

# DESENVOLVIMENTO DE UMA BIBLIOTECA PARA O USO DO SENSOR LIDAR EM DISPOSITIVOS IOS

Gabriel Luís Fernando Vieira de Souza, 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  
glfsouza@furb.br, dalton@furb.br

**Resumo:** *O resumo é uma apresentação concisa dos pontos relevantes de um texto. Informa suficientemente ao leitor, para que este possa decidir sobre a conveniência da leitura do texto inteiro. Deve conter OBRIGATORIAMENTE o OBJETIVO, METODOLOGIA, RESULTADOS e CONCLUSÕES. O resumo não deve ultrapassar 10 linhas e deve ser composto de uma sequência corrente de frases concisas e não de uma enumeração de tópicos. O resumo deve ser escrito em um único texto corrido (sem parágrafos). Deve-se usar a terceira pessoa do singular. As palavras-chave, a seguir, são separadas por ponto, com a primeira letra maiúscula. Caso uma palavra-chave seja composta por mais de uma palavra, somente a primeira deve ser escrita com letra maiúscula, sendo que as demais iniciam com letra minúscula, desde que não sejam nomes próprios.]*

**Palavras-chave:** *Ciência da computação. Sistemas de informação. Monografia. Resumo. Formato.*

## 1 INTRODUÇÃO

Em um mundo globalizado e digitalmente conectado nos deparamos com possibilidades que nos permitem registrar um momento para que ele seja recordado posteriormente de maneira cada vez mais tecnológica. Diariamente se fotografa ou se filma cenas, seja para compartilhar uma viagem em uma rede social, guardar uma recordação especial ou até mesmo registrar um quadro na sala de aula para não esquecer o seu conteúdo. Com essa premissa, as empresas de tecnologia para dispositivos móveis estão constantemente inovando na forma como utilizamos ferramentas de filmagem e fotografia, introduzindo novas funcionalidades, que vão além de poder utilizar *zoom* nos registros dos momentos, até a possibilidade de efeito panorâmico na captura das imagens. Uma dessas inovações, anunciada primeiramente no iPad Pro da Apple, em 2020, e, posteriormente no iPhone 12 Pro, no mesmo ano, foi o LiDAR, sigla para Light Distance And Ranging. O LiDAR consiste em “um método de sensoriamento ativo que pode precisamente medir distâncias, transmitindo energia laser e analisando a energia retornada” (BAUWENS *et al.*, 2016 p. 2, tradução nossa).

Historicamente esta tecnologia já vem sendo utilizada antes mesmo do interesse da Apple, principalmente na área florestal. Giongo *et al.* (2010) explicam que o LiDAR tem muitas vantagens para fazer uma digitalização em comparação com imagens de satélite para um mapeamento florestal, visto que não depende da luz solar, fazendo assim que seja mais fácil ignorar sombras ocasionadas pelas nuvens, assim como seus feixes de laser que são capazes de penetrar as copas das árvores facilitando na descoberta do relevo do terreno nas florestas. Outra aplicação foi utilizada com o sensor Zephyr LiDAR, no ano de 2017, para construir um medidor de velocidade do vento para usinas eólicas pelo pesquisador Nassif, que enfatizou que o “LiDAR é um sistema confiável, robusto e a custos relativamente baixos” (NASSIF, 2017, p. 74), que favorece a adoção nas mais diversas áreas, destacando a performance e o custo-benefício da tecnologia. Para dispositivos móveis, a principal funcionalidade do sistema é perceber profundidade através da distância em um ambiente e mapear o relevo de objetos.

O LiDAR nos dispositivos Apple entra para agregar novas funcionalidades no ARKit, o framework de Realidade Aumentada (ou Augmented Reality – AR) do iOS. O ARKit se propõe a adicionar objetos virtuais 3D ou 2D através da visualização da câmera do dispositivo de uma forma que pareça que esses objetos estejam interagindo com o mundo real como exalta Apple (2022b). Uma dessas funcionalidades adicionadas no ARKit com o lançamento do LiDAR nos dispositivos da Apple foi a Depth API, que faz possível a captura da profundidade da cena pixel a pixel combinado com uma mesh 3D, gerando um mapeamento que pode ser utilizado para gerar medidas mais precisas de um ambiente ou na geração de um Scene Geometry, que consiste em um mapa topológico do mundo real com categorizações de objetos também reais para serem interageis em uma aplicação que utiliza o ARKit.

### [Falta digitalização]

Com o contexto descrito acima, este estudo se propõe a entender e desenvolver uma biblioteca que irá digitalizar objetos reais em objetos virtuais, utilizando as tecnologias do ARKit. É importante destacar que o presente estudo dispõe de diversos termos técnicos ao longo de seu desenvolvimento e que seus significados não são destacados diretamente, pois entende-se que o estudo se dirige à um contexto de estudantes e profissionais técnicos com conhecimentos prévios dos termos em questão.

