

INTRODUÇÃO À REALIDADE AUMENTADA E UMA APLICAÇÃO NA MONTAGEM ELETROMECHANICA

Dickson Juan Frutuoso da Silva – dickson_juan@outlook.com

Gustavo Hochgraf – gusthoch@gmail.com

Ms. Hélio Pekelman (Orientador) – helio.pekelman@mackenzie.br

RESUMO

Os serviços de montagem de equipamentos de maior complexidade são orientados através de manuais textuais, podendo até mesmo exigir um treinamento e habilidade prévios de compreensão e leitura de termos técnicos por parte do usuário responsável por realizar essas tarefas. Observa-se uma oportunidade de ganho de eficiência ao fornecer um método que garanta maior compreensão do usuário de suas funções. Com isso, estudos sugerem um formato mais visual, como a Realidade Aumentada (RA), para criação de um manual de instruções para montagem de equipamentos eletromecânicos que permita a visualização das etapas de montagem. Como contribuição é proposto, por meio de uma revisão bibliográfica, um entendimento do funcionamento desse tipo de sistema, além da construção de um aplicativo em RA demonstrando sua aplicação. Portanto, este trabalho apresenta o desenvolvimento de dois aplicativos de RA em diferentes plataformas, explicitando as limitações das tecnologias e ferramentas inerentes ao desenvolvimento dessas soluções tecnológicas. Palavras-chave: Realidade aumentada. Montagem. Manutenção. Aplicativo.

INTRODUCTION TO AUGMENTED REALITY AND AN APPLICATION IN ELECTROMECHANICAL ASSEMBLY

ABSTRACT

The assembly assignments of more complex equipment are carried out through textual manuals and may even require prior training by the user who will perform these tasks. There is an opportunity to gain efficiency by providing a method that ensures greater understanding by the users of their jobs. Studies suggest a more visual format, such as Augmented Reality (AR), to create an instruction manual for the assembly of electromechanical equipment that allows the visualization of the assembly steps. As a contribution, through a literature review, an introduction that allows an understanding of this type of system is proposed, as well as the construction of an AR program demonstrating its purpose. Therefore, this work presents the development of two AR applications on different platforms, explaining the limitations of the technologies and tools inherent to the development of these technological solutions.

Keywords: Augmented reality. Assembly. Maintenance. App.

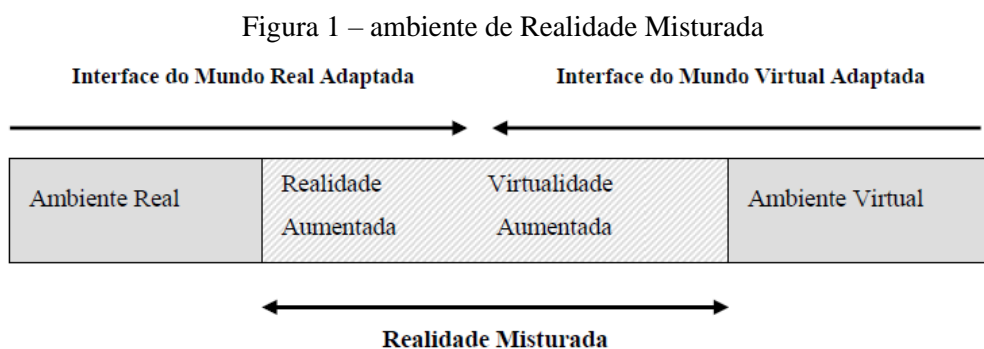
1 INTRODUÇÃO

A montagem e a manutenção de objetos são tarefas que não estão somente presentes na indústria, mas também no cotidiano em geral. Estas tarefas tanto podem ser simples, como abrir a tampa de um controle remoto para trocar suas pilhas quanto complexas se considerarmos a tarefa de trocar um capacitor queimado. Assim, para prevenir danos à montagem e/ou manutenção de dispositivos complexos, a leitura e correta compreensão de manuais e guias escritos são necessários para evitar danos aos componentes desses dispositivos.

Com o aumento da complexidade dos equipamentos e/ou dispositivos, surgem mais dificuldades de seguir suas instruções, requerendo maior tempo de dedicação para montagem e aumentando a possibilidade de ocorrerem erros durante esse processo. Kloberdanz (2017) se refere a um estudo feito pela *General Electric* (GE), no qual se mostrou uma perda de milhões de dólares na indústria de turbinas a jato devido ao aperto incorreto de parafusos na montagem do equipamento.

A RA é definida como uma variação da Realidade Virtual (RV). Na RV visualiza-se um ambiente e objetos totalmente virtuais, enquanto na RA há uma mescla do ambiente real com objetos e informações virtuais (AZUMA, 1997), portanto a RA atua como um complemento do ambiente real, ao invés de substituí-lo completamente como na RV.

A Figura 1 ilustra a relação entre o mundo totalmente real e o totalmente virtual (RV). A intersecção entre esses mundos, através da imersão de objetos ou ambientes, é classificada como Realidade Misturada ou Mista. Dentro da Realidade Mista o que difere a RA da Virtualidade Aumentada é a predominância do ambiente real ou virtual (TORI; KIRNER; SISCOOTTO, 2006).



Fonte: Tori, Kirner e Siscoutto (2006)

Já na Figura 2 nota-se usuários capazes de observar e manipular componentes internos de um carro protótipo em escala reduzida com a RA, através de uma plataforma colaborativa. Este é um exemplo de uso dessa imersão de novos objetos no mundo real através de dispositivos móveis e dedicados, como *tablets* e óculos de RA.

Figura 2 – uso da RA para visualização de componentes internos

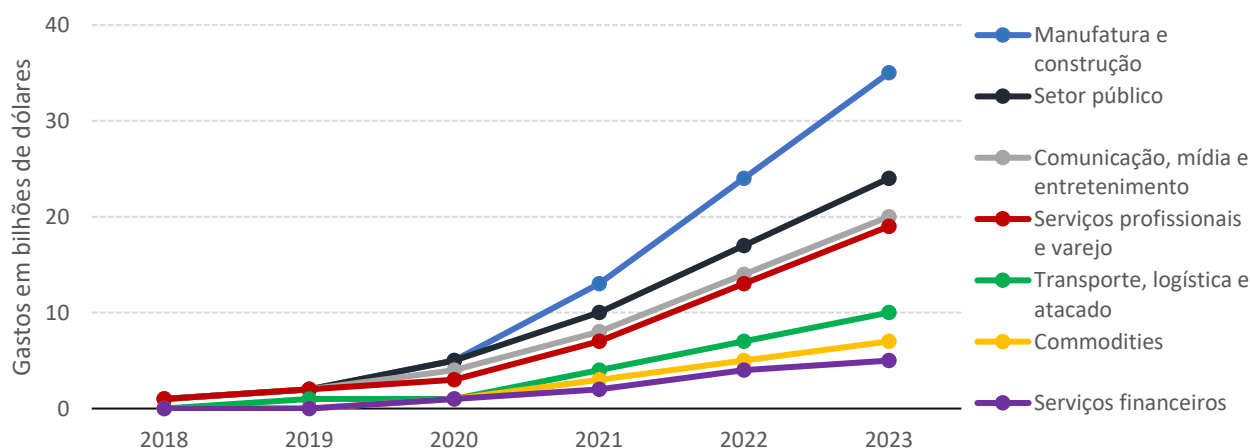


Fonte: BOSCH AUTO TOOLS (2019)

Por meio de estudos práticos, Watson *et al.* (2008) identificaram que a montagem de objetos guiada por meios textuais apresenta maior dispêndio de tempo ao aprendiz, assim como maior tempo para execução da tarefa em relação aos meios visuais. Ademais, Schwald e Laval (2003) também destacam o grande potencial desses meios visuais, como a RA, em serviços de manutenção em ambientes industriais, frente às documentações escritas e materiais digitais usualmente utilizados.

