessoas sem deficiência, a tecnologia torna as coisas mais fáceis. Para as pessoas com deficiência, a tecnologia torna as coisas possíveis.”. Ainda de acordo com Silva (2019), o principal objetivo da tecnologia assistiva é proporcionar às pessoas com deficiência visual uma independência, qualidade de vida e inclusão social, por meio de melhorias na comunicação, mobilidade e autonomia no seu ambiente. Segundo Cambraia e Nazima (2021) a deficiência visual é um grave problema de saúde pública, visto que o índice de pessoas com deficiência visual aumenta conforme a população aumenta e envelhece, além de diminuir a qualidade de vida, existem dificuldades emocionais para se adaptarem e aumenta o risco de quedas e até mortes.

Pessoas com baixa visão ou cegas, precisam lançar mão de algum recurso que possibilite uma locomoção segura, sendo a bengala a opção mais conhecida e utilizada para a mobilidade, na medida em que serve para identificar obstáculos que se encontram à frente ou ao nível do chão; os chamados obstáculos rasteiros. Ocorre que no uso da bengala, por meio da técnica conhecida como varredura, o usuário a movimenta de um lado para o outro desenhando um arco à sua frente, mas precisa tocar no obstáculo para identificá-lo e somente depois desviar dele (CASTRO, 2019, p.12).

Freitas (2018) cita que a comunicação é essencial para a pessoa deficiente conseguir perceber a presença de outras pessoas, intenções de ajuda e orientação. Visando a parte técnica, existem algumas Application Programming Interface (APIs) que permitem realizar comandos vocais utilizando o Text to Speech (TTS), Speech to Text (STT) e Automatic Speech Recognition (ASR) (GOULART, 2016).

Tendo em vista a parte técnica, existe o Machine Learning Kit ou então “Kit de ML [que] é um SDK para dispositivos móveis que leva a experiência em aprendizado de máquina no dispositivo do Google a apps Android e iOS” (GOOGLE, 2021, p. 1). O Kit de ML consegue detectar até cinco objetos ao mesmo tempo em uma categoria de 400 objetos mais encontrado nas fotos (GOOGLE, 2021).

Diante disso, este trabalho apresenta um aplicativo de reconhecimento de estruturas utilizando o sensor LiDAR como protagonista do reconhecimento de superfícies, tendo como um dos seus aliados o ML Kit para auxiliar no reconhecimento de objetos, e a utilização do TTS e STT para realizar a comunicação com o usuário. Sendo o objetivo principal disponibilizar um aplicativo na plataforma iOS que auxilie pessoas com deficiência visual a identificar objetos durante o seu deslocamento utilizando o LiDAR e o ML Kit. O objetivo específico é: determinar a distância em que o objeto se encontra do dispositivo móvel.

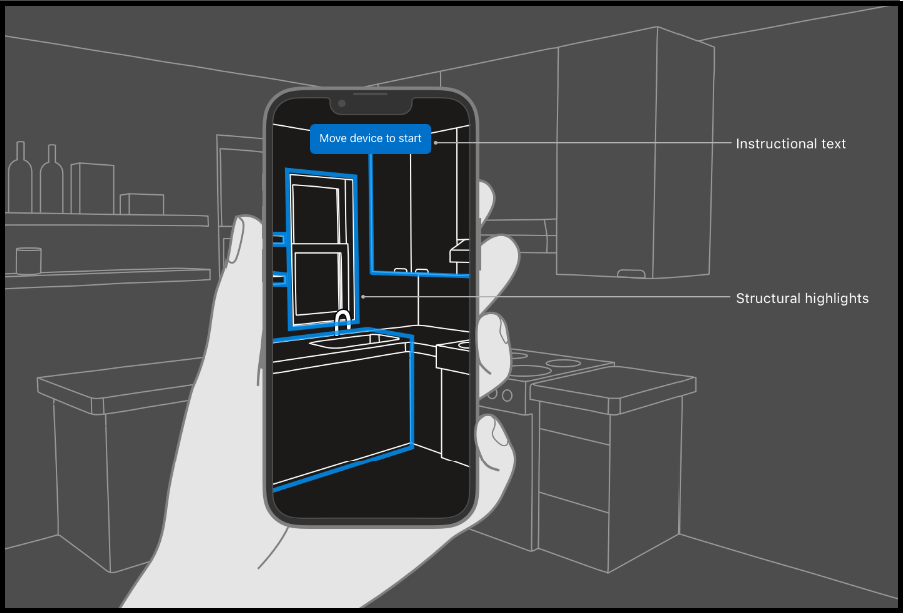
# FUNDAMENTAÇÃO TEÓRICA

Nesta seção será apresentado os conceitos para o entendimento do projeto realizado. A subseção 2.1 descreve as principais características da biblioteca utilizada pelo LiDAR (Room Plan). A subseção 2.2 aborda o ML Kit. Na subseção 2.3 é realizado um breve resumo sobre o reconhecimento de voz. Por fim, na subseção 2.4 são listados os trabalhos correlatos.

