UNIVERSIDADE REGIONAL INTEGRADA DO ALTO URUGUAI E DAS MISSÕES
CAMPUS DE ERECHIM
DEPARTAMENTO DE ENGENHARIAS E CIÊNCIA DA COMPUTAÇÃO
CURSO DE CIÊNCIA DA COMPUTAÇÃO

CRISTIAN ABRAMCHUK

PROJEÇÃO DE DESIGN DE INTERIORES DE UMA PLANTA BAIXA COM REALIDADE AUMENTADA PERMITINDO ALTERAÇÕES EM TEMPO REAL UTILIZANDO MARCADORES

ERECHIM - RS
2020

CRISTIAN ABRAMCHUK

PROJEÇÃO DE DESIGN DE INTERIORES DE UMA PLANTA BAIXA COM REALIDADE AUMENTADA PERMITINDO ALTERAÇÕES EM TEMPO REAL UTILIZANDO MARCADORES

Trabalho de Conclusão de Curso apresentado como requisito parcial à obtenção do grau de Bacharel, Departamento de Engenharias e Ciência da Computação da Universidade Regional Integrada do Alto Uruguai e das Missões Campus de Erechim.

Orientador: Daniel Menin Tortelli

ERECHIM - RS
2020

AGRADECIMENTOS

Agradeço primeiramente aos meus pais e amigos por todo o apoio que me deram durante a realização deste curso.

Também agradeço a todos os professores do curso que, de alguma forma, puderam passar um pouco de seu conhecimento durante todo esse período.

Agradecimento em especial ao meu orientador e professor, Daniel Menin Tortelli, por ter me orientado neste trabalho.

RESUMO

Modelar o design de interiores de imóveis é algo trabalhoso, tendo em vista que a possível insatisfação do cliente com o resultado mal sucedido pode acarretar em trabalho constante de remodelagem e, em casos extremos, a perda de interesse do cliente. Aplicativos com realidade aumentada tem sido desenvolvidos para facilitar o trabalho dos designers e proporcionar uma experiência mais interativa e atrativa para o cliente, através da projeção 3D da planta baixa. O principal objetivo deste trabalho foi a confecção de um aplicativo mobile em Android, capaz de criar uma projeção 3D de uma planta baixa para que seja possível inserir, visualizar e reposicionar os móveis no ambiente através de marcadores, auxiliando no design de interiores.

Palavras-chave: Design de interiores. Realidade aumentada. Projeção 3D.

ABSTRACT

Modeling the interior design of real estate is somewhat laborious, considering that the possible customer dissatisfaction with the unsuccessful result can result in constant remodeling work and, in extreme cases, the loss of the client's interest. Augmented reality applications have been developed to facilitate the work of designers and provide a more interactive and attractive experience for the client, through the 3D projection of the floor plan. The main objective of this work was the making of a mobile application on Android, capable of creating a 3D projection of a floor plan so that it is possible to insert, visualize and reposition the furniture in the environment through markers, assisting in the interior design.

Keywords: Interior Design. Augmented reality. 3D Projection.

LISTA DE ILUSTRAÇÕES

LISTA DE ABREVIATURAS E SIGLAS

API	*Application Programming Interface*
FPS	*Frames per Second*
GPS	*Global Positioning System*
HMD	*Head Mounted Displays*
IDE	*Integrated Development Environment*
OpenGL	*Open Graphics Library*
RA	Realidade Aumentada
RV	Realidade Virtual
SDK	*Software Development Kit*
UWP	*Universal Windows Platform*
VA	Virtualidade Aumentada

SUMÁRIO

1 INTRODUÇÃO

A realidade aumentada tem se mostrado muito presente nas áreas de arquitetura e de design de interiores, trazendo consigo praticidade e agilidade na desenvoltura dos projetos. Segundo o site Terra (2017), o mercado de decoração subiu 23% no primeiro trimestre de 2017 em comparação ao mesmo período de 2016.

Tendo em vista que o mercado de decoração vem crescendo, novos problemas vão surgindo ao longo do tempo. Um bom exemplo é a grande demanda de projetos que, caso não forem feitos de maneira assertiva, deverão ser refeitos, acarretando em atraso nas entregas.

Neste trabalho foi desenvolvido um aplicativo *mobile* utilizando realidade aumentada para servir como uma ferramenta capaz de agilizar o trabalho dos designers de interiores na hora de desenvolver o seu projeto, dando a eles várias opções de modelos de móveis e a possibilidade de alteração do ambiente em tempo real. Logo, ao utilizar o aplicativo, os designers tendem ser mais assertivos, pois conseguem mostrar para seus clientes um *preview* mais próximo possível do ambiente real.