Assim sendo, será realizada uma revisão bibliográfica consultando a documentação do ARKit e artigos sobre o assunto como base de estudo, bem como serão testadas as capacidades do dispositivo de fazer a digitalização de forma escalável contemplando diversos cenários para que sejam identificados os possíveis problemas em um uso cotidiano.

## 1.1 OBJETIVOS

O objetivo geral desse trabalho é criar uma biblioteca que permita que uma aplicação digitalize objetos reais em objetos que possam ser manipulados em um ambiente 3D virtual utilizando LiDAR dos dispositivos iPad Pro e iPhone 12 Pro da Apple.

Os objetivos específicos são:

- permitir o desenvolvimento de aplicações com digitalização 3D;
- construir abstrações à tecnologia LiDAR para a digitalização 3D com as bibliotecas do ARKit.

## 2 FUNDAMENTAÇÃO TEÓRICA

LiDAR, é uma tecnologia promissora que com sua adição em dispositivos móveis se torna mais atraente ao consumidor final. Na área florestal, como indagam Bauwens *et al.* (2016), já é estudada a possibilidade da utilização de sensores LiDAR portáteis, pois a metodologia de mapeamento aéreo não é capaz de destacar todos os elementos das florestas, onde um mapeamento de campo poderia resultar em um detalhamento mais abrangente das árvores. Com a chegada desse sensor nos dispositivos Apple, caso a tecnologia se prove eficiente, esta pesquisa poderia ser feita até mesmo por um aparelho celular.

Além da área florestal, a digitalização tridimensional é muito útil para preservação histórica. No trabalho de Gonzo *et al.* (2007) foram utilizados diversos métodos de modelagem 3D com o intuito de catalogar monumentos históricos na Itália. Dentre eles é elencado a modelagem baseada em imagem e a digitalização a laser por alcance. A modelagem baseada em imagem se trata de uma técnica que através de uma imagem estática, modela-se um objeto tridimensional que se assemelhe ao real e, posteriormente, mapeia-se a imagem como textura, utilizando os limites geométricos e sombras das imagens como ponto de referência para montar relevo. Na Figura 7, pode ser observado na esquerda uma foto real do Castel Valer na Itália e a direita uma modelagem 3D do mesmo.

Figura 7 – Castel Valer Real x modelagem 3D



Fonte: Gonzo *et al.* (2007).

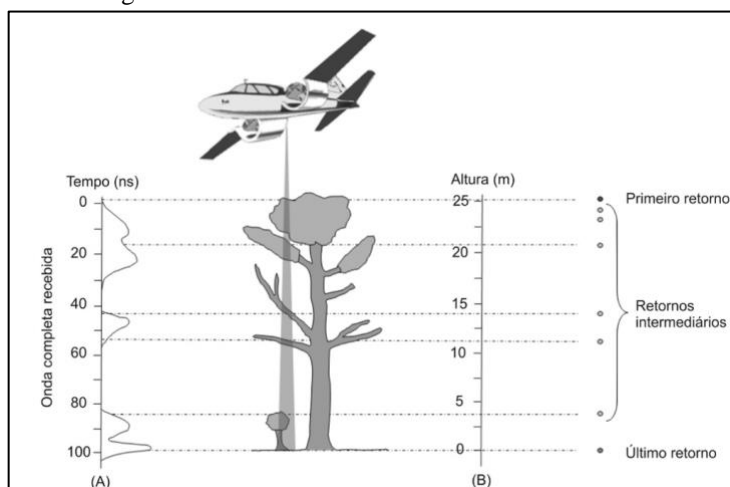
Embora seu custo seja reduzido em comparação com outros métodos, essa digitalização necessita que a câmera esteja nas condições perfeitas que, segundo os autores, são consideravelmente difíceis de atingir e o tempo despendido para fazer essa modelagem é consideravelmente alto. Outro método de modelagem é a digitalização a laser por alcance, também conhecido como LiDAR. Utilizando essa forma de digitalização o processamento é rápido e capaz de detalhar ambientes complexos sem muito esforço. O seu ponto negativo é que pelo fato de mapear apenas as distancias para um objeto 3D, ele sozinho precisaria de muita edição para se mostrar útil. Logo, a sugestão levantada por Gonzo *et al.* (2007) é a de combinar os dois métodos, o fotográfico para a texturização do ambiente e o laser para a criação das modelagens, que é utilizado neste estudo.