## Room PLAN

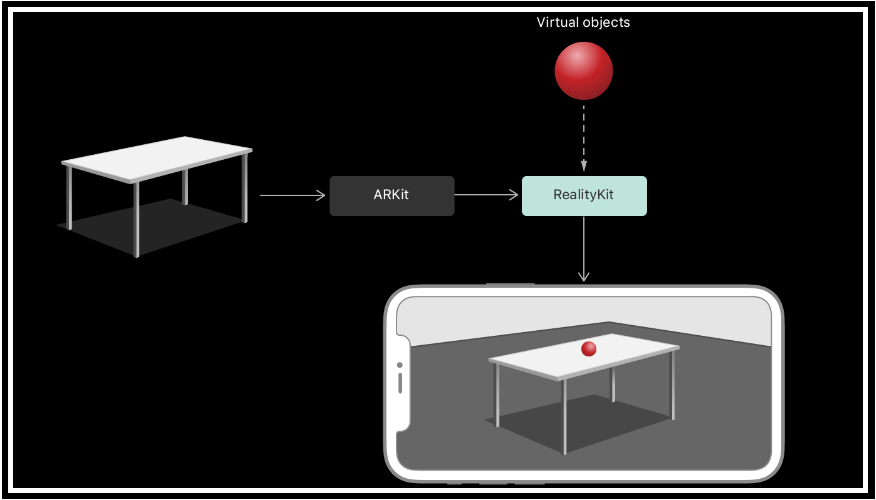
Segundo a Apple (2022c), o Room Plan é um *framework* principalmente utilizado para criar modelos 3D do interior de uma sala usufruindo dos sensores dos dispositivos móveis, além de utilizar de *Machine Learnings* treinadas e os recursos de renderização do RealityKit para capturar o ambiente físico de uma sala interna. O RealityKit é um *framework* que simula e renderiza conteúdo 3D para os seus aplicativos de realidade aumentada (APPLE, 2022b). O Room Plan consegue inspecionar a câmera e aplicando o sensor LiDAR é possível fazer identificação como paredes, portas, janelas e objetos no geral. Conforme é percorrido o interior do local do usuário é possível visualizar o resultado da sala interna em uma pequena escala 3D (APPLE, 2022c). Na Figura 1 é mostrado o resultado da inspeção do Room Plan e a Figura 2 mostra como o RealityKit consegue adicionar um objeto no mundo virtual. Mais .. sobre LiDAR?? O que tinha no projeto?

Figura 1 – Resultado de uma inspeção do Room Plan



Fonte: Apple (2022c).

Figura 2 – RealityKit

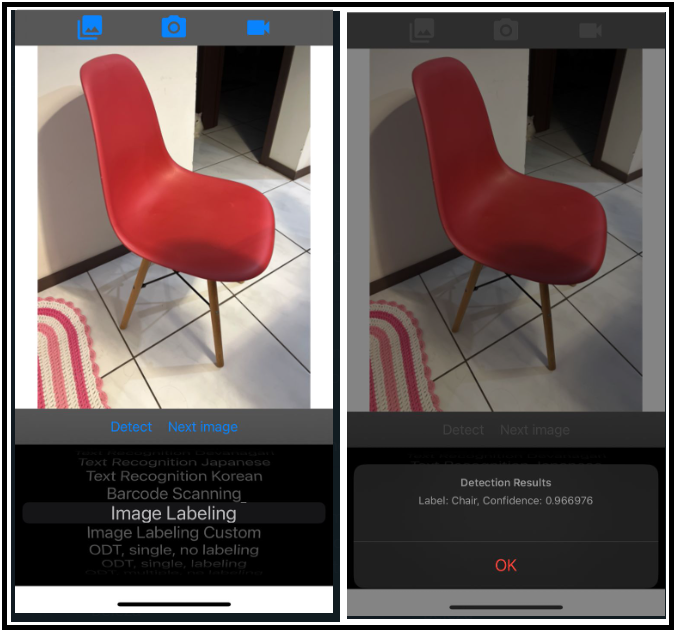


Fonte: Apple (2022b).

## ml kit

Segundo a Google (2022), o Kit de ML é um SDK que permite o usuário utilizar várias ferramentas de maneira muito prática por meio de um dispositivo móvel. Também é conhecido por ser um facilitador de resolução de desafios comuns ou conhecidos por ser um criador de experiências para os usuários. Dentre as suas funcionalidades o Kit de ML possui reconhecimento de texto, detecção facial, detecção de malha de rosto, detecção e rastreamento de objetos, reconhecimento de tinta digital, entre outros que estão recebendo atualizações (GOOGLE, 2022). Na Figura 3 é mostrado a identificação de objetos do ML Kit. Mas ...

Figura 3 – Identificação de uma cadeira a partir do ML Kit



Fonte: Digitalizado pelo autor.

## Speech to Text e Text to speech

Segundo Herchonvicz, Franco e Jasinski (2019), embora o sistema de reconhecimento de fala tenha sido aprimorado para utilização prática, alguns desafios permanecem. Por exemplo, o reconhecimento de fala em vários idiomas ainda não é resolvido devido à escassez de dados de conversação e textos com dicionários de pronúncia, a falta de convenções de linguagem e a lacuna entre a lacuna entre a tecnologia e linguagem perícia. Em contrapartida, a Apple possui a *framework* AVFoundation que, dentre as várias funcionalidades, a principal utilizada nesse projeto é a que permite transformar textos em saídas de voz, garantindo assim a interação com o usuário final (APPLE, 2022a). Já o *framework* Speech permite fazer o processo contrário, ou seja, reconhecer palavras faladas ou em áudio gravado ao vivo (APPLE, 2022d). No Quadro 1 é apresentado um trecho do código do aplicativo que realiza a conversão de texto para fala utilizando o AVFoundation. Mais ..

Quadro 1 – Conversão de voz de texto para áudio



Fonte: Elaborado pelo autor.