Dar maior flexibilidade para o designer na hora da modelagem do design do interior de um cômodo, proporcionando uma visão ampla do cenário em 3 dimensões, permite validar conceitos e atender às especificações do cliente, ao mesmo tempo que as exibe. Proporcionar ao cliente final uma experiência muito mais próxima da realidade de como o design da sua casa/apartamento ficará antes mesmo da conclusão do projeto.

A utilização da realidade aumentada é de suma importância para o dinamismo da aplicação, onde os designers poderão posicionar os objetos e escolher entre seus modelos, buscando sempre a melhor combinação.

Os conteúdos deste trabalho estão organizados em 6 capítulos, onde cada capítulo descreve uma parte essencial do mesmo.

O segundo capítulo apresenta o Continuum de Virtualidade, conceitos de realidade virtual, conceitos e realidade aumentada aprofundando em suas técnicas de interações e dispositivos de visualização.

O terceiro capítulo apresenta conceitos de design de interiores, arquitetura e urbanismo, diferença entre arquiteto e designer e a realidade aumentada no design de interiores.

O quarto capítulo apresenta as ferramentas e SDKs utilizados para o desenvolvimento da aplicação. Primeiramente, é abordada a *Game Engine* Unity onde foi possível criar a cena que unificou todas as etapas do desenvolvimento e gerou a aplicação. Posteriormente é apresentado o Vuforia, responsável por detectar as imagens para aplicar os recursos de realidade aumentada. O 3ds *Max*, software de modelagem tridimensional, utilizado para unificação das partes da planta baixa. E, por último, o SDK do Android, responsável por compilar a aplicação.

O quinto capítulo descreve as etapas de desenvolvimento do aplicativo *mobile*, que inicia pela confecção dos marcadores, configuração da chave do Vuforia, do banco de dados

e uploads das imagens. Posteriormente foi criada a planta baixa com o auxílio da Unity e do 3ds *Max* para unificação das partes que compõem a planta. Em seguida, é descrito o desenvolvimento da aplicação final, como foi feita integração e unificação das partes desenvolvidas anteriormente. Por último é descrito o funcionamento da aplicação de realidade aumentada, demonstrando como utilizá-lo

O sexto e último capítulo, apresenta as conclusões obtidas ao finalizar o trabalho, bem como as propostas de melhorias e trabalhos futuros.

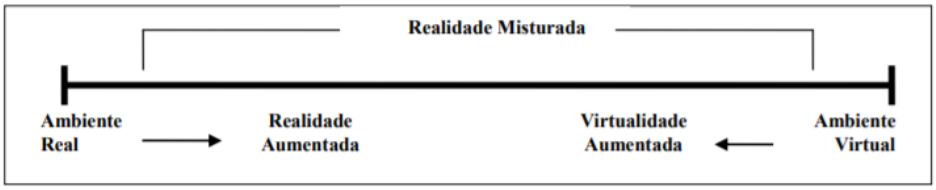
2 REALIDADE AUMENTADA

Neste capítulo será apresentado o Continuum de Virtualidade (gráfico mostrado no artigo *"A Taxonomy of Mixed Reality Visual Displays"* de 1994 feito por Milgram e Kishino) e a explicação sobre os seus ambientes aprofundando e dando mais ênfase sobre a realidade aumentada. O gráfico da taxonomia está representado na Figura 1.

2.1 Continuum de Virtualidade

O continuum de virtualidade, proposto por Milgram (1994), foi designado para caracterizar e diferenciar os tipos de ambientes de realidade misturada. Conforme a Figura 1 é possível observar que nos extremos possuem os ambientes real e virtual. Quando o ambiente é mais voltado para o virtual, ele é denominado de Virtualidade Aumentada (VA), e quando ele é voltado para o real ele se denomina Realidade Aumentada (RA). A Realidade Misturada é a união da VA com a RA.

Figura 1 – Continuum de Virtualidade



Fonte: (RODELLO, 2010)

2.2 Conceitos de Realidade Virtual

A realidade virtual (RV) é um ambiente gerado computacionalmente com recursos tridimensionais podendo simular objetos e ambientes do mundo real, onde o usuário pode ser imergido com alguns dispositivos eletrônicos. Na Figura 2 pode-se visualizar o *oculus rift*, um dispositivo para imersão na RV.