### 2.1 TECNOLOGIA LIDAR

Nesta seção iremos elencar formas de escaneamento utilizando LiDAR dentre eles o escaneamento aéreo, o escaneamento fixo e o escaneamento móvel. O escaneamento aéreo, como já foi trazido anteriormente, é utilizado, principalmente, na engenharia florestal para escaneamento de florestas. Como explica Muhadi (2020), se trata de um veículo aéreo que pode ser um helicóptero ou uma aeronave de asa fixa que carrega sensores e dentre eles o sensor LiDAR, que emite um feixe de laser em direção ao chão mapeando o que encontrar e o que for retornado ao sensor é gravado, em que, por exemplo, o tempo de viagem do laser é utilizado para medir a distância até o objeto. Na Figura 2 podemos ver um exemplo desse escaneamento que é utilizado na área florestal para o mapeamento do solo, onde pode-se observar que se trata de um processo em que o sensor, quando envia o feixe de luz para baixo, retorna informação da distância em metros calculada (Altura (m) na imagem) a cada n nano segundos (Tempo (ns) na imagem) que pode ser observado nas

réguas a esquerda e a direita respectivamente. Cada linha pontilhada na horizontal representa os retornos de informação que o lidar traz, concluindo que para medir uma altura de 25 metros o sensor LiDAR utilizado pelos autores Giungio *et. al.* (2010) demoraria 100 nano segundos.

Figura 1 - Escaneamento Aéreo utilizando LiDAR.



Fonte: Giungio *et. al.* (2010).

Nassif (2017) exemplifica o sensor fixo por meio de seu trabalho de utilização para medição de velocidade do vento, utilizando um sensor que “emite raio laser para o centro de um espelho que fica rotacionando e reflete o raio laser” (MUHADI *et. al.*, 2020) mapeando o ambiente. Por fim, o móvel, como enfatiza Muhadi (2020), é muito parecido com o escaneamento aéreo, pois digitaliza apenas uma direção e o portador do sensor se move para mapear o ambiente que geralmente é combinado com um sensor de Global Position System (GPS), responsável por fazer as correções necessárias devido o movimento. No caso dos dispositivos Apple, a forma utilizada é o sensor móvel e como é mencionado por Apple (2022b), esse sensor em conjunto com o ARKit ajuda a digitalizar formas do mundo real transformando as coordenadas da distância em vértices que formam um *mash* a partir dos polígonos formados com a ligação desses vértices.

## 2.2 ARKIT

ARKit se trata de um framework que contempla algumas abstrações para trabalhar com realidade aumentada nos dispositivos iOS, que mescla a utilização da visualização da câmera com os outros sensores do dispositivo. O ARKit é capaz de prover uma série de Application Programming Interfaces (APIs) para se trabalhar com realidade aumentada, dentre elas estão a Depth API, a Scene Geometry e a Instant AR, que se destacam no uso para digitalização de objetos com utilização do LiDAR, introduzidas no ARKit 4 e aprimoradas no ARKit 5. A combinação de ambas são capazes de gerar uma cena e entender cada objeto dessa cena.

Depth API, como é especificado por Apple (2022b), tem como premissa, utilizar os dados de profundidade do ambiente gerados por pixel pelo LiDAR para gerar *mashes* (polígonos 3D) que são interpretados pelo Scene Geometry, permitindo entender o que cada objeto na cena representa. Além disso, o Geometry Scene também é capaz de aplicar física do mundo real em objetos virtuais fazendo aplicações de realidade aumentada mais imersivas. Por último, é relacionado também ao DepthAPI ao InstantAR, que é capaz de identificar superfícies planas e ancoras na cena quase instantaneamente, fazendo com que a digitalização seja muito mais fluida. As três APIs combinadas com auxílio do framework como um todo são capazes de criar cenas complexas de realidade aumentada com oclusão de objetos virtuais (quando um objeto virtual fica atrás de um objeto real) e com a adição do LiDAR é possível mapear cenas inteiras e de transformá-las em objetos 3D, além da capacidade de projetar objetos.

## 2.3 TRABALHOS CORRELATOS

Nessa seção são apresentados dois trabalhos acadêmicos e uma descrição de funcionalidade de um aplicativo que discursa sobre digitalização tridimensional, trazendo semelhanças em outras áreas acadêmicas com o presente escrito. Serão apresentados por meio da apresentação e três quadros comparativos que contribuirão para uma compreensão mais objetiva de cada trabalho correlato.

O Quadro 1 apresenta um artigo que compara os sensores de profundidade (LiDAR e TrueDepth) da Apple com soluções industriais para digitalização de objetos 3D (VOGT; RIPS; EMMELMANN, 2021). O quadro 2 traz outro artigo que aborda uma digitalização de objetos tridimensionais utilizando o Kinect V1 e uma câmera digital de alta resolução (LOURA *et al.*, 2018). Por fim, no quadro 3 é evidenciada a descrição de funcionamento de um aplicativo comercial chamado Polycam, que faz a digitalização de objetos tridimensionais utilizando o LiDAR dos dispositivos Apple e disponibiliza objetos virtuais 3D (POLYCAM, 2021).