## TRABALHOS CORRELATOS

A seguir são apresentados três trabalhos correlatos. No Quadro 2 é apresentado o projeto de Rossi, Freitas e Reis (2019), que consiste em um mapeamento tridimensional em ambientes internos. No Quadro 3 é apresentado o trabalho de Franco e Heinzle (2018), que criaram um aplicativo para reconhecimento de objetos em imagens para pessoas portadoras de deficiência visual. Por fim, no Quadro 4 é descrito o trabalho de Pereira (2022), que compara mapas de suscetibilidade a deslizamentos com dados do LiDAR e VANT.

Quadro 2 – Mapeamento 3D de um ambiente interno

|  |  |
| --- | --- |
| Referência | Rossi, Freitas e Reis (2019) |
| Objetivos | O trabalho tem como objetivo desenvolver um protótipo robótico com o sensoriamento em 3D feito com base em um sensor LiDAR, capaz de efetuar o mapeamento em ambientes simulados e reais |
| Principais funcionalidades | Funciona com o princípio de triangulação, que é uma metodologia empregada para medir distâncias de um objeto. Para funcionar dessa maneira, o sensor emite um feixe de laser que é refletido e captado por uma lente fotossensível. Dessa forma, quando o objeto a ser mapeado está mais próximo ou mais distante, os feixes tocam em uma parte diferente da lente e pelo princípio da triangulação geométrica é estimada a posição do objeto. |
| Ferramentas de desenvolvimento | Foi utilizado o sensor LiDAR modelo YDLIDARX4 para emitir o laser. Uma placa de desenvolvimento ESP32 com antena integrada. Uma Unidade Inercial de Medida (IMU) que é composto por um Acelerômetro e um Giroscópio MPU-6050. E um Encoder Rotativo KY-040. |
| Resultados e conclusões | Os resultados obtidos mostraram que a precisão de ambos os modelos é maior que 0,70, a área sob a curva (AUC) é maior que 0,80, e o modelo gerado a partir dos dados LiDAR é mais preciso. |

Fonte: elaborado pelo autor.

Na Figura 4 são ilustrados os resultados capturados de uma varanda, que segundo Rossi, Freitas e Reis (2019), (a) é uma nuvem de pontos gerada através dos dados do sensor LiDAR com base nos movimentos de rotação e translação, (b) é um mapa desse mesmo ambiente gerado pela grade de ocupação tridimensional de uma biblioteca, chamada de OctoMap, com base na nuvem de pontos.

Figura 4 – Resultados obtidos do sensor

Tela de jogo de vídeo game

Descrição gerada automaticamente com confiança baixa

Fonte: Rossi, Freitas e Reis (2019).

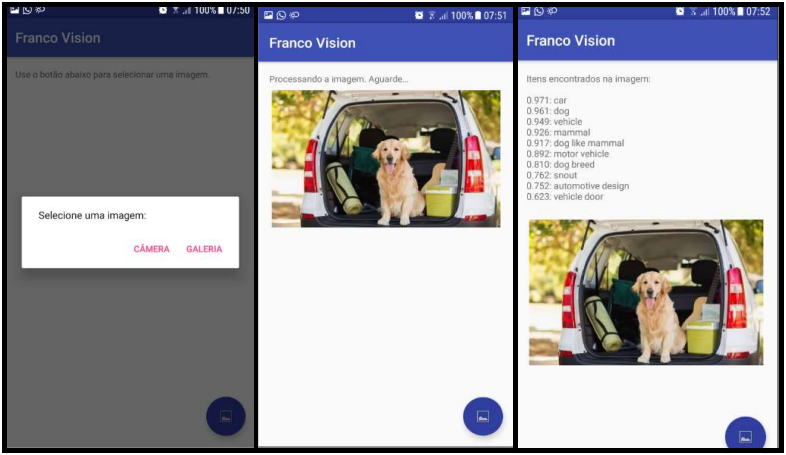
Quadro 3 – Aplicativo de reconhecimento de objetos

|  |  |
| --- | --- |
| Referência | Franco e Heinzle (2018) |
| Objetivos | O objetivo era criar um aplicativo para instruir usuários, principalmente com deficiência visual, por meio da sintetização de voz em cada etapa do seu aplicativo de reconhecimento de objetos. |
| Principais funcionalidades | Instruções de voz para orientação e navegação pelo sistema. Reconhecer um objeto de uma foto da galeria ou do *feed* da câmera. |
| Ferramentas de desenvolvimento | Cloud Text-to-Speech para a sintetização de voz. API Cloud Vision para o reconhecimento de objetos. |
| Resultados e conclusões | O aplicativo não estava 100% apto para pessoas com deficiência visual. Dependendo de alguns fatores como luz, distância e outros ruídos o resultado do reconhecimento é excelente, mas quando a distância é incrementada a sua precisão começa a declinar. |

Fonte: elaborado pelo autor.

Na Figura 5 é apresentado um breve comportamento do aplicativo de reconhecimento de imagens mencionado anteriormente, onde o usuário anexou uma imagem da galeria e a API retornou possíveis objetos que se encontram na imagem.

Figura 5 – Aplicativo de reconhecimento de imagem



Fonte: Franco e Heinzle (2018).