É projetado um crescimento acelerado no mercado de RA, em especial no setor de manufatura e construção, como observado no Gráfico 1. Saracco (2020) justifica essa evolução como uma resposta às restrições de circulação de pessoas durante a pandemia do COVID-19, no que se refere à necessidade de entregar objetos e experiências de forma virtual ao usuário devido à restrição na circulação de pessoas. Além disso, também é destacado o potencial dessa tecnologia na redução de custos e na execução de treinamentos em geral que têm se atualizado constantemente devido à crescente complexidade dos produtos, maiores requisitos de qualidade, aperfeiçoamento das ferramentas e maior interação entre diferentes equipes.

Gráfico 1 – projeção do mercado de RA por setor



Fonte: adaptado de Saracco (2020)

Este trabalho tem como objetivo conceituar, entender e desenvolver uma aplicação de montagem de equipamentos utilizando-se da Realidade Aumentada (RA) por meio de dispositivos móveis e empregando como estudo de caso a composição de uma garra robótica.

Para este objetivo foram estudados conceitos de RA, das tecnologias de hardware que viabilizam a solução e das tecnologias para desenvolvimento de sistemas em RA, além disso, também foram definidos requisitos do aplicativo e o equipamento a ser utilizado.

Por fim, foi feito o desenvolvimento e teste de um produto viável mínimo no formato de um aplicativo. Este, composto de uma interface intuitiva, mostra com RA as etapas de montagem de uma garra robótica. Essas etapas foram representadas em um objeto tridimensional animado que é posicionado por meio da câmera de um dispositivo móvel em um ambiente real.

2 REVISÃO DA LITERATURA

2.1 MAPEAMENTO E NAVEGAÇÃO

Uma importante característica e novidade tecnológica da RV e RA é a experiência da navegação tridimensional pelo usuário de forma natural no ambiente, ou seja, é possível se movimentar e visualizar de diferentes perspectivas a mesma cena (TORI; KIRNER; SISCOOTTO, 2006). Para que não haja prejuízo na experiência, o dispositivo utilizado necessita interpretar os movimentos nos seis diferentes graus de liberdade, três referentes à translação e três referentes à rotação, além de processar as alterações necessárias à exibição.

Para a interpretação do sensoriamento do ambiente, um importante método para o mapeamento e estimativa de posição do dispositivo é a *Simultaneous Localization and Mapping* (SLAM). Como abordado por Mataric (2014) e Taheri e Xia (2021), nesse método são feitos o mapeamento tridimensional do ambiente sem conhecimento prévio e sem interferência do usuário e simultaneamente à localização do dispositivo no mapa gerado. Para tal, durante esse mapeamento é feita a análise e composição das informações obtidas através das câmeras, sensores ultrassônicos, infravermelhos, acelerômetros e giroscópios, gerando uma nuvem de pontos observada na Figura 3.

Figura 3 – nuvem de pontos SLAM



Fonte: adaptado de Roberto *et al.* (2019)

Como exposto por Wu et al. (2020) e Taheri e Xia (2021) o sensor de *light detection and ranging* (LiDAR) vem sendo amplamente utilizado na criação da nuvem de pontos. Esse sensor emite um feixe de raios infravermelhos, os quais são refletidos ao encontrar um corpo no ambiente e retornam ao mesmo. Com o tempo necessário ao retorno, o sensor é capaz de calcular a distância desse ponto. Ao repetir esse processo com milhões de feixes por segundo, é possível criar em tempo real uma nuvem precisa de pontos para a navegação.

2.2 AMBIENTE VIRTUAL: ÂNCORAS E PONTOS DE CARACTERÍSTICA

Para garantir o realismo das aplicações de RA é necessário que a posição e a orientação dos objetos inseridos no ambiente sejam mantidas, mesmo com a movimentação do dispositivo. Para que essa diretriz seja cumprida é necessário o uso das âncoras virtuais (GOOGLE ARCORE, 2020).

De acordo com Choi, Lim e Ko (2020), âncoras virtuais são pontos de referência do mundo real utilizados para orientar o mundo virtual. Quanto aos pontos de características, também conhecidos como *feature points*, são obtidos a partir de objetos reais em cena e dão origem ao mapa de pontos, permitindo assim, que as âncoras sejam inseridas no mapa sem a necessidade de definir objetos específicos para serem reconhecidos, tendo sua posição em relação ao dispositivo constantemente alterada conforme à atualização do SLAM.

2.3 MARCADORES

As âncoras, também chamadas de marcadores, podem ser de diferentes tipos, de acordo com Herpich, Guarese e Tarouco (2017):

- 1- Reconhecimento de texto: o dispositivo reconhece um texto pré-determinado pela aplicação ou pelo usuário;
- 2- Reconhecimento de imagem: o dispositivo reconhece imagens que podem ser tanto desenhos e fotos, como também códigos únicos como códigos *Quick Response* (QR) gerados somente para este objeto. Um exemplo do reconhecimento de imagens está presente na Figura 4, na qual se nota o emprego da tecnologia para a tradução de objetos bidimensionais em objetos tridimensionais.

Figura 4 – reconhecimento de planta baixa



Fonte: Gevorgyan (2017)

- 3- Reconhecimento de objetos tridimensionais: podem ser utilizados tanto modelos computacionais, criados com *softwares* CAD para se assemelhar aos objetos reais, como também a partir de modelos tridimensionais criados por escaneamentos destes objetos reais;
- 4- Reconhecimento sem marcadores: é possível reconhecer e determinar posições com precisão e sem a necessidade de marcadores por meio do mapeamento do ambiente e seus pontos de interesse.

Além do posicionamento dos objetos virtuais através dos marcadores, é importante que essa posição e orientação sejam preservadas mesmo quando o marcador não está mais presente no campo de visão do dispositivo (UNITY TECHNOLOGIES, 2020). Essa funcionalidade é conhecida como rastreamento estendido e está relacionada com o mapeamento SLAM e seus *feature points* e *anchors points*.

2.4 OBJETOS VIRTUAIS

O sistema visual humano binocular possibilita a visualização dos ambientes tridimensionais e suas características, o que é feito a partir da interpretação do cérebro das duas imagens recebidas dos olhos. Esse efeito é conhecido como estereoscopia e suas características de perspectiva, iluminação, oclusão e sombra são alvo de estudo dos sistemas de RA e RV, sendo fundamentais para o realismo dos objetos virtuais (TORI; KIRNER; SISCOOTTO, 2006).

O que define a relação de tamanho e posicionamento entre os objetos virtuais e a cena real é o efeito de perspectiva. Ele dá a noção da distância do objeto e a compreensão do seu tamanho real ao fazer comparações com a cena. A perspectiva permite, através do redimensionamento e posicionamento dos corpos, trazer tridimensionalidade e realismo às imagens planas.

Outro efeito importante é o de iluminação, responsável por trazer aos corpos mais detalhes sobre suas superfícies e geometria. Já a oclusão é a sobreposição dos objetos a partir do ponto de vista do usuário, por meio dela a posição relativa entre dois objetos é entregue, permitindo que seja compreendido qual está à frente do outro e contribuindo para a noção de profundidade. Na Figura 5 há um exemplo de oclusão em RA com um personagem virtual, permitindo reconhecer como esse efeito garante mais realismo à aplicação.

Figura 5 – efeito de oclusão em ambientes reais



Fonte: Ahir (2020)

Outro efeito responsável por entregar noções de posição é a sombra, com ela é possível reconhecer se o objeto está em contato direto com a superfície e qual sua posição relativa com a fonte de luz da cena.

A combinação sinérgica desses efeitos entrega características tridimensionais da cena ao usuário, inserindo o objeto virtual de forma mais realista e trazendo mais detalhes sobre o mesmo (TORI; KIRNER; SISCOOTTO, 2006).