Figura 2 – *Oculus Rift*

Fonte: (BARROS, 2015)

A utilização do *oculus rift* serve para a projeção das imagens do ambiente virtualizado diretamente nos olhos do usuário, se tornando assim, os olhos da pessoa que o estiver usando dando a sensação de estar dentro do ambiente.

> Outra definição é o uso de computadores e interfaces com o usuário para criar o efeito de mundos tridimensionais que incluem objetos interativos com uma forte sensação de presença tridimensional [Bryson, 1996]. Além disso, a RV engloba um conjunto de técnicas e ferramentas gráficas 3D que permite aos usuários interagir com um ambiente gerado por computador, em tempo real, com uma pequena ou nenhuma consciência de que está usando uma interface usuário-computador [Leston, 1996]. (NETTO, 2002)

A realidade virtual pode ser aplicada em diversas áreas como jogos, simulação de eventos, criação de cenários (planta de uma casa), na saúde com treinamentos médicos, entre outros.

2.3 Conceitos de Realidade Aumentada

A realidade aumentada é a junção do mundo virtual com o mundo real através de imagens geradas computacionalmente, dando uma experiência interativa do mundo virtual no mundo real.

Cardoso (2007) define a realidade aumentada à associação dos dados computacionais ao mundo real, com o objetivo de uma interação simples e prática pelo usuário ao manuseio dos elementos virtuais inseridos no nosso mundo.

> A Realidade Aumentada proporciona ao usuário uma interação segura e agradável, eliminando em grande parte a necessidade de treinamento, pelo fato de trazer para o ambiente real os elementos virtuais, enriquecendo e ampliando a

> visão que ele tem do mundo real. Para que isso se torne possível, é necessário combinar técnicas de visão computacional, computação gráfica e realidade virtual, o que origina como resultado a correta sobreposição de objetos virtuais no ambiente real. (AZUMA; BAILLOT; BEHRINGER, 2001).

Dito isso, percebe-se que a realidade aumentada vai muito mais além do que "simplesmente" adicionar objetos computacionais em ambientes reais, mas sim de nos dar a possibilidade de interação com os mesmos através das mãos com algum dispositivo tecnológico. Na Figura 3 é exibida uma ilustração da realidade aumentada utilizada no jogo Pokémon GO, onde o usuário captura pokémons utilizando a câmera do celular interagindo com objetos virtuais.

Figura 3 – Pokemon GO

Fonte: (SOUZA, 2018)

A RA surgiu como um conceito que se completa à Realidade Virtual (RV), uma vez que considera o real como o ambiente que se envolve o usuário, ao contrário da realidade virtual, em que o usuário é envolto por um ambiente sintético e simulado. Como na RA não se tem o controle total do ambiente, ela é considerada mais complexa do que a RV, na qual o ambiente é mantido sob total controle (BIMBER; RASKAR, 2005).

Segundo Azuma, Baillot e Behringer (2001), um sistema de realidade aumentada deve ter três propriedades: combinar objetos reais e virtuais no ambiente real; ser interativo em tempo real e alinhar objetos reais e virtuais uns com os outros, colocando-os no mesmo plano.

2.3.1 Técnicas de Interação com Realidade Aumentada

De acordo com LaViola et al. (2017), "interação é um método que permite a um usuário realizar uma tarefa através da interface do usuário. Uma técnica de interação inclui tanto componentes de *hardware* (dispositivos de entrada/saída) quanto de software. As técnicas de interação utilizadas nos componentes de software são responsáveis por mapear a informação de

um dispositivo de entrada em alguma ação dentro do sistema, e por mapear a saída do sistema de forma que esta possa ser interpretada pelos dispositivos de saída".

A diferença entre as formas de interação com os elementos virtuais é percebida ao se analisar os equipamentos envolvidos e/ou necessários para permitir acesso ao conteúdo e à maneira como ocorre a entrada de dados para a interpretação da máquina e inserção das informações virtuais ao ambiente físico.