Quadro 4 – Comparação de dados derivados de LiDAR e VANT

|  |  |
| --- | --- |
| Referência | Pereira (2022) |
| Objetivos | O objetivo é comparar mapas de suscetibilidade a deslizamentos gerados pelo algoritmo de aprendizado de máquina floresta aleatória (Random Forest - RF) com dados de LiDAR e VANT |
| Principais funcionalidades | Utilizar o LiDAR e VANT para extrair métricas como a Precisão, Acurácia, Recall e F1-Score (medida de precisão) e comparar os dois modelos de mapeamento de suscetibilidade a deslizamentos utilizando o RF. |
| Ferramentas de desenvolvimento | Modelo Digital de Elevação, Modelos Digital de Terrenos, sensor LiDAR, Veículo Aéreo Não-Tripulado (VANT) |
| Resultados e conclusões | Os resultados obtidos mostraram que a precisão de ambos os modelos é maior que 0,70, a área sob a curva (AUC) é maior que 0,80, e o modelo gerado a partir dos dados LiDAR é mais preciso. |

Fonte: elaborado pelo autor.

Os trabalhos correlatos tiveram um papel importante principalmente na parte de entendimento do LiDAR e os possíveis problemas que podem aparecer, seja de precisão, jeito de utilização ou outro impedimento.

# DESCRIÇÃO do Aplicativo

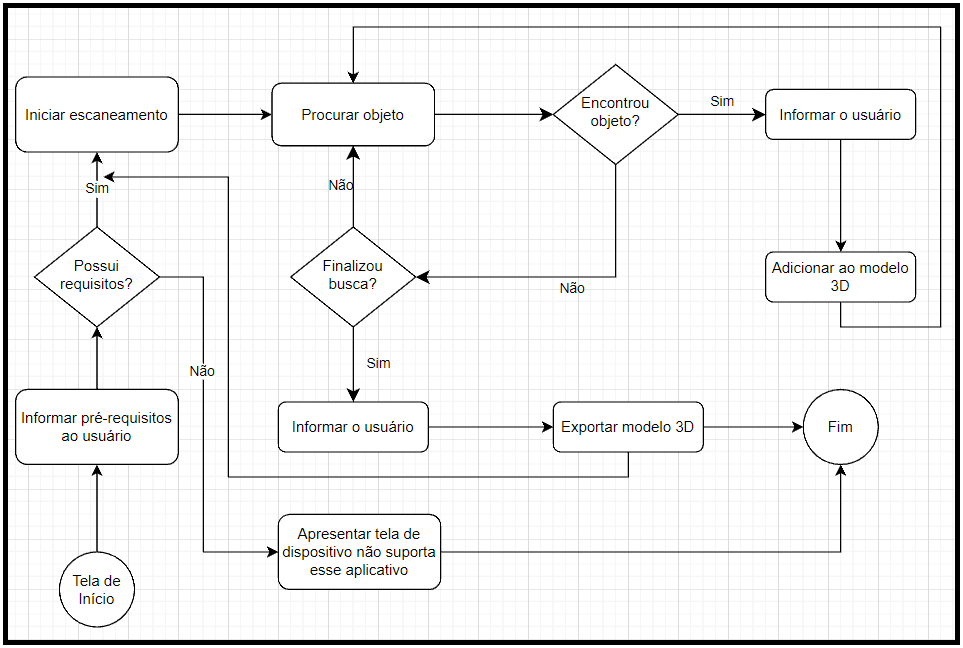
Para o desenvolvimento e teste da aplicação se utilizou o smartphone iPhone 13 Pro com o iOS 16.0, mas qualquer versão ou modelo de dispositivos móvel que seja superior e contenha o sensor LiDAR deve suportar a aplicação desenvolvida. Para realizar o desenvolvimento, foi utilizado um MacBook Pro com 8Gb de RAM, com processador i3 2.33Ghz, sem placa gráfica e um SSD.

A seguir na subseção 3.1 é apresentada a especificação, já na subseção 3.2 é abordado detalhes da implementação e funcionalidade do projeto.

## Especificação

O fluxograma e o diagrama foram construídos utilizando a ferramenta de trabalho Draw.IO, disponível gratuitamente na internet. O fluxograma de comportamento é representado na Figura 6 e se inicia informando ao usuário deficiente visual o que ele deve falar para conseguir interagir com o aplicativo e como iniciar o escaneamento. Ao sinalizar o início da busca do ambiente, o aplicativo abrirá a câmera e orienta o usuário que ele movimente o celular para cima e para baixo para uma melhora de captura do cenário atual, sempre buscando uma parede como ponto de partida. Com o escaneamento já em andamento, o aplicativo procurará novos objetos no cenário, ao encontrar um novo objeto ele informará ao usuário por comando de voz, adicionará o objeto encontrado ao modelo 3D do ambiente interno e seguirá com o escaneamento do ambiente. Se o aplicativo não encontrar nenhum objeto novo após alguns segundos, o escaneamento é finalizado e apresentado a tela que permite a exportação do modelo 3D gerado. O usuário pode finalizar o escaneamento a qualquer momento que ele desejar. Na tela de exportação do modelo 3D é possível voltar ao início e recomeçar o escaneamento do ambiente. Se ao iniciar o aplicativo o usuário não possuir o sensor LiDAR, será apresentado uma tela de aviso informando ao usuário que ele não consegue executar o aplicativo.