Quadro 1 – Comparison of iPad Pro®'s LiDAR and TrueDepth Capabilities with an Industrial 3D Scanning Solution

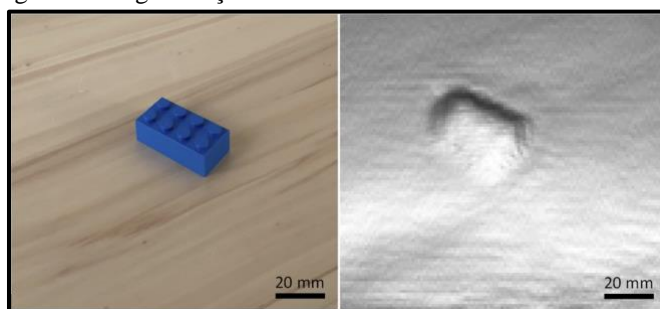
Referência	Vogt, Rips e Emmelmann (2021).
Objetivos	Avaliar a capacidade de digitalização 3D dos sensores LiDAR e True Depth se comparados com soluções industriais.
Principais funcionalidades	Digitalização 3D de objetos usando LiDAR e True Depth.
Ferramentas de desenvolvimento	Aplicativo Hedges 3D com MeshLab para tratamento do objeto digitalizado.
Resultados e conclusões	que o LiDAR ainda não está pronto para o consumidor final utilizar como uma ferramenta de digitalização 3D para objetos pequenos devido a sua baixa resolução.

Fonte: elaborado pelo autor.

Nesse primeiro trabalho correlato, apresentado através do artigo desenvolvido por Vogt, Rips e Emmelmann (2021) é possível descobrir se as tecnologias dos dispositivos Apple se comparam com ferramentas comerciais já consolidadas no mercado. Sua premissa é medir a real capacidade da digitalização tridimensional com as tecnologias LiDAR e TrueDepth do iPad Pro 2020, a partir da digitalização de blocos de lego de diferentes cores e tamanhos.

Para isso, foram criados diversos cenários e observado como cada tecnologia se comporta. A tecnologia de mercado que foi contemplada é a Artec Space Spider com Blue Light Technology, que se trata de uma ferramenta muito comum para digitalização 3D de objetos (VOGT; RIPS; EMMELMANN, 2021). Do lado da Apple foi utilizado apenas o TrueDepth para digitalização, visto que o LiDAR não consegue extrair características em objetos pequenos. Na Figura 1 pode ser observado a esquerda uma peça de lego, objeto real de dimensão pequena, que quando digitalizado usando o lidar perde maior parte de suas características, como pode ser observado na imagem em tons de cinza a direita.

Figura 1 – Digitalização usando sensor LiDAR do iPad Pro 2020



Fonte: Vogt, Rips e Emmelmann (2021).

Por fim, os autores entendem que a tecnologia ainda não está pronta para o consumidor final utilizar como uma ferramenta de digitalização 3D para objetos pequenos, devido a sua baixa resolução.

O segundo trabalho correlato estudado foi de Loura *et al.* (2018), conforme apresentado resumidamente no Quadro 2.

Quadro 2 – Reconstrução 3D de Objetos com Kinect e Câmera Digital

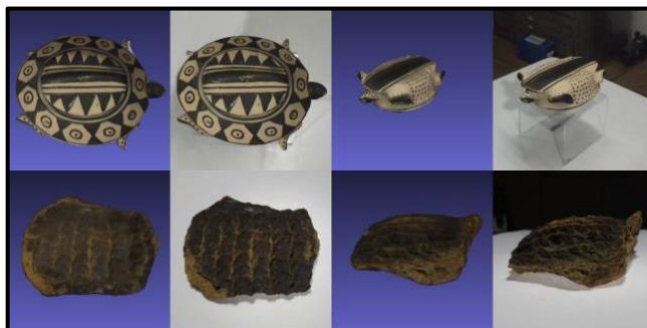
Referência	Loura <i>et al.</i> (2018).
Objetivos	Realizar a digitalização 3D para preservação histórica de artigos de museu utilizando um Kinect e uma câmera digital.
Principais funcionalidades	A digitalização 3D usando o sensor Kinect e uma câmera digital.
Ferramentas de desenvolvimento	Um software próprio que utiliza Open CV, o Kinect Fusion da própria Microsoft e o driver open-source OpenKinect.
Resultados e conclusões	O resultado para os autores foi acima do esperado sendo evidenciado uma porcentagem baixa (geralmente abaixo de 10%) para erros positivos (quando a geometria do objeto virtual é maior do que do objeto real e erros negativos, quando a geometria do objeto real é maior do que a do objeto virtual).

Fonte: elaborado pelo autor.