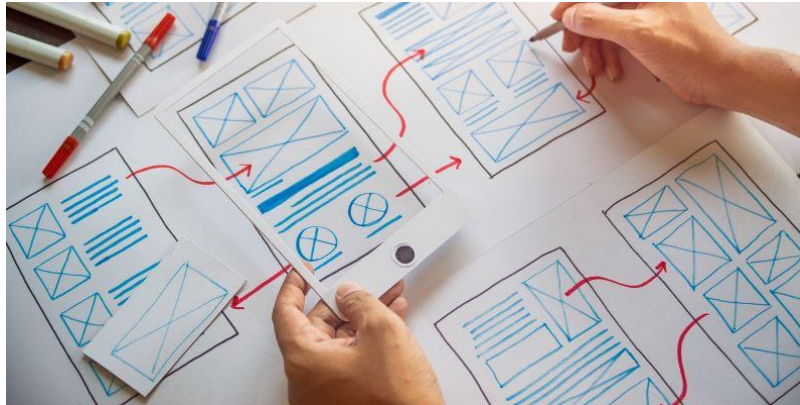
2.5 EXPERIÊNCIA DE USUÁRIO

A experiência de usuário, do inglês *User Experiencie (UX)*, é a área que estuda não somente a usabilidade dos produtos como também a ligação emocional e percepção deles (HAMMADY, 2018), garantindo assim que o usuário possa usar produtos tangíveis e não-tangíveis de forma intuitiva e ainda ter uma experiência agradável.

Conforme Hammady (2018), a UX pode ser dividida em três fases, a fase de requisitos, a fase de *design* e a fase de testes. O que mais diferencia a RA de aplicações comuns, como sites e aplicativos móveis, é a fase do *design*, mais especificamente o *design* da interface de usuário, ou *User Interface (UI)* que, de forma geral, são os recursos que permitem que o usuário interaja com o sistema. Essa diferença da RA quanto aos outros sistemas se baseia na dificuldade da construção da UI por conta da grande quantidade de dispositivos e maneiras de interação possíveis em uma aplicação.

Na Figura 6 pode-se observar a elaboração do fluxograma de um protótipo de aplicativo, também chamado de *wireframe*. Segundo Hardianto e Karmilasari (2019), o *wireframe* é utilizado para explicitar o posicionamento dos componentes da UI. Além disso, esse fluxograma também tem por finalidade ilustrar o funcionamento do fluxo entre as telas do aplicativo integrando parte do procedimento de estudo da UX.

Figura 6 – elaboração de UX e UI



Fonte: Yopper (2020)

2.6 DESENVOLVIMENTO DE SISTEMAS EM RA (SDK, FRAMEWORKS E LINGUAGENS)

Devido à alta complexidade para o processo de desenvolvimento de aplicativos de realidade aumentada, ao longo dos anos foram construídas tecnologias com objetivo facilitar este processo, denominadas *frameworks* ou *Software Development Kits* (SDKs) (HANAFI; ELAACHAK; BOUHORMA, 2019). Estes SDKs englobam muitas funcionalidades necessárias à RA, como o reconhecimento de marcadores, o rastreamento do ambiente e a renderização dos conteúdos 3D (HERPICH; GUARESE; TAROUCO, 2017). Essas funcionalidades são fornecidas de forma predefinida ao usuário, dispensando a necessidade de programação. Para Hanafi, Elaachak e Bouhorma (2019) os SDKs mais recomendados seriam o GoogleARCore, ARKit (plataforma desenvolvida pela Apple) e o Vuforia (plataforma desenvolvida pela PTC) pela possibilidade de serem utilizados com a plataforma Unity.

O Unity é uma plataforma que permite o desenvolvimento de aplicações com objetos 3D, realizando a renderização e animação dos objetos de forma otimizada e simulando a física do projeto (JULIANI *et al.*, 2020). Além disso, Juliani *et al.* (2020) também descreve o Unity, mais especificamente o Unity Editor, como uma interface gráfica que fornece ferramentas para o desenvolvimento dessas aplicações. Nessa plataforma, é possível adicionar os SDKs como pacotes complementares, permitindo ao usuário utilizar as funcionalidades desejadas para a construção da sua solução (HERPICH; GUARESE; TAROUCO, 2017).

Ao longo dos anos, com o aumento da qualidade de aplicações *web* somada ao aumento da largura de barramento (velocidade em bits/s) da *internet*, como no caso das redes 5G, foi surgindo uma nova modalidade de RA voltada para aplicativos *web* (QIAO *et al.*, 2019). Também de acordo com Qiao *et al.* (2019) essa nova modalidade permite a construção de aplicações complexas que, por serem hospedadas em servidores, não necessitam que o usuário as instale em seu dispositivo tornando-as mais leves.

Como descrito por MacIntyre e Smith (2018), dado o grande número de dispositivos que executam RA, existe a dificuldade em criar alguma tecnologia que permita desenvolver uma única aplicação compatível com todos. Assim surge a WebXR, tecnologia que permite criar aplicações *web* de RA para diversos dispositivos.

No Quadro 1 pode-se observar os tipos de licença de cada um dos SDKs. O ARKit não contém uma licença para desenvolvimento gratuito. Já no caso da licença de código aberto (*open source*), somente o ARCore libera esta permissão, que pode ser muito benéfico para o desenvolvimento de melhorias na tecnologia porque permite que diversos colaboradores contribuam com a sua evolução, não se tornando dependente do desenvolvimento por parte da empresa de origem.

Quadro 1 – comparativo entre SDKs: tipos de licença

AR SDK		ARCore	ARKit	Vuforia
Tipo				
Licença	Código Aberto	Sim	Não	Não
	Gratuito	Sim	Não	Sim
	Comercial	Sim	Sim	Sim

Fonte: adaptado de Hanafi, Elaachak e Bouhorma (2019)

O Quadro 2 contém um estudo comparativo entre plataformas que os SDKs podem ser executados. Pode-se perceber que o desenvolvimento para o sistema operacional IOS, utilizado em produtos da Apple, se limita ao uso do ARKit. Entretanto, para produtos com sistema Android e Windows, pode-se usar tanto o ARCore quanto o Vuforia.

Quadro 2 – Compatibilidade entre sistemas operacionais

AR SDK		ARCore	ARKit	Vuforia
Tipo				
Plataforma	IOS	Não	Sim	Não
	Android	Sim	Não	Sim
	Windows	Sim	Não	Sim

Fonte: adaptado de Hanafi, Elaachak e Bouhorma (2019)

No Quadro 3 estão especificadas as diferentes linguagens de programação e suas compatibilidades com os SDKs. O Unity e Unreal são os mais recomendados por necessitarem codificações mínimas, além de serem utilizados para projetos tridimensionais como jogos e ferramentas de engenharia.

Quadro 3 – comparativo entre SDKs: linguagens de programação aceitas

AR SDK			
Linguagens de Programação	ARCore	ARKit	Vuforia
Java	Sim	Não	Sim
Objectif C	Não	Sim	Sim
Unity	Sim	Em fase experimental como um <i>plugin</i> na Unity Asset Store	Sim
Unreal	Sim	Não	Não
C/C++	Não	Não	Sim
JavaScript	Não	Não	Não
Outros	N/A	Xcode	N/A

Fonte: adaptado de Hanafi, Elaachak e Bouhorma (2019)

Em relação às técnicas de rastreamento, Hanafi, Elaachak e Bouhorma (2019) analisam no Quadro 4 a qualidade destas funcionalidades de forma comparativa entre os SDKs. Destaca-se que o Vuforia é o mais eficiente em relação a reconhecimento de imagens. Em contrapartida, esse SDK se mostra menos eficiente em relação ao reconhecimento de movimentos. Já no caso da detecção de planos o Vuforia utiliza um método diferente e menos estável do que o do ARCore e do ARKit, baseado na orientação do dispositivo e não nos *feature points*. Somando-se a isso, os outros dois SDKs se mostram estáveis nessa detecção, porém ocorrem problemas quando há mudanças na orientação da tela do dispositivo.