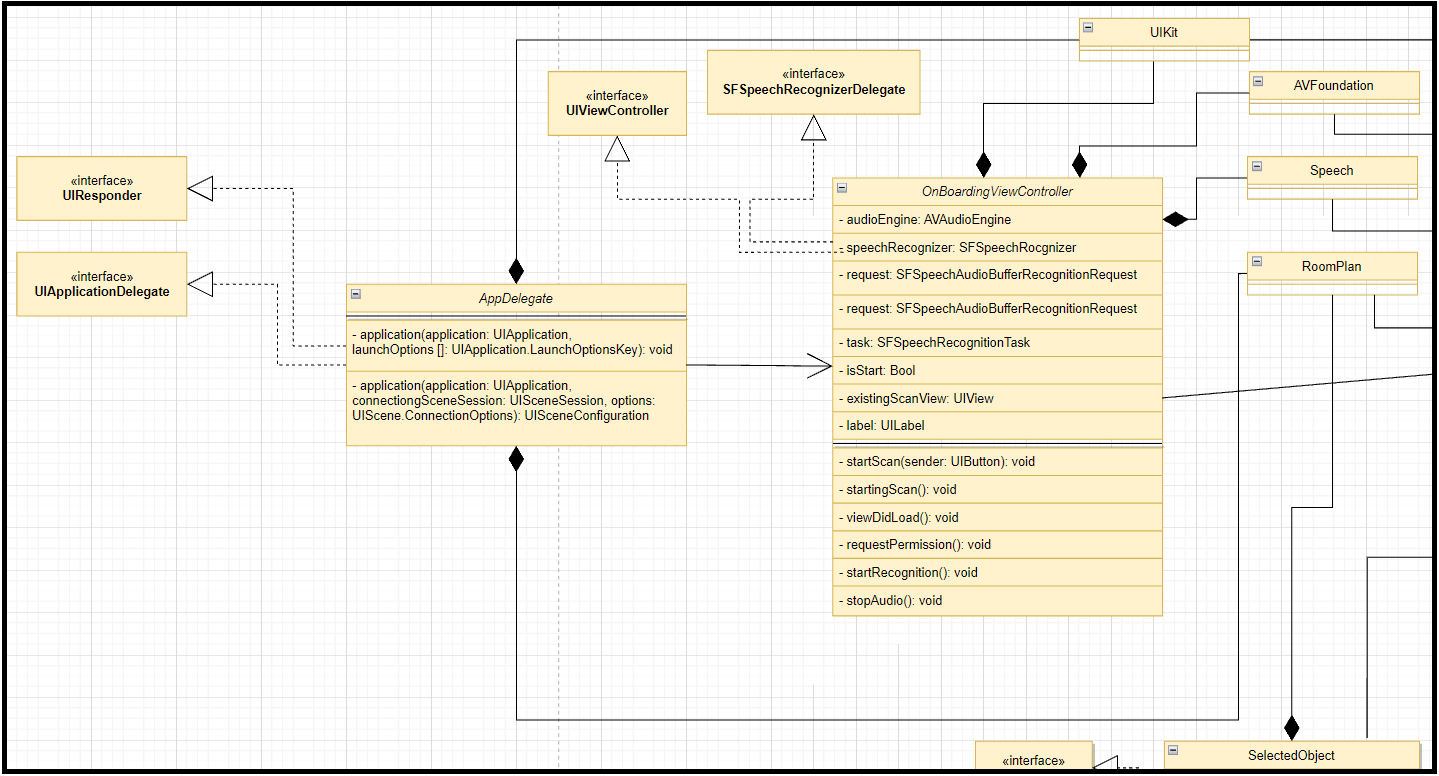
Figura 6 – Fluxograma de telas do aplicativo



Fonte: Elaborado pelo autor.

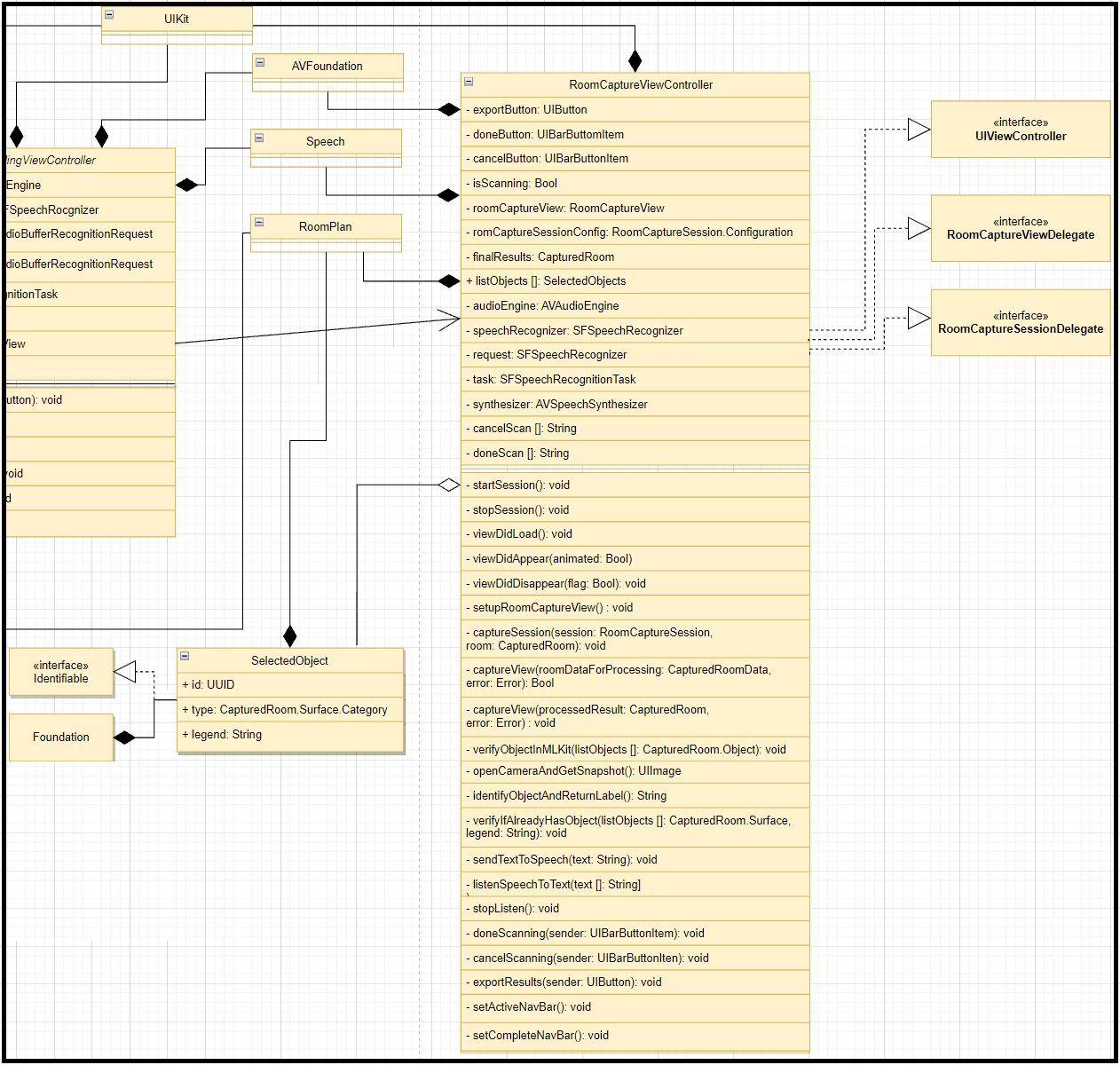
Na Figura 7 e Figura 8 é apresentado o Diagrama de Classes do aplicativo, onde mostra todas as classes, variáveis, métodos, agregações, composições e implementações de interface. As figuras foram divididas para se obter uma melhor compreensão e visualização do diagrama.