O trabalho de Loura *et al.* (2018) relata uma solução de baixo custo que utiliza um Kinect V1, uma câmera digital e um software desenvolvido por eles para fazer a digitalização de objetos reais os transformando em objetos Computer Aided Design (CAD) que podem ser manipulados em um software de modelagem 3D posteriormente. Os objetos escolhidos pelos autores para serem digitalizados foram obras de artes consideradas culturalmente importantes do Museu de Arqueologia e Etnologia (MAE), na Universidade Federal da Bahia.

Para comparar resultados com ferramentas externas foi utilizado um programa chamado Kinect Fusion que também faz a digitalização de objetos. Dado os resultados encontrados foi perceptível que a solução dos escritores se mostrou precisa o suficiente para fazer uma digitalização de qualidade deixando apenas alguns ruídos na digitalização. Uma comparação entre os objetos virtuais (colunas ímpares com o fundo azul) e objetos reais podem ser observados na Figura 2.

Figura 2 – Objetos virtuais e objetos reais



Fonte: Loura *et al.* (2018).

Por fim, apresenta-se no Quadro 3, uma descrição de funcionalidade de um aplicativo como trabalho correlato do presente estudo.

Quadro 3 – Polycam

Referência	Polycam (2021).
Objetivos	Aplicativo que gera modelos 3D a partir de escaneamento do ambiente utilizando o LiDAR e permite a exportação como um Computer Aided Design (CAD).
Principais funcionalidades	Escaneamento 3D, compartilhamento de objeto e exportação.
Ferramentas de desenvolvimento	Swift.

Fonte: elaborado pelo autor.

Polycam (2021) é um aplicativo de código fechado para os dispositivos móveis da Apple que permite a digitalização de um ambiente e sua manipulação, recortando áreas e isolando objetos. A digitalização é feita a partir de um mapeamento de polígonos utilizando o sensor LiDAR que consegue determinar a profundidade do ambiente. Já a texturização é feita através de uma filmagem usando a câmera do dispositivo, que com as imagens capturadas se torna possível criar a textura do objeto tridimensional virtual. Na figura 3 é possível observar a tela do aplicativo durante uma digitalização onde as partes destacadas em azul representam áreas não mapeadas e os polígonos com linhas brancas representam a área mapeada.

Figura 3 – Digitalização de um ambiente sendo feito utilizando o Polycam

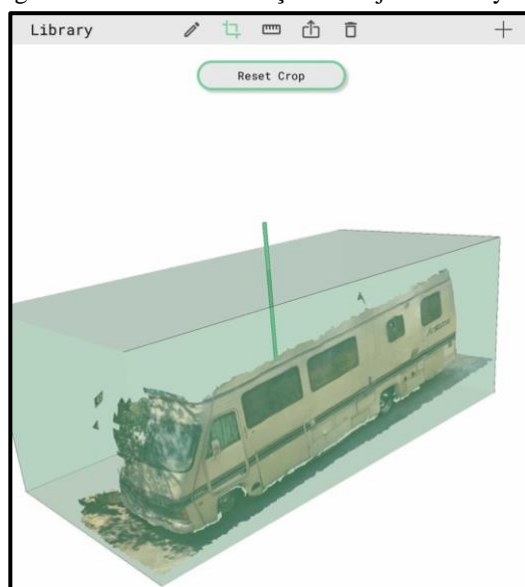


Fonte: Polycam (2021).

Após a finalização da digitalização e o mapeamento do objeto e sua textura, é possível delimitar a área do ambiente capturado para fazer um recorte apenas do objeto de interesse, como pode ser observado na Figura 4.



Figura 4 – Interface de edição de objeto do Polycam



Fonte: Polycam (2021).

Conforme imagem, é possível delimitar uma área que formará o objeto final utilizando a ferramenta de corte que é representada pelo paralelepípedo verde na imagem.

### 3 DESCRIÇÃO DA BIBLIOTECA

Nessa seção serão descritos as funcionalidades e o desenvolvimento da biblioteca do ARKit. Com essa finalidade 4 subseções foram criadas. Na primeira será a especificação da biblioteca, os diagramas de classe e de fluxo. Na segunda subseção será descrita a implementação destacando os principais trechos de código da biblioteca. Na terceira subseção, será demonstrado um passo a passo da instalação da biblioteca em um aplicativo iOS e sua utilização. Por último, na quinta seção, serão mostrados os cenários de testes realizados.

#### 3.1 ESPECIFICAÇÃO

Para desenvolver a biblioteca foi utilizada a linguagem de programação Swift na sua versão 5.7 e o encapsulamento em uma biblioteca através de um Swift Package, que pode ser adicionado em qualquer projeto Swift. Para fazer o mapeamento do ambiente com o LiDAR foi utilizada a Depth API do *framework* ARKit, em conjunto com a SceneGeometry do *framework* SceneKit onde são capturadas as informações dos pontos no mundo e salvos em um objeto de *meshes* que posteriormente, com esse objeto construído, são salvos em um arquivo *.dae* para ser manipulado em um software de modelagem 3D.