Quadro 4 – comparativo entre SDKs: recursos de rastreamento

Técnicas	Reconhecimento de Imagem	Rastreamento de Movimento	Detecção de Planos
SDKs			
ARCore	Não estável, muitos escorregamentos inesperados	Estável na maior parte do tempo	Estável, mas escorregamentos ocorrem devido à problemas com a ancoragem enquanto durante a mudança de orientação da tela
ARKit			
Vuforia	Muito estável	Não estável, ocorrem disparidades na orientação	Aparenta que sua detecção é baseada na orientação e não com <i>feature points</i>

Fonte: adaptado de Hanafi, Elaachak e Bouhorma (2019)

A aplicação de RA pode ter sua capacidade medida através de suas funcionalidades, no Quadro 5 são demonstradas as diferenças entre elas. O ARCore é o único SDK que somente contém o método SLAM para sensoriamento do ambiente e seus objetos, sendo assim, pode ser considerado um SDK mais simples por não possuir métodos diversos para reconhecimento.

Quadro 5 – comparativo entre SDKs: funcionalidades

SDK	Reconhecimento			SLAM	Múltiplos marcadores	Rastreamento estendido
	2D	3D	Nuvem			
ARCore	Não	Não	Não	Sim	Não aplicável	Não
ARKit	Sim	Sim	Não	Sim	Sim	Não
Vuforia	Sim	Sim	Sim	Sim	Sim	Sim

Fonte: adaptado de Hanafi, Elaachak e Bouhorma (2019)

Além dessas SDKs comparadas nos quadros 1 a 5, a WebXR, de acordo com MacIntyre e Smith (2018), é uma tecnologia muito atraente por ser *open source*. Ademais, foi criada a partir de uma comunidade de estudantes da RA e da RV que buscavam uma tecnologia que garantisse maior compatibilidade entre dispositivos, além de garantir maior leque de recursos disponíveis para desenvolvimento, sendo tudo oferecido de forma gratuita.

A WebXR apresenta licença gratuita e de código aberto, tendo como sua linguagem de programação o Javascript, uma das linguagens mais modernas de programação da atualidade. Apesar da alta compatibilidade, de acordo com MDN CONTRIBUTORS (2021) a tecnologia somente pode ser utilizada por intermédio de *browsers* de *internet*, sendo eles o Microsoft Edge e Google Chrome para dispositivos *desktop* ou o Google Chrome Android e Samsung Internet para dispositivos móveis. A maior vantagem é que dispositivos móveis, que permitam o uso desses *browsers* e que têm suporte para o uso da RA e RV, poderão utilizar o WebXR sem se preocupar com problemas de compatibilidade pelo sistema operacional.

2.7 DISPOSITIVOS EM RA: MÓVEIS E *WEARABLES*

O avanço no desenvolvimento e diminuição no custo das tecnologias e sensores para o reconhecimento de objetos e ambientes permitiu que a RA pudesse começar a ser utilizada em dispositivos que já haviam sido comprados para usos pessoais como: computadores de mesa, *laptops* com câmeras, *tablets* e até *smartphones* (AZUMA, 2016), dentre eles classificam-se como dispositivos móveis os *tablets* e os *smartphones*.

A maior quantidade de pontos reconhecidos no ambiente permite uma experiência de RA mais real, porém exige um maior processamento do dispositivo. Portanto se o dispositivo não atender esta demanda de processamento podem ocorrer quedas de performance da aplicação e, como consequência, uma experiência ruim para o usuário. Então nota-se que quanto melhor for a capacidade de processamento do dispositivo, melhor será a experiência do usuário (CHOI; LIM; KO, 2020).

No Quadro 6 há uma comparação da compatibilidade dos aplicativos nos principais sistemas operacionais, com base no SDK utilizado para seu desenvolvimento. É importante ressaltar que o ARKit é um SDK desenvolvido pela Apple e restrito para uso apenas com o sistema da própria empresa, o IOS. Os dispositivos baseados no sistema Windows têm menor compatibilidade com os SDKs, isso porque o foco do desenvolvimento de RA está mais voltado para dispositivos móveis e óculos inteligentes. Por outro lado, os dispositivos baseados em IOS apresentam maior compatibilidade com os SDKs, pois seu fabricante tem interesse no desenvolvimento de tecnologias na RA (APPLE, 2021a).

Quadro 6 - Compatibilidade dos SDKs com os principais sistemas operacionais

Sistema operacional	Android	IOS	Windows
SDKs			
ARCore	≥ 7.0	≥ 11	Não compatível
ARKit	Não compatível	≥ 12	Não compatível
Vuforia	≥ 4.1	≥ 9	10
WebXR ¹	Sim	Sim	Sim

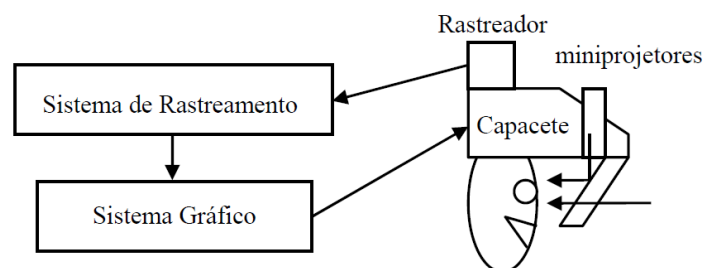
Fonte: Elaborado a partir de Apple ([2021?]), Google ARCore (2020), MDN Contributors (2021) e UNITY TECHNOLOGIES (2019)

Atualmente os dispositivos móveis que apresentam maior tecnologia voltada para RA são os que contêm sensores dedicados para medição de profundidade, como os seguintes dispositivos da Apple: iPhone 12 Pro, iPhone 12 Pro Max e iPad Pro. Esse fato é decorrente da decisão de empresas de incluir sensores LiDAR, como decidiu a própria Apple em sua nova linha de aparelhos em 2020. Além da inclusão do sensor nos aparelhos, a fabricante também atualizou seu SDK para permitir o desenvolvimento de experiências de RA com maior precisão com o uso do mesmo (APPLE, 2021a).

Além da Apple, outras empresas estão investindo em tecnologias para aprimorar as experiências de RA em seus dispositivos. Para tal, pode-se fazer o uso de múltiplas câmeras que, com a comparação de suas imagens, conseguem captar com maior precisão a profundidade do ambiente (WADHWA; ZHANG, 2019).

Como definido por Azuma (1997), a utilização da RA pode ser feita através de um *Head-Mounted Display* (HMD), dispositivos semelhantes à um capacete ou óculos e que possuem todo o conjunto de *hardware* e *software* de sensoriamento e visualização dos objetos aumentados. Na Figura 7 se observa um diagrama esquemático de um HMD, no qual as lentes são semitransparentes e permitem a visualização simultânea do ambiente real e das imagens projetadas pelos projetores dos óculos.

Figura 7 – esquema de funcionamento HMD



Fonte: Tori, Kirner e Siscoutto (2006)

¹ A compatibilidade do WebXR não depende do sistema operacional, mas do navegador utilizado

Os HMDs apresentam maior penetração no mercado industrial, por permitirem a visualização da informação de forma simultânea ao trabalho realizado, não ocupando as mãos e permitindo que os trabalhadores os carreguem livremente pela fábrica (FRAGA-LAMAS *et al.*, 2018). A pesquisa de Cardoso, Mariano e Zorzal (2020) sobre a RA no ambiente industrial expõe que os HMDS são o principal tipo de dispositivo escolhido nas soluções, estando presentes em 40% das mais de 100 produções científicas analisadas.

Para Diaz (2020), o HoloLens e o Magic Leap One podem ser considerados como dispositivos completos para RA e sendo focados para o uso profissional, por exemplo para engenheiros e mecânicos. Por outro lado, o Google Glass Enterprise 2 possui um foco para sobrepor informações e imagens ao usuário, mas sem possibilitar interações complexas ou manipulações de objetos virtuais.

3 METODOLOGIA

Por intermédio de um estudo de caso, este trabalho tem como objetivo conceituar, entender e desenvolver uma aplicação de RA para dispositivos móveis na montagem utilizando como estudo de caso uma garra robótica.

Para este objetivo foram estudados conceitos de RA, as tecnologias de hardware que viabilizam a solução e as tecnologias para desenvolvimento de sistemas em RA, além disso, também foram definidos requisitos do aplicativo e qual equipamento seria utilizado.