Broll et al. (2005), classifica as técnicas de interação para AR em quatro categorias principais, de acordo com a tarefa realizada pelo usuário, como:

- **Interação espacial (*spatial interaction*)**: ocorre a manipulação das propriedades espaciais dos objetos físicos, em geral, por meio de interfaces tangíveis que permitem a interação através de objetos reais da cena;
- **Interação baseada em comandos (*command based interaction*)**: entrada de dados se dá através do rastreamento de gestos simbólicos ou alguma outra forma de comando, como a voz, que representam instruções a serem interpretadas;
- **Interação por controle virtual (*virtual control interaction*)**: manipulação de símbolos gráficos tridimensionais apresentados ao usuário e que permitem a comunicação com a máquina;
- **Interação por controle físico (*physical control interaction*)**: a comunicação homem-máquina ocorre por meio de ferramentas físicas ou painéis de controle que possibilitam acesso tanto ao ambiente físico como aos objetos virtuais da cena;

De acordo com Cuperschmid, Freitas e Ruschel (2012), não há um consenso geral sobre a classificação das técnicas de interação. Encontram-se variantes desta ideia, sempre no intento de definir as formas de relação entre o usuário e os computadores

2.3.2 Dispositivos de Visualização

Existem vários dispositivos de visualização que têm sido usados para RA. Bimber e Raskar (2005) categorizam os dispositivos de visualização como *Head-Mounted Displays*, *Hand-Held Displays* (*smartphones* e *tablets*) e dispositivos espaciais.

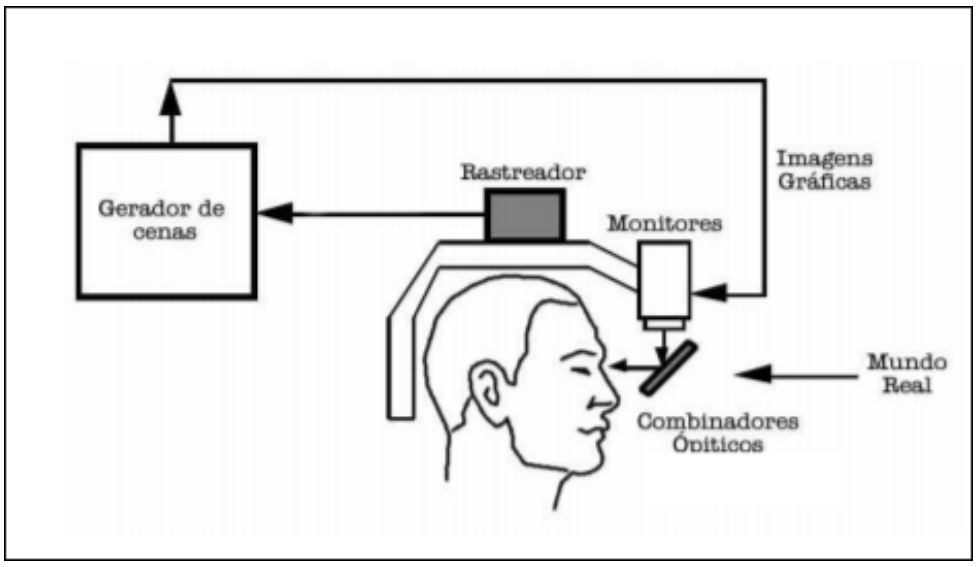
2.3.2.1 *Head Mounted Displays*

De acordo com Bimber e Raskar (2005), os HMDs (*Head Mounted Displays*) fazem parte de uma classe de dispositivos imersivos, utilizados em RA, caracterizados pela habilidade de permitir ao observador ver diretamente, através da mídia, o mundo a seu redor, alcançando a máxima possibilidade de presença e imersão. Existem duas tecnologias diferentes de HMD: baseado em ótica e baseado em vídeo.

Sistemas baseado em ótica usam dispositivos formados por lentes que permitem que o usuário veja o mundo real com o mundo virtual projetado nas lentes, que são posicionadas em frente dos olhos. As lentes translúcidas permitem que o usuário olhe diretamente por ela para

ver o mundo real. Assim, o usuário vê o mundo virtual superposto ao físico. (CUPERSCHMID; FREITAS; RUSCHEL, 2012). Na Figura 4 encontra-se um diagrama representando o sistema e na Figura 5 um exemplo de óculos usado para esse tipo de sistema.