A biblioteca é estruturada basicamente em 3 componentes principais. O *ARReceiver* é o núcleo da leitura das informações do ARKit, essa classe pode ser utilizada em qualquer contexto desde que ela receba uma instância de *ARSession*, que servirá como a fonte das informações. O *ARReceiver* também dispõe dos dados e de funções *helpers* para a digitalização 3D. Essa classe consegue ser utilizada independente de uma *View*, podendo ser implementada até como um processo de plano de fundo. Outra classe disponível é a *ARDataProvider* que utiliza o *ARReceiver*, porém abstrai seus métodos para serem utilizados com uma *ARView* que é recebida como parâmetro. Essa *ARView* é a provedora do *ARSession* utilizado pelo *ARReceiver*. O componente final é a *ARDepthView* que utiliza os dois outros componentes e disponibiliza um componente para ser adicionado em um SwiftUI e ser manipulado em tela. Na Figura 5 é possível de ser observado o diagrama das classes descritas.

```

classDiagram
    class ARDataReceiver {
        +protocol=
        +ARDataReceiver
        +onNewARData(arData:)
    }
    class ARProvider {
        +origDepthWidth
        +origDepthHeight
        +arReceiver : ARReceiver
        +lastARData : ARData?
        +arView : ARView
        +debugData : DepthData?
        +init()
        +start()
        +pause()
        +switchCaptureMetrics()
        +createModel()
        +onNewARData(arData:)
    }
    class ObservableObject {
    }
    class DepthData {
        +data
        +matrixSize
        +topRightCorner : Float32?
        +topLeftCorner : Float32?
        +center : Float32?
        +bottomRightCorner : Float32?
        +bottomLeftCorner : Float32?
        +capturingMetrics : Bool
        +init()
        +set(x:y:FloatData:)
        +get(x:y:)
        +getAll()
        +updateOffsets(depthMap:)
        +updateMetrics()
    }
    class ARData {
        +depthMap : CVPixelBuffer?
        +anchors : [ARMeshAnchor]?
        +captureedImage : CVPixelBuffer?
        +cameraIntrinsics
        +cameraResolution
    }
    class ARSessionDelegate {
    }
    class ARReceiver {
        +arData
        +arSession : ARSession
        +delegate : ARDataReceiver?
        +init(arSession:)
        +start()
        +pause()
        +createVertexDescriptor(vertices:)
        +createMesh(vertices:submeshes:vertexBuffer:)
        +parseVertices(geometry:toWorldSpace(vertices:geometry:meshAnchor:)
        +createVertexBuffer(vertices:geometry:meshAnchor:allocator:)
        +createSubmesh(faces:indexBuffer:)
        +createIndexBuffer(faces:allocator:)
        +createModel()
        +session(_didUpdate:)
    }
    class SCNSceneExportDelegate {
    }
    class UIViewRepresentable {
    }
    class ARDepthView {
        +<struct>=
        +arProvider : ARProvider
        +debugData : LidarProviders.DepthData
        +init(arProvider:)
        +makeCoordinator()
        +makeUIView(context:)
        +updateUIView(_context:)
    }

    ARDataReceiver --|> ARProvider
    ARDataReceiver --|> ObservableObject
    ObservableObject --|> DepthData
    ARSessionDelegate --|> ARReceiver
    SCNSceneExportDelegate --|> ARReceiver
    UIViewRepresentable --|> ARDepthView
    ARDepthView --|> ARDepthView
  
```

Também foi criada uma aplicação que utiliza a biblioteca a fim de testar e comprovar as funcionalidades. Essa aplicação consiste em uma `View` utilizando `SwiftUI` que tem um atributo de `ARDepthView`, um botão que trazem métricas de *debug*, e um botão que salva um modelo 3D no dispositivo e permite compartilhar esse modelo utilizando o modal de compartilhamento do `iOS`. Na Figura 6, pode-se observar o diagrama de classes da aplicação.

```

classDiagram
    class App
    class View
    class PreviewProvider
    class SampleApp {
        <<struct>>
        body : some Scene
    }
    class ScanView {
        <<struct>>
        arProvider : ARProvider
        documentsPath
        fileName
        showingSheet : bool
        debugData : LidarProviders.DepthData
        showingDebugMetrics
        topRightCorner : Float32?
        topLeftCorner : Float32?
        center : Float32?
        bottomRightCorner : Float32?
        bottomLeftCorner : Float32?
        body : some View
    }
    class SwiftUIPreviewProvider {
        <<struct>>
        SwiftUIPreviewView : PreviewView
        previews : some View
    }
    App <|-- SampleApp
    View <|-- ScanView
    PreviewProvider <|-- SwiftUIPreviewProvider
    ScanView <|-- SwiftUIPreviewProvider
    
```

The diagram illustrates the class hierarchy and structure of the components. **SampleApp** is a struct that inherits from **App**. **ScanView** is a struct that inherits from **View**. **SwiftUIPreviewProvider** is a struct that inherits from **PreviewProvider** and **ScanView**. The **ScanView** struct contains several properties and methods, including `arProvider`, `documentsPath`, `fileName`, `showingSheet`, `debugData`, `showingDebugMetrics`, `topRightCorner`, `topLeftCorner`, `center`, `bottomRightCorner`, `bottomLeftCorner`, and `body`.