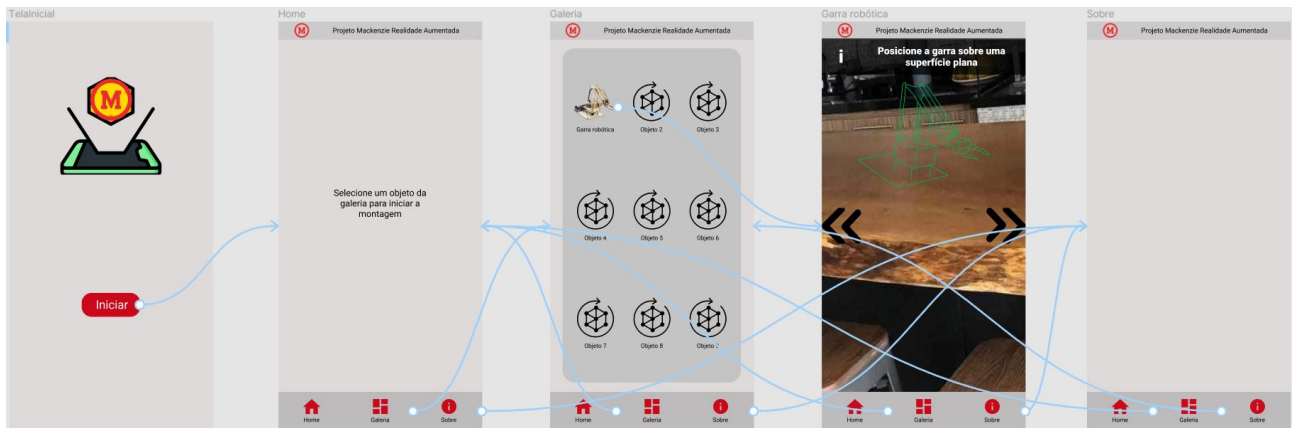
Por fim, foram feitos o desenvolvimento e o teste de um produto viável mínimo no formato de um aplicativo, composto de uma interface intuitiva e demonstrando com RA as etapas de montagem de uma garra robótica. Essas etapas foram representadas em um objeto tridimensional animado que é posicionado por meio da câmera de um dispositivo móvel em um ambiente real.

4 RESULTADOS E DISCUSSÃO

A descrição do processo de desenvolvimento dos aplicativos foi realizada de forma cronológica com o intuito de facilitar seu entendimento, além de auxiliar possíveis interessados a replicar os passos e produzirem seu próprio aplicativo.

Este processo se iniciou no desenvolvimento do protótipo com a utilização da ferramenta Figma, uma ferramenta para prototipagem de aplicativos web e mobile. No Figma é possível diagramar os fluxos que o aplicativo deverá seguir, assim como sua UI, desta forma, permitindo a criação de uma UX. Na Figura 8 é possível ver como o Figma foi utilizado para criar o protótipo da aplicação.

Figura 8 – ferramenta Figma para UI



Fonte: produzido pelos próprios autores

A partir da conclusão do protótipo foi iniciada a pesquisa dos sistemas a serem utilizados para o desenvolvimento, pois já era possível entender o que seria necessário para o aplicativo.

O *framework* definido foi o AR Foundation, fornecido pela empresa Unity, por conter numerosos tutoriais disponíveis para consulta, assim como grande engajamento da comunidade de desenvolvedores em seu fórum, viabilizando a rápida resolução de possíveis problemas encontrados no desenvolvimento. Além disso, o AR Foundation é gratuito para as necessidades deste projeto, além de ser multiplataforma, ou seja, permite desenvolver tanto para Android, quanto para aparelhos iOS.

Para o estudo, foram escolhidos os aparelhos do tipo Android por ser a plataforma disponível para teste. Esta escolha limitou a utilização de somente funções do pacote Google ARCore, inviabilizando o uso de marcadores do tipo de reconhecimento de objeto, pois estes somente funcionam com o pacote ARKit, também fornecido pelo AR Foundation.

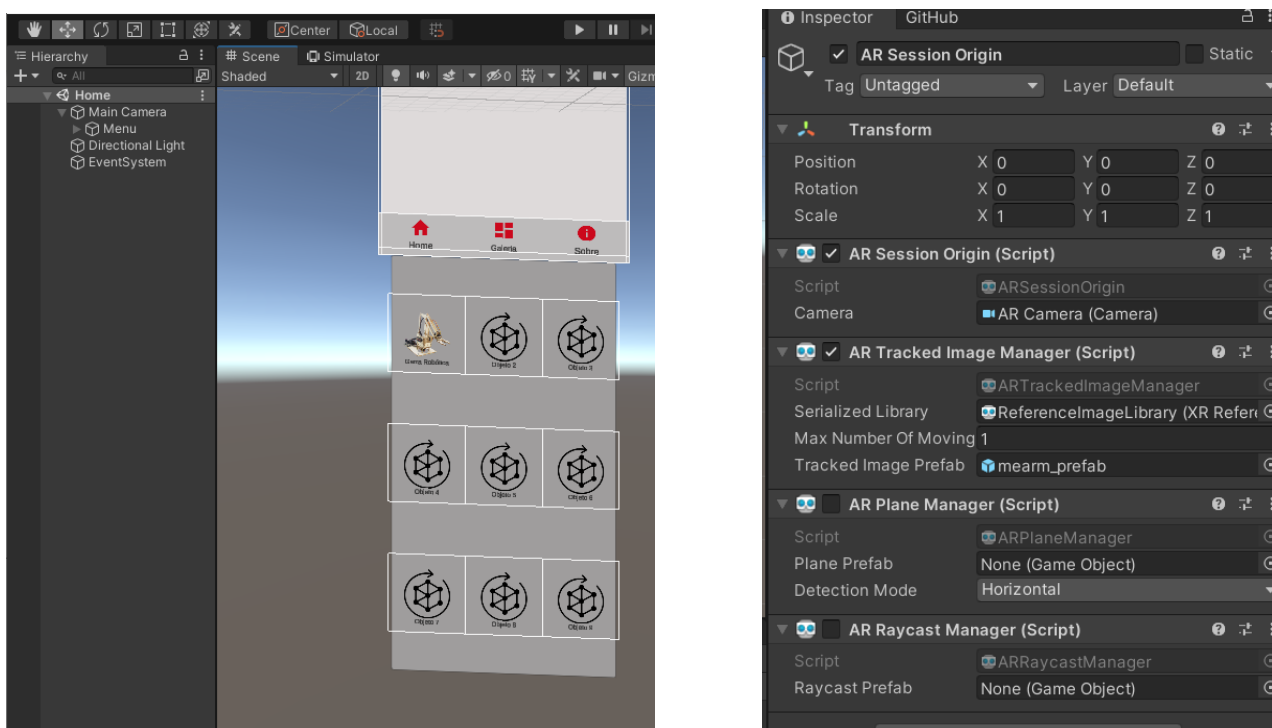
A ferramenta escolhida para a construção do aplicativo foi o editor da Unity por garantir ótima compatibilidade com o AR Foundation, por serem da mesma empresa e possuírem boa uma experiência de usuário, facilitando tanto a codificação quanto a parte de animações tridimensionais necessárias.

Uma vez definidas as ferramentas, foi criado um projeto no Unity com as configurações necessárias para dispositivos móveis com sistema operacional Android. Nesse projeto, foi definida uma cena inicial que se trata de um espaço tridimensional de trabalho, similar ao de outras ferramentas de modelagem 3D e de *Computer Aided Design* (CAD).

Essa primeira cena foi utilizada para criação da interface, vista na Figura 9 à esquerda, utilizando as bibliotecas padrões do Unity. Essas bibliotecas suportam a inclusão de elementos gráficos, como figuras geométricas e imagens, sem qualquer codificação. Também foram utilizadas poucas linhas de programação na linguagem C# para garantir a interatividade de algumas funções dessa interface. Esses recursos permitem a construção em conformidade com o protótipo especificado durante o desenvolvimento da UI. Além disso, o editor possui um emulador de tela de celular que permite uma rápida visualização e teste durante a produção do aplicativo.