Figura 4 – Diagrama do sistema de visão ótica direta



Fonte: (KIRNER; ZORZAL, 2005)

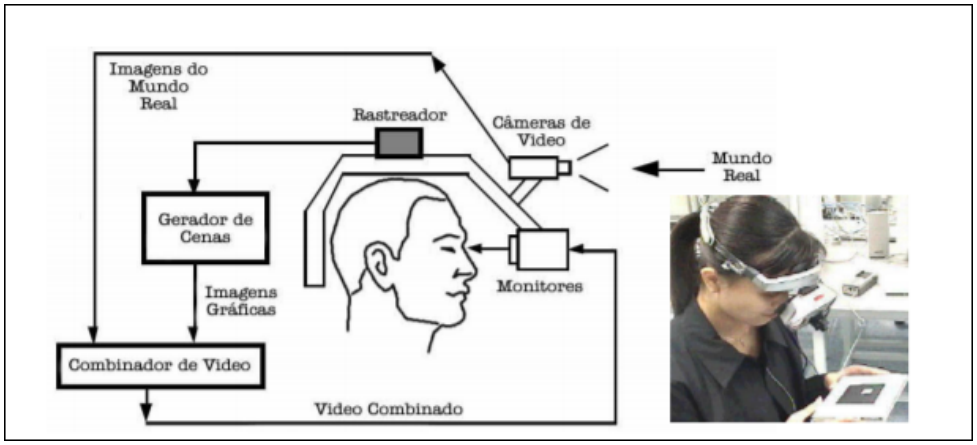
Figura 5 – Óculos usado em sistemas de visão ótica direta

Fonte: (CRAIG, 2016)

Sistemas baseado em vídeo usam câmeras de vídeo que fornecem ao usuário uma visão do mundo real. O vídeo dessas câmeras é combinado com imagens virtuais, gerando a

cena misturada entre o mundo real e virtual. O resultado pode ser visualizado por meio do *Head Mounted Display* que fornece uma visão fechada por meio de monitores em frente dos olhos dos usuários. (CUPERSCHMID; FREITAS; RUSCHEL, 2012). Na Figura 6 encontra-se um diagrama representando o sistema juntamente com um exemplo de objeto real

Figura 6 – Diagrama e dispositivo do sistema de visão direta por vídeo

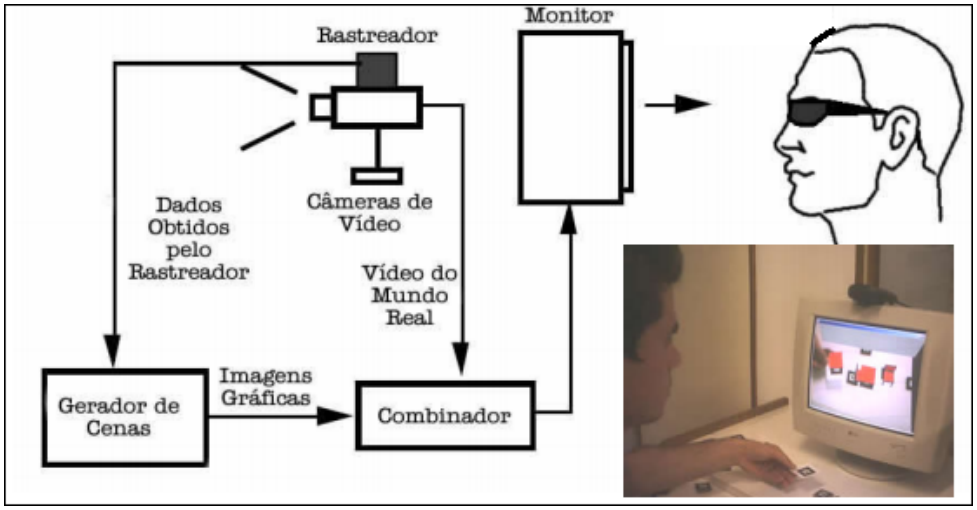


Fonte: (KIRNER; ZORZAL, 2005)

2.3.2.2 *Smartphones* e *Tablets*

Os dispositivos móveis combinam, em um único aparelho, o processador, a memória, a visualização, a tecnologia de interação, GPS e compasso digital. Nesses sistemas, câmeras integradas capturam o vídeo do ambiente real e processam a mistura dos ambientes antes de exibi-las. (BIMBER; RASKAR, 2005). Na Figura 7 encontra-se um diagrama representando o sistema e na Figura 8 um tablet utilizando um software de RA.

Figura 7 – Diagrama do sistema de visão por vídeo baseado em monitor



Fonte: (KIRNER; ZORZAL, 2005)

Figura 8 – *Tablet* utilizando aplicação de RA

Fonte: (ZAPP2PHOTO, 2017)

2.3.2.3 Dispositivos Espaciais

De acordo com Bimber e Raskar (2005), os dispositivos espaciais integram o elemento virtual ao ambiente real de maneira que o usuário não necessite usar (vestir) equipamentos, pois estes estão conectados ao ambiente. Nesta categoria estão presentes os visores ópticos espaciais transparentes, exibições espaciais baseadas em projeção e visores transparentes de vídeo baseados em tela.