A interface dessa aplicação foi criada de uma forma a contribuir com praticidade ao se realizar os testes necessários e que possibilitasse a exportação de objetos 3D. Na Figura 7, é possível observar que a tela principal possui dois botões na parte inferior, o esquerdo serve para compartilhar o modelo mapeado até o momento, e o direito serve para abrir o modal que se encontra na parte superior da tela que demonstra as distâncias nos *pixels* de cada canto da tela e do centro da tela para motivos de testes.

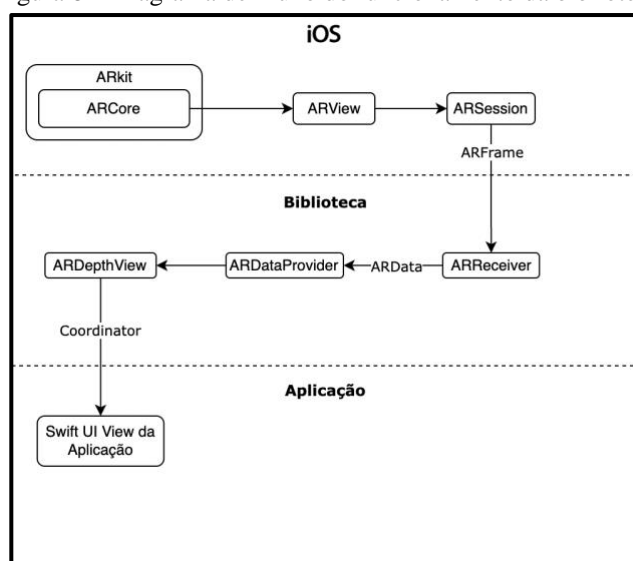
Trabalho de Conclusão de Curso - Ano/Semestre: 2022/1

### 3.2 IMPLEMENTAÇÃO

A cada atualização do `ARSession`, um fluxo é gerado e um objeto de `ARFrame` com as atualizações capturadas pela câmera e pelos sensores do dispositivo é enviado para pôr qualquer objeto que estenda a classe `ARSessionDelegate` e sobrescreva o método `session(_didUpdate)`, para a biblioteca essa classe é o `ARReceiver` que a cada frame de atualização captura as informações necessárias para a digitalização 3D e guarda em um objeto de `ARData` que posteriormente é consumido para gerar o objeto 3D.

Para facilitar a implementação mais comum de uma aplicação de escaneamento a classe `ARDepthView` é disponibilizada e que se for adicionada ao Body de uma SwiftUI, ela cria um objeto de `ARDataProvider` que disponibiliza as APIs necessárias para o escaneamento 3D e demonstra uma `ARView`. Na Figura 8 pode ser observado um diagrama com o passo a passo feito pelo fluxo descrito acima.

Figura 8 – Diagrama de Fluxo do funcionamento da biblioteca



Fonte: elaborado pelo autor.

## 4 RESULTADOS

De modo a ampliar o seu caráter científico, todos os TCCs devem apresentar e discutir resultados não limitados à comparação com os trabalhos correlatos. Devem ser apresentados os casos de testes do software, destacando objetivo do teste, como foi realizada a coleta de dados e a apresentação dos resultados obtidos, preferencialmente em forma de gráficos ou tabelas, fazendo comentários sobre eles. Também é sugerida a comparação com os trabalhos correlatos apresentados na fundamentação teórica.

## 5 CONCLUSÕES

As conclusões devem refletir os principais resultados alcançados, realizando uma avaliação em relação aos objetivos previamente formulados. Deve-se deixar claro se os objetivos foram atendidos, se as ferramentas utilizadas foram adequadas e quais as principais contribuições do trabalho sociais ou práticas para o seu grupo de usuários bem como para o desenvolvimento científico e ou tecnológico da área.

Deve-se incluir também as limitações e as possíveis extensões do TCC.

## REFERÊNCIAS

APPLE. **Apple anuncia novo iPad Pro com inovador scanner LiDAR e trackpad para uso com iPadOS**, 2022a. Disponível em: <https://www.apple.com/br/newsroom/2020/03/apple-unveils-new-ipad-pro-with-lidar-scanner-and-trackpad-support-in-ipados/>. Acesso em: 12 Jun. 2022

APPLE. **Framework ARKit**, 2022b. Disponível em: <https://developer.apple.com/documentation/arkit/>. Acesso em: 12 Jun. 2022



APPLE. **More to Explore with ARKit 6**, 2022c. Disponível em: <https://developer.apple.com/augmented-reality/arkit/>. Acesso em: 12 Jun. 2021.