Assim prosseguiu-se a criação de uma segunda cena dedicada ao ambiente de RA, inserindo-se as funções de RA fornecidas pelo pacote AR Foundation observadas na Figura 9 à direita, que viabilizaram a detecção de ambientes e imagens.

Figura 9 – cena de interface e configuração do pacote AR Foundation no Unity



Fonte: produzido pelos próprios autores

Inicialmente, o posicionamento do objeto virtual foi feito utilizando a detecção de planos horizontais na cena. Após o reconhecimento, o objeto é inserido no local desejado através de um clique na tela.

Para tal função foi feito uso de programação em C# para que, ao clicar na tela, fosse criada uma linha virtual invisível perpendicular ao celular. Essa linha cria uma variável com as coordenadas de seu cruzamento com o plano mapeado a partir dos pontos de característica do ambiente detectados pelo pacote do AR Foundation. As coordenadas adquiridas são utilizadas para criação de uma âncora virtual que serve para posição e orientação do objeto virtual no ambiente mapeado.

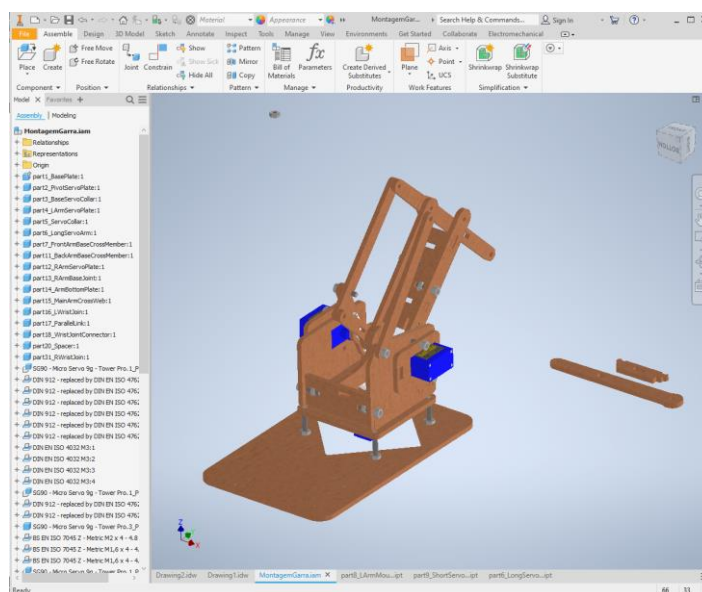
Com o uso dessa técnica de detecção, notou-se que a oclusão do plano horizontal com objetos ou mãos influenciava no posicionamento do objeto virtual, resultando em uma perda de sua ancoragem, ou seja, um prejuízo de sua orientação.

Visando obter uma ancoragem mais consistente, deu-se início a utilização de uma imagem como marcador. Neste método não houve necessidade de codificação, uma vez que tal função é fornecida pelo AR Foundation, sendo somente necessário indicar a imagem alvo e o objeto a ser inserido em cena, tais características são configuráveis dentro da própria área de trabalho do Unity.

Em princípio se experimentou utilizar a própria base da garra como marcador. Porém esse tipo de reconhecimento requer imagens com formas geométricas e aparência mais facilmente identificáveis, sendo muito comum o uso de códigos QR. Nesse sentido, gerou-se um código QR para servir de marcador ao objeto, o que se mostrou altamente preciso, mesmo com a sua movimentação e até oclusão parcial o objeto manteve sua posição e orientação devidamente ancoradas.

Para criação do objeto virtual da RA, foi realizada a modelagem da garra robótica no Autodesk Inventor, como exposto na Figura 10. Esse modelo serviu como base para a composição das animações conforme as instruções de montagem.

Figura 10 – modelo 3D da garra robótica



Fonte: produzido pelos próprios autores

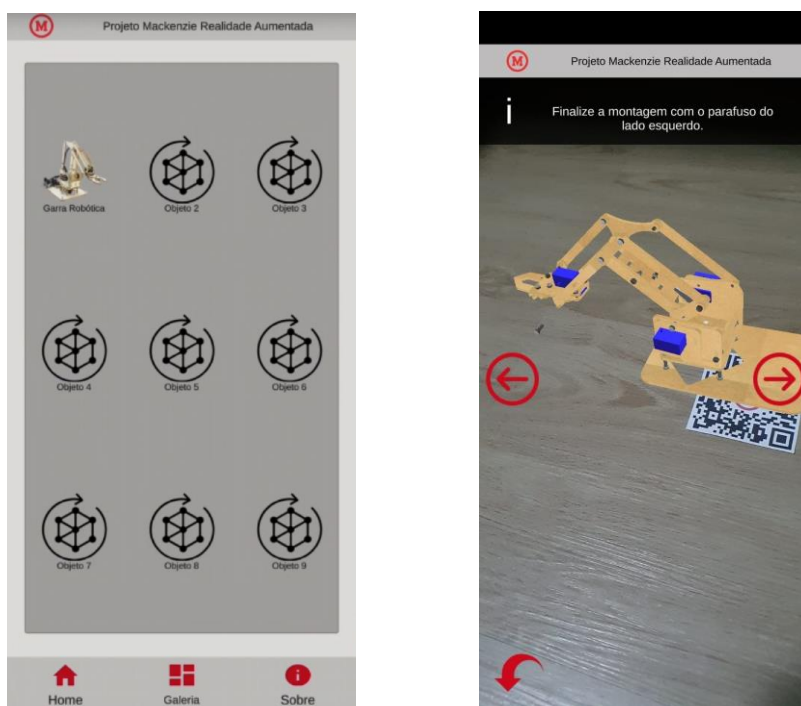
Para a elaboração dessas animações, primeiramente se avaliou a utilização do Blender - programa de código aberto para modelagem, animação, texturização, composição, renderização e edição de vídeo. Por outro lado, esse *software* possui uma complexidade na criação e manipulação das animações, diante disso optou-se por utilizar o módulo de animações do próprio Unity, pois ele permitia a movimentação das peças conforme necessidade das instruções.

Uma vez definida a ferramenta, o modelo da garra foi importado para o Unity na extensão OBJ. As animações foram baseadas em linha do tempo a partir do *Animation Controller*, função pertencente à própria ferramenta.

Já os botões utilizados para o avanço das instruções exigiram a utilização de códigos em C#, os quais possibilitam a troca das animações e dos textos instrutivos, cujos trechos foram armazenados em um arquivo de texto contido no projeto.

Finalizadas essas etapas, o aplicativo pôde ser finalizado e instalado nos dispositivos Android, como exposto na Figura 11.