Figura 7 – Diagrama parcial parte 1



Fonte: Elaborado pelo autor.

Figura 8 – Diagrama parcial parte 2



Fonte: Elaborado pelo autor.

Precisa explicar o conteúdo das figuras 7 e 8.

## implementação

Esta seção apresenta o desenvolvimento do aplicativo de orientação de pessoas com deficiência visual em ambientes fechados. O Quadro 5 apresenta os Requisitos Funcionais (RF), já o Quadro 6 apresenta os Requisitos Não Funcionais (RNF).

Quadro 5 – Requisitos funcionais

|  |
| --- |
| **Requisitos funcionais** |
| RF01: o aplicativo deverá informar ao usuário quando há um novo obstáculo no visor da câmera |
| RF02: o aplicativo deverá permitir interação vocal |
| RF03: o aplicativo deverá criar uma planta em 3D do ambiente escaneado |
| RF04: o aplicativo deverá permitir exportar o modelo 3D do ambiente |

Fonte: elaborado pelo autor.

Quadro 6 – Requisitos não funcionais

|  |
| --- |
| **Requisitos não funcionais** |
| RNF01: a detecção principal de objetos deve ser feita com a *framework* RoomPlan, quando não for possível, deve utilizar o ML Kit |
| RNF02: ao detectar um obstáculo, o aplicativo deve informar qual é o objeto encontrado |

Fonte: elaborado pelo autor.

Para a implementação do aplicativo de orientação de pessoas com deficiência visual foi utilizado a linguagem de programação Swift, utilizando o XCode 14.0.1 e a versão compatível do MacOS 12.6. Os *frameworks* utilizados estão em constantes atualizações, porém elas não possuem um controle de versionamento. O desenvolvimento foi dividido em 3 partes, 1) Inicialização do escaneamento do LiDAR; 2) Identificação dos objetos; 3) conversação com o aplicativo em todos os momentos.

Ao iniciar o aplicativo é necessário aprovar diversas permissões para que seja permitido o uso da câmera, microfone e alto falante. A inicialização do escaneamento é feita por meio da biblioteca Room Plan (ver subseção 2.1) com um evento privado nomeado StartSession, e após o disparo desse evento o LiDAR começa a funcionar. À medida que o LiDAR identifica novos objetos ele dispara um outro evento informando os objetos já identificados, e esse evento é chamado de captureSessioncom uma sobreposição específica que é o didAdd*.* Nesse captureSession é possível ver quais são todos os objetos já capturados pelo LiDAR, além disso o Room Plan faz uma triagem do objeto encontrado. A triagem é dividida em 5 grupos, sendo eles, portas, paredes, aberturas, janelas e objetos. No Quadro 7 é ilustrado o código do aplicativo onde é mencionado o evento do captureSession e os grupos de objetos recebidos.

Quadro 7 – Utilizando a biblioteca Room Plan



Fonte: Elaborado pelo autor.

No Quadro 7, o objeto retornado room possui vários vetores de Surface ou Object. Um Object ou Surface possui algumas propriedades como identificador (UUID), categoria, precisão, entre outras que não foram utilizados. Não é possível determinar a distância que o objeto se encontra da câmera, porque infortunadamente a *framework* não disponibiliza essa propriedade.

Como o foco principal do Room Plan não é informar quais são os objetos encontrados no ambiente ele acaba generalizando apenas para objeto, e nesses casos se optou em utilizar o ML Kit (ver subseção 2.2). No escopo atual o Room Plan não consegue retornar a imagem do *feed* da câmera como um objeto utilizável, no caso o CVPixelBuffer, então é necessário utilizar a própria câmera para tirar uma foto desse instante que é encontrado o objeto e encaminhar essa imagem para o ML Kit para realizar a sua identificação. Após receber uma imagem da câmera no formato UImage ela é encaminhada para o ML Kit para realizar a identificação do objeto e a sua precisão, com o resultado da identificação, é informado ao usuário final em comando de voz. Na Quadro 8 é apresentado o trecho do código onde é recebido a UImage, enviada para identificação e recebendo o resultado da identificação.