APPLE. **Displaying a Point Cloud Using Scene depth**, 2021a. Disponível em: [https://developer.apple.com/documentation/arkit/environmental\\_analysis/displaying\\_a\\_point\\_cloud\\_using\\_scene\\_depth](https://developer.apple.com/documentation/arkit/environmental_analysis/displaying_a_point_cloud_using_scene_depth). Acesso em: 12 Jun. 2021.

APPLE. **Visualizing and Interacting with a Reconstructed Scene**, 2021b. Disponível em: [https://developer.apple.com/documentation/arkit/content\\_anchors/visualizing\\_and\\_interacting\\_with\\_a\\_reconstructed\\_scene](https://developer.apple.com/documentation/arkit/content_anchors/visualizing_and_interacting_with_a_reconstructed_scene). Acesso em: 27 nov. 2021.

BAUWENS, Sébastien; BARTHOLOMEUS, Harm; CALDERS, Kim; LEJEUNE, Philippe. Forest Inventory with Terrestrial LiDAR: a comparison of static and hand-held mobile laser scanning. **Forests**, Basel, v. 7, n. 12, p. 127, 21 jun. 2016. MDPI AG. Disponível em: <http://dx.doi.org/10.3390/f7060127>.

GIONGO, Marcos; KOEHLER, Henrique Soares; MACHADO, Sebastião do Amaral; KIRCHNER, Flavio Felipe; MARCHETTI, Marco. LiDAR: princípios e aplicações florestais. **Pesquisa Florestal Brasileira**, [S.L.], v. 30, n. 63, p. 231-244, 28 out. 2010. Embrapa Florestas. <http://dx.doi.org/10.4336/2010.pfb.30.63.231>.

GONZO, L.; VOLTOLINI, F.; GIRARDI, S.; RIZZI, A.; REMONDINO, F.; EL-HAKIM, S.F. Multiple Techniques Approach to the 3D Virtual Reconstruction of Cultural Heritage. *In: EUROGRAPHICS ITALIAN CHAPTER CONFERENCE*, 2, 2007, Trento. **Anais R. De Amicis and G. Conti**. p. 213 – 216. Disponível em: <http://dx.doi.org/10.2312/LocalChapterEvents/ItalChap/ItalianChapConf2007/213-216>.

LOURA, Daniel de Sousa Alves; OLIVEIRA, Yuri de Matos Alves de; RAIMUNDO, Pedro Oliveira; AGÜERO, Karl Philips Apaza. Reconstrução 3D de Objetos com Kinect e Câmera Digital. **Revista Eletrônica de Iniciação Científica em Computação**, Bahia, v. 16, n. 6, p. 1-17, 8 dez. 2018. Sociedade Brasileira de Computacao - SB. Disponível em: <http://dx.doi.org/10.5753/reic.2018.1077>.

MUHADI, N. A. *et al.* The Use of LiDAR-Derived DEM in Flood Applications: a review. **Remote Sensing**, [S.L.], v. 12, n. 14, p. 2308, 18 jul. 2020. MDPI AG. Disponível em: <http://dx.doi.org/10.3390/rs12142308>. Acesso em: 15 maio 2021.

NASSIF, Felipe de Barros; PASSOS, Júlio César; PIMENTA, Felipe Mendonça. **A Tecnologia Lidar Aplicada A Medições Eólicas Sobre Corpos Hídricos E Oceano**. 2017. 111 f. Dissertação (Mestrado) - Curso de Engenharia Mecânica, Universidade Federal de Santa Catarina, Florianópolis, 2017. Disponível em: <https://repositorio.ufsc.br/handle/123456789/186188>. Acesso em: 10 nov. 2021.

POLYCAM. **FAQ**. 2021 Disponível em: <https://poly.cam/learn> Acesso em: 20 set 2021.

RIGUES, Rafael. **Snapchat será um dos primeiros apps a usar o Lidar no iPhone 12 Pro**. 2020. Disponível em: <https://olhardigital.com.br/2020/10/14/noticias/snapchat-sera-um-dos-primeiros-apps-a-usar-o-lidar-no-iphone-12-pro/>. Acesso em: 20 set. 2021.

VOGT, Maximilian; RIPS, Adrian; EMMELMANN, Claus. Comparison of iPad Pro®'s LiDAR and TrueDepth Capabilities with an Industrial 3D Scanning Solution. **Technologies**, Basel, v. 9, n. 2, p. 25, 7 abr. 2021. MDPI AG. Disponível em: <http://dx.doi.org/10.3390/technologies9020025>.