Figura 11 – funcionamento do aplicativo para Android

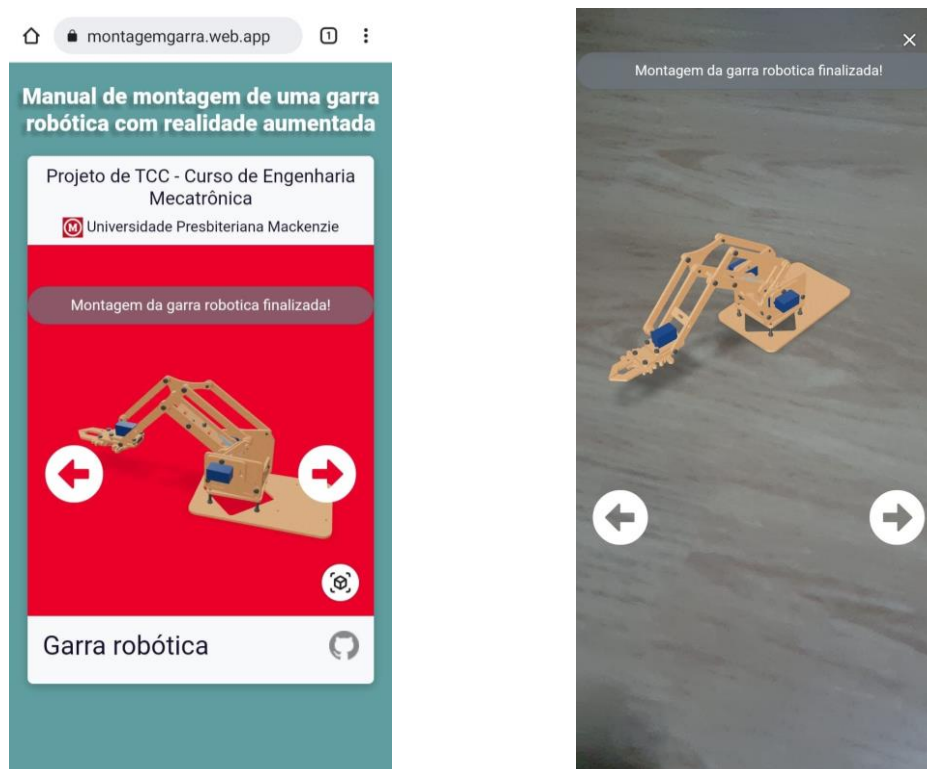


Fonte: produzido pelos próprios autores

Apesar do alcance dos objetivos propostos, notou-se certa complexidade na instalação do aplicativo por outros usuários, visto que ele não se encontrava publicado na loja oficial de aplicativos da Google e não era compatível com dispositivos de sistema operacional iOS. Portanto, decidiu-se iniciar o desenvolvimento de um aplicativo no formato *web*, a fim de garantir melhor compatibilidade entre dispositivos, facilitar o uso e o compartilhamento, bem como evitar a ocupação de memória dos dispositivos dos usuários. Para o desenvolvimento desse segundo aplicativo foi utilizado o SDK WebXR.

A criação da interface do aplicativo se iniciou a partir da construção de uma página simples representada na Figura 12, uma vez que o foco do estudo é a RA. Para aplicação do WebXR se utilizou o pacote *Model-Viewer*, cujo fornecimento é feito pela Google. Com ele foi possível elaborar de maneira simplificada a RA, sem exigência de uma programação complexa.

Figura 12 – funcionamento do aplicativo *Web*



Fonte: produzido pelos próprios autores

Com relação ao uso de modelos 3D e suas animações, o *Model-Viewer* requer o uso de arquivos com extensão GLTF, portanto diferente daquela produzida no Unity anteriormente. Primeiramente se buscou converter os formatos, todavia os produtos obtidos apresentavam erros e mau funcionamento. Sendo assim, optou-se por recriar as animações no Blender no formato adequado, sendo um modelo animado único para cada etapa. A escolha de exportar arquivos separados se deu pela complexidade de uso da ferramenta.

Para viabilizar a troca das animações e instruções, a linguagem Javascript foi utilizada na troca das propriedades do *Model-Viewer*, assim ao avançar a etapa é feita a troca do modelo e da instrução textual. Devido ao uso de modelos individuais, observou-se lentidão na troca das animações em razão do carregamento de um novo arquivo a cada passo.

O WebXR apresentou uma simplicidade no desenvolvimento da solução, sendo utilizado o reconhecimento de planos para posicionamento do objeto virtual. Esse método apresentou uma ancoragem estável e segura, o que evitou o uso do código QR como marcador. Além disso, o WebXR possui por padrão funcionalidades para rotacionar, redimensionar e mover o objeto virtual pelo ambiente. Essas funções se mostraram muito proveitosas para uma melhor visualização de detalhes das animações.

Em virtude das características mencionadas, percebe-se que o aplicativo *web* se provou mais competente para o objetivo deste trabalho, dado que pôde garantir maior compatibilidade, agilidade no compartilhamento e dispensando o armazenamento local no dispositivo. Além disso, ainda apresentou um desenvolvimento facilitado.

Com estes dois aplicativos, é possível verificar que os recentes avanços da RA, apesar de se tratar de uma tecnologia nova, tornaram-na mais acessível aos usuários e desenvolvedores. Além disso, a RA se mostra cada vez mais uma tecnologia eficiente para usos práticos como na montagem de equipamentos, facilitando o entendimento de instruções de forma visual.

5 CONSIDERAÇÕES FINAIS

A partir dos resultados, foi possível compreender o funcionamento da tecnologia de RA, o seu potencial na melhoria das atividades de manutenção e a sua elevada projeção de crescimento no número de usuários e valor de mercado. Também se elucidou como se dá o desenvolvimento de uma aplicação, com ênfase nas tecnologias disponíveis e suas características.

Ademais, concluiu-se com sucesso a construção de dois aplicativos, sendo um em formato instalável em Android e outro para página *web*. Destaca-se que ambos cumpriram com os requisitos de guiar a montagem da garra robótica com instruções visuais e textuais.

Os arquivos de ambos os aplicativos, local e *web*, foram disponibilizados nos seguintes repositórios públicos:

- Aplicativo Android: <https://github.com/GustavoHochgraf/Manual-de-montagem-com-realidade-aumentada>
- Aplicativo *web*: <https://github.com/GustavoHochgraf/MontagemGarraWebApp>
 - Página do aplicativo: <https://montagemgarra.web.app/>

Como sugestão para futuros trabalhos, é indicado o desenvolvimento de uma aplicação *web* compatível com dispositivos iOS, visando atender de forma plena o uso em dispositivos móveis. Para isso, seria necessário converter as animações para a extensão proprietária USDZ e compilá-las com o uso do *software* AR Composer da Apple.

REFERÊNCIAS

- AHIR, K. **Occlusion in Augmented Reality**. [S.l., 2020]. Medium. Disponível em: <https://kumarahir.medium.com/occlusion-in-augmented-reality-2090911b3da1>. Acesso em: 16 mai. 2021.
- APPLE. **Introducing ARKit 4**. [S.l., 2021a]. Apple Developer. Disponível em: <https://developer.apple.com/augmented-reality/arkit/>. Acesso em: 21 mai. 2021
- APPLE. **ARKit**. [S.l., 2021?]. Apple Developer. Disponível em: <https://developer.apple.com/documentation/arkit#overview>. Acesso em: 16 mai. 2021
- AZUMA, R. T. A Survey of Augmented Reality. **Presence: Teleoperators and Virtual Environments**, [S.l.], v. 6, n. 4, p. 355-385, 1997. Disponível em: <https://direct.mit.edu/pvar/article/6/4/355/18336/A-Survey-of-Augmented-Reality>. Acesso em: 4 abr. 2021