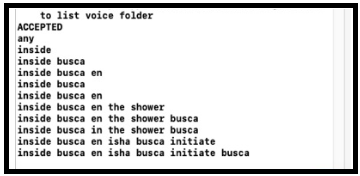
Quadro 8 – Trecho de código ML Kit



Fonte: Elaborado pelo autor.

Todos os botões do aplicativo podem ser interagidos por comandos de voz utilizando os *frameworks* Speech e AVFoundation. Esse procedimento ficou demasiado simples com um pequeno impedimento, no Speech é possível determinar qual a língua a ser falada, já o AVFoundation não permite isso de maneira tão simplificada, onde o padrão é o inglês. Não somente isso, o AVFoundation entende diversas línguas ao mesmo tempo, ou seja, ele entende outros idiomas e escreve corretamente no seu *Buffer* de resposta a palavra em português, espanhol e afins, mas a língua que possui a maior precisão de entendimento é o inglês. Em função da língua predominante ser o inglês, ao entender uma palavra em outro idioma o AVFoundation efetua a tradução para o inglês, resultando em uma dificuldade para extrair o idioma falado pelo usuário. No Quadro 9 é ilustrado o comportamento do AVFoundation onde o usuário fala o comando Iniciar busca, como resultado a palavra iniciar é traduzida para o inglês da forma que é compreendida resultando em *inside* e *initiate*, já a palavra busca é permanecida em português.

Quadro 9 – Retorno do Buffer do microfone



Fonte: Elaborado pelo autor.

# RESULTADOS

Esta seção mostra os testes da arquitetura, desafios e resultados. Na subseção 4.1 é detalhado os desafios encontrados no desenvolvimento do projeto, juntamente com os testes realizados e na subseção 4.2 é apresentado os resultados obtidos.

## Testes e desafios

Para facilitar a utilização do sensor LiDAR o desenvolvimento se iniciou com a biblioteca Room Plan desenvolvida em Swift. Ao mesmo tempo que ela facilita o acesso aos resultados em função da sua triagem, ela não permite determinar a distância entre o objeto e a câmera, causando um impedimento nessa métrica. Sem a distância exata do objeto no momento de informar o usuário sobre o objeto fica relativamente vago aonde o objeto se encontra, podendo atrapalhar a orientação do usuário.

O segundo desafio foi encontrar uma Machine Learning que fazia integração com o iOS de maneira facilitada, isso porque os modelos de ML já existentes da Apple, o CoreModel e o CoreVision não atendiam ao escopo de objetos de uma casa. O ML Kit da Google atende muito bem aos requisitos de um ambiente interno, contudo as configurações necessárias para executá-lo possuem conflito direto com as configurações do Room Plan. Como consequência desse conflito, ocasionou um alto consumo de tempo para consertar esse conflito de configuração e gerou uma instiga por buscar uma outra ML.

O aplicativo foi testado com duas pessoas que não possuem deficiência visual, para tentar ser mais fidedigno ao público-alvo foi colocado uma venda sob os olhos dos usuários. Os usuários relataram que eles conseguiam interagir com o aplicativo totalmente apenas por comandos de voz, mas constataram que em português ele não compreende corretamente, a assertividade é maior quando é falado em inglês. As pessoas comentaram que ao encontrar um objeto, ficava à espera dele tentando encontrá-lo fazendo uma predição de onde ele encontrava-se. Por não conseguir pegar a localização exata do objeto, a informação é apenas jogada para ele, deixando o usuário perdido de onde está o objeto, mas avisando-o. Ao finalizar o escaneamento, o usuário com deficiência visual não consegue interagir com o modelo criado em 3D, sendo inutilizado pelo público-alvo, mas pode agregar para uma futura extensão desse projeto.

Uma constatação mencionada por uma cobaia foi “Quando eu encontro um objeto pela primeira vez, o aplicativo me informa a respeito dele, mas quando eu vou para um segundo objeto e volto no anterior, o aplicativo não informa que há um objeto na minha frente”. Essa foi uma falha de rotina encontrada pelo usuário final e que pode ser ajustada em uma extensão desse projeto.

Mais detalhes dos testes feitos, qual era o perfil dos avaliadores? Onde se testou, cenário, etc.

Precisaria ter dados mais precisos para avaliar e gerar resultados...

## Resultados obtidos

??..

# CONCLUSÕES

Ao efetuar os testes foi visto que o aplicativo atingiu o objetivo principal que é informar o usuário que há um obstáculo no seu campo de visão e que ele está próximo, porém o objetivo específico de determinar a distância do objeto não foi alcançado devido as limitações da biblioteca utilizada. A comunicação do aplicativo com o usuário foi excepcional, permitindo que ele saiba exatamente em qual tela que ele se encontra e informando em todos os momentos que um objeto novo é encontrado.