- AZUMA, R. T. The Most Important Challenge Facing Augmented Reality. **Presence: Teleoperators and Virtual Environments**, [S.l.], v. 25, n. 3, p. 234-238, 2016. Disponível em: <https://direct.mit.edu/pvar/article/25/3/234/58932/The-Most-Important-Challenge-Facing-Augmented>. Acesso em: 29 mar. 2021.
- BOSCH AUTO TOOLS. **Bosch Augmented Reality in Use with Microsoft HoloLens 2**. [S.l., 2019]. Youtube. Disponível em: https://www.youtube.com/watch?v=4YYsu4F2-vM&ab_channel=BoschAutoTools. Acesso em: 11 abr. 2021.
- CARDOSO, L. F. DE S.; MARIANO, F. C. M. Q.; ZORZAL, E. R. A Survey of Industrial Augmented Reality. **Computer & Industrial Engineering**. [S.l.], v. 139, artigo 106159, 2020. Disponível em: <https://www.sciencedirect.com/science/article/abs/pii/S036083521930628X>. Acesso em: 2 mai. 2021
- CHOI, H. B.; LIM, K. W.; KO, Y. B. Improved Virtual Anchor Selection for AR-assisted Sensor Positioning in Harsh Indoor Conditions. **2020 Global Internet of Things Summit (GIoTS)**. Dublin, 2020. Disponível em: <https://ieeexplore.ieee.org/abstract/document/9119679/references#references>. Acesso em: 24 abr. 2021
- DIAZ, J. **From Apple Glasses to HoloLens 2: AR glasses you can buy now (and soon)**. [S.l., 2020]. Tom's Guide. Disponível em: <https://www.tomsguide.com/reference/ar-glasses>. Acesso em: 23 mai. 2021.
- FRAGA-LAMAS, P. *et al.* **A Review on Industrial Augmented Reality Systems for the Industry 4.0 Shipyard**. IEEE Access, [S.l.], v. 6, p. 13358-13375, 2018. Disponível em: <https://ieeexplore.ieee.org/document/8298525>. Acesso em: 5 mai. 2021
- GEVORGYAN, M. **Triple-E Start-Up: Making Augmented Reality Accessible**. [Armênia, 2017]. Hetq Online. Disponível em: <https://hetq.am/en/article/81460>. Acesso em: 9 mai. 2021
- GOOGLE ARCORE. **Dispositivos com suporte ARCore**. [S.l., 2020]. Google Developers. Disponível em: <https://developers.google.com/ar/devices>. Acesso em: 20 mai. 2021
- HAMMADY, R.; MA, M.; POWELL, A. **User Experience of Markerless Augmented Reality Applications in Cultural Heritage Museums: 'MuseumEye' as a Case Study**. International Conference on Augmented Reality, Virtual Reality and Computer Graphics (AVR 2018), Ugento, p.349-369, 2018. Disponível em: https://link.springer.com/chapter/10.1007%2F978-3-319-95282-6_26. Acesso em: 03 mai. 2021.
- HANAFI, A.; ELAACHAK, L.; BOUHORMA, M. **A Comparative Study of Augmented Reality SDKs to Develop an Educational Application in Chemical Field**. NISS19: Networking, Information Systems & Security, Rabat, artigo 53, p. 1-8, 2019. Disponível em: <https://dl.acm.org/doi/10.1145/3320326.3320386>. Acesso em: 05 mai. 2021.
- HARDIANTO, Z. I. P.; KARMILASARI. **Analysis and Design of User Interface and User Experience (UI / UX) E-Commerce Website PT Pentasada Andalan Kelola Using Task System Centered Design (TCSD) Method**. Proceedings of 2019 4th International Conference on Informatics and Computing (ICIC 2019), Indonésia, p.1-8, 2019. Disponível em: <https://ieeexplore.ieee.org/abstract/document/8985854>. Acesso em: 23 mai. 2021
- HERPICH, F.; GUARESE, R. L. M.; TAROUCO, L. M. R. A Comparative Analysis of Augmented Reality Frameworks Aimed at the Development of Educational Applications. **Creative Education**, [S.l.], v. 08, n. 09, p. 1433–1451, 2017. Disponível em: <https://www.scirp.org/journal/paperinformation.aspx?paperid=77994>. Acesso em 03 mai. 2021.
- JULIANI, A. *et al.* **Unity: A General Platform for Intelligent Agents**. [S.l.], v. 2, 2020. Disponível em: <https://arxiv.org/abs/1809.02627v2>. Acesso em: 14 mai. 2021
- KLOBERDANZ, K. **Smart Specs: OK Glass, Fix This Jet Engine**. [S.l., 2017]. General Electric. Disponível em: <https://www.ge.com/news/reports/smart-specs-ok-glass-fix-jet-engine>. Acesso em: 3 abr. 2021.
- MACLNTYRE, B.; SMITH, T. F. **Thoughts on the Future of WebXR and the Immersive Web**. 2018 IEEE International Symposium on Mixed and Augmented Reality Adjunct (ISMAR-Adjunct). Munique, p. 338-342, 2018. Disponível em: <https://ieeexplore.ieee.org/document/8699303/>. Acesso em: 6 mai. 2021

- MATARIĆ, Maja J. **Introdução à Robótica**. São Paulo, SP: Editora Unesp, 2014. Disponível em: <https://app.minhabiblioteca.com.br/#/books/9788521208549>. Acesso em: 22 maio 2021.
- MDN CONTRIBUTORS. **WebXR Device API**. [S.l., 2021]. MDN Web Docs. Disponível em: https://developer.mozilla.org/en-US/docs/Web/API/WebXR_Device_API%0A. Acesso em: 20 mai. 2021
- QIAO, X. *et al.* Web AR: A Promising Future for Mobile Augmented Reality—State of the Art, Challenges, and Insights. **Proceedings of the IEEE**, [S.l.], v. 107, n. 4, p.651-666, 2019. Disponível em: <https://ieeexplore.ieee.org/document/8643424/>. Acesso em: 21 mai. 2021
- ROBERTO, R. *et al.* Geometrical and Statistical Incremental Semantic Modeling on Mobile Devices. **Computers & Graphics**, [S.l.], v. 84, p. 199-211, 2019. Disponível em: <https://www.sciencedirect.com/science/article/abs/pii/S0097849319301542>. Acesso em: 25 abr. 2021
- SARACCO, R. **AR market gets a boost from pandemic**. [S.l., 2020]. IEEE Future Directions. Disponível em: <https://cmte.ieee.org/futuredirections/2020/07/19/ar-market-gets-a-boost-from-pandemic/>. Acesso em: 9 mai. 2021
- SCHWALD, B.; LAVAL, B. DE. **An Augmented Reality System for Training and Assistance to Maintenance in the Industrial Context**. International Conference in Central Europe on Computer Graphics, Visualization and Computer Vision'2003, República Tcheca, 2003. Disponível em: http://wscg.zcu.cz/wscg2003/Papers_2003/I23.pdf. Acesso em: 3 abr. 2021
- TAHERI, H.; XIA, Z. C. SLAM; Definition and Evolution. **Engineering Applications of Artificial Intelligence**, [S.l.], v.97, artigo 104032, 2021. Disponível em: <https://www.sciencedirect.com/science/article/abs/pii/S0952197620303092?via%3Dihub>. Acesso em: 25 abr. 2021
- TORI, R.; KIRNER, C.; SISCOUTTO, R. (EDS.). **Fundamentos e Tecnologia de Realidade Virtual e Aumentada**. Belém, PA: Editora SBC, 2006. Disponível em: <https://pcs.usp.br/interlab/wp-content/uploads/sites/21/2018/01/Fundamentos_e_Tecnologia_de_Realidade_Virtual_e_Aumentada-v22-11-06.pdf>. Acesso em: 24 abr. 2021
- UNITY TECHNOLOGIES. **Vuforia hardware and software requirements**. [S.l., 2019]. Unity Documentation. Disponível em: https://docs.unity3d.com/2019.1/Documentation/Manual/vuforia_requirements.html. Acesso em: 20 mai. 2021
- UNITY TECHNOLOGIES. **Unity User Manual (2019.2)**. [S.l., 2020]. Unity Documentation. Disponível em: https://docs.unity3d.com/2019.2/Documentation/Manual/vuforia_tips.html. Acesso em: 16 mai. 2021.
- WADHWA, N.; ZHANG, Y. **Improvements to Portrait Mode on the Google Pixel 4 and Pixel 4 XL**. [S.l., 2019]. Google AI Blog. Disponível em: <https://ai.googleblog.com/2019/12/improvements-to-portrait-mode-on-google.html>. Acesso em: 23 mai. 2021
- WATSON, G. *et al.* **The Effect of Using Animated Work Instructions Over Text and Static Graphics When Performing a Small Scale Engineering Assembly**. Collaborative Product and Service Life Cycle Management for a Sustainable World - Proceedings of the 15th ISPE International Conference on Concurrent Engineering (CE 2008), Belfast, p. 541-550, 2008. Disponível em: https://link.springer.com/chapter/10.1007%2F978-1-84800-972-1_51. Acesso em: 2 abr. 2021
- YOPPER. **UX Design: o que é e como impacta nas suas conversões**. [S.l., 2020]. Disponível em: <https://blog.yopper.com.br/ux-design-o-que-e-e-como-impacta-nas-suas-conversoes/>. Yopper. Acesso em: 9 mai. 2021

AGRADECIMENTOS

Aos familiares pelo apoio e incentivo que serviram de alicerce para nossas realizações.

Aos professores pela excelência no ensino e pelas valiosas contribuições ao longo de nossa formação acadêmica e profissional.