Apesar dos resultados obtidos, durante a utilização do ML Kit há vários fatores que influenciam no seu resultado, sendo os principais a iluminação e se o objeto encontrado não é o objeto principal do cenário, causando assim uma pequena diferenciação de qual objeto ele informa ao usuário. Considerando que o STT é uma tecnologia que existe há um determinado tempo, ela ainda não está adequada para todos os idiomas, funcionando bem apenas para o inglês. Com isso, esse aplicativo não pode ser utilizado em situações reais por deixar o usuário sem noção exata de onde o objeto se encontra, entretanto, este aplicativo pode engajar trabalhos futuros nessa linha de pesquisa. As possíveis propostas de extensão são: utilizar o modelo 3D para alguma funcionalidade como a localização *indoor* utilizando Beacons; melhorar a utilização do LiDAR com o objetivo de trazer a localização exata do objeto; e utilizar outras bibliotecas para o STT permitindo que o usuário se comunique apenas com o português.

Referências

APPLE. **AVFoundation**. [2022a] Disponível em: https://developer.apple.com/documentation/avfoundation. Acesso em: 23 nov. 2022.

APPLE. **RealityKit**. [2022b]. Disponível em: https://developer.apple.com/documentation/realitykit. Acesso em: 23 nov. 2022.

APPLE. **RoomPlan**. [2022c]. Disponível em: https://developer.apple.com/documentation/roomplan. Acesso em: 04 nov. 2022.

APPLE. **Speech**. [2022d]. Disponível em: https://developer.apple.com/documentation/speech/. Acesso em: 10 nov. 2022.

CAMBRAIA, Maria Izabel de Albuquerque; NAZIMA, Maira Tiyomi Sacata Tongu. Avaliação dos sintomas de depressão em pessoas com deficiência visual.**Rev. SBPH**, São Paulo,  v. 24, n. 1, p. 79-90, jun.  2021.   Disponível em http://pepsic.bvsalud.org/scielo.php?script=sci\_arttext&pid=S1516-08582021000100008&lng=pt&nrm=iso. Acesso em: 24 nov. 2022.

CASTRO, Oliveiros Barone. **Relações entre percepção auditiva e orientação e mobilidade em um grupo de pessoas com deficiência visual usuárias de cão**. 2019. 117 f. Dissertação (Mestrado) - Curso de Fonoaudiologia, Pontifícia Universidade Católica de São Paulo, São Paulo, 2019. Disponível em: https://tede.pucsp.br/bitstream/handle/22132/2/Oliveiros%20Barone%20Castro.pdf. Acesso em: 24 nov. 2022.

FREITAS, Daniela Gonçalves da Silveira. **Orientações para interação com deficientes visuais e auditivos**. IFBA - Instituto Federal de Educação, Ciência e Tecnologia da Bahia, [S. L.], 2018. Disponível em: https://portal.ifba.edu.br/conquista/napnee/paginas-links/orientacoes-para-interacao-com-deficientes-visuais-e-auditivos#wrapper. Acesso em: 23 nov. 2021.

GIONGO, M.; KOEHLER, H. S.; MACHADO, S. do A.; KIRCHNER, F. F.; MARCHETTI, M. LiDAR: princípios e aplicações florestais. **Pesquisa Florestal Brasileira**, *[S. l.]*, v. 30, n. 63, p. 231, 2010. DOI: 10.4336/2010.pfb.30.63.231. Disponível em: https://pfb.cnpf.embrapa.br/pfb/index.php/pfb/article/view/148. Acesso em: 24 nov. 2022.

GOULART, Gielez Feldhaus. **Aplicativo para auxiliar crianças autistas no desenvolvimento e aquisição da linguagem**. 2016. 50 f. TCC (Graduação) - Curso de Ciência da Computação, Centro de Ciências Exatas e Naturais, Universidade Regional de Blumenau, Blumenau, 2016.

HERCHONVICZ, Andrey L.; FRANCO, Cristiano R.; JASINSKI, Marcio G.. A comparison of cloud-based speech recognition engines. In: COMPUTER ON THE BEACH, 10., 2019, Florianópolis. **Anais [...].**Florianópolis: Computer On The Beach, 2019. p. 366-375. Disponível em: https://periodicos.univali.br/index.php/acotb/article/view/14332. Acesso em: 24 nov. 2022.

PEREIRA, Felicia França. **Comparação de dados derivados de LiDAR e VANT para o mapeamento de suscetibilidade de deslizamento usando o algoritmo Random Forest**. 2022. 21 f. Dissertação (Mestrado) - Curso de Desastres Naturais, O Instituto de Ciência e Tecnologia, Universidade Estadual Paulista (Unesp), São José dos Campos, 2022. Disponível em: https://repositorio.unesp.br/handle/11449/237072. Acesso em: 18 nov. 2022.

ROSSI, Túlio Xavier; FREITAS, Elias José de Rezende; REIS, Agnaldo José da Rocha. **Mapeamento Tridimensional de Ambientes Internos Utilizando um Sensor LIDAR**. 2019. 62 f. Monografia (Especialização) - Curso de Engenharia de Controle e Automação, Cecau, Universidade Federal de Ouro Preto -Ufop, Ouro Preto, 2019. Disponível em: https://monografias.ufop.br/handle/35400000/2439. Acesso em: 24 set. 2021.

SILVA, William Lopes da. **Black Glasses**: assistente para deficientes visuais via geolocalização. 2019. 20 f. TCC (Graduação) - Curso de Bacharel em Ciência da Computação, Universidade Regional de Blumenau (FURB), Blumenau, 2019. Disponível em: http://dsc.inf.furb.br/arquivos/tccs/monografias/2019\_2\_william-lopes-da-silva\_monografia.pdf. Acesso em: 22 jun. 2022.

Citações do texto não encontradas nas referências.

Franco e Heinzle (2018

GOOGLE, 2022

GOOGLE, 2021

GOULART, 2016