- **Visores ópticos espaciais transparentes**: geram imagens que são alinhadas com o ambiente físico. Utilizam combinação ótica espacial como espelhos planos ou curvos, telas transparentes ou hologramas ópticos;
- **Exibições espaciais baseadas em projeção**: usam a projeção frontal para projetar imagens diretamente nas superfícies de objetos físicos. Para essa técnica, são usados projetores estáticos individuais, ou estéreos, ou múltiplos;
- **Visores transparentes de vídeo baseados em tela**: fazem uso de uma câmera para captar o mundo real, processam as informações e exibem as imagens misturadas em um monitor regular ou projetam a cena composta em uma tela plana (como o monitor de um computador ou uma tela com projeção de datashow);

Conforme Cuperschmid, Freitas e Ruschel (2012) essa forma de experimentar RA fornece um baixo nível de imersão, o que, nesses ambientes, é comumente ligado ao campo de visão do observador. Nesse caso, o campo de visão é limitado e restrito ao tamanho do monitor ou da tela. Essa é uma técnica comum para experimentar RA, pois não demanda equipamentos dispendiosos, somente um computador, web câmera e monitor, oferecendo, assim, o melhor custo-benefício.

3 DESIGN DE INTERIORES

Neste capítulo são abordados conceitos de design de interiores, arquitetura e urbanismo e a diferença entre as suas funções no mercado de trabalho.

3.1 Conceito de Design de Interiores

Design de interiores é a área em que se planeja e organiza o interior de um ambiente, harmonizando móveis, objetos e acessórios de um determinado espaço buscando beleza e praticidade.

> Numa acepção mais ampla, significa o planejamento, a organização, a decoração e a composição do layout espacial de mobiliário, equipamentos, acessórios, objetos de arte, etc., dispostos em espaços internos habitacionais, de trabalho, cultura, lazer e outros semelhantes, como veículos aéreos, marítimos e terrestres – aviões, navios, trens, ônibus e automóveis, por exemplo. (FILHO, 2006).

O design de interiores está presente em ambientes de trabalhos em geral, auxiliando na melhor decoração do espaço, proporcionando conforto e lazer.

Segundo Gubert (2011), o design é um processo que visa alcançar um objetivo funcional ou estético através da organização de materiais, com suas cores, traços, texturas e formas diferentes.

O designer define cores, iluminação, texturas, posicionamento dos móveis, acabamentos, entre tantas outras coisas para tornar o ambiente confortável e funcional. O Guia do Estudante (2019), cita em alguns tópicos o que o designer faz no seu dia a dia.

- **Desenho de móveis**: projetar e criar o móvel de acordo com o espaço disponível e a necessidade do cliente;
- **Decoração e paisagismo**: cuidar do posicionamento dos móveis e acessórios nos ambientes internos e nos externos organizar, projetar deixando-os mais bonitos com plantas e jardins;
- **Gerenciamento**: fazer orçamentos de móveis, acessórios, pinturas, pisos e mão de obra;
- **Projeto**: organização do ambiente prezando as necessidades do cliente ou da empresa. Elaboração de plantas e maquetes para melhor entendimento do design no ambiente selecionado;
- **Sustentabilidade**: preza os cuidados ao meio ambiente e economia nas escolhas dos materiais e a redução do impacto ambiental;

Portanto, pode-se dizer que o designer é de extrema importância na hora de planejar a decoração de um ambiente, seja ele doméstico ou empresarial. Para Fiorini (2020), os materiais expressam emoções e sentimentos. O designer é o responsável por manusear e organizar esses objetos em um espaço para que possa sentir essas emoções e a interação entre indivíduos.

Dessa forma, conclui-se que o profissional de designer cuida dos aspectos dos ambientes e não da sua estrutura, não tendo autoridade para, por exemplo, ordenar a derrubada de uma parede, pois isso seria uma função do arquiteto.

3.2 Conceito de Arquitetura e Urbanismo

A arquitetura é um processo artístico, organizado, criativo e técnico na hora da elaboração e projeção de edifícios e outros espaços urbanos em geral, buscando a melhor ocupação do solo, melhor iluminação, acústica, manutenções simples e práticas e menos impactos ambientais.

> Nos dias atuais, a arquitetura pode ser definida como a relação entre o homem e o espaço, ou melhor, a forma como ele interfere no meio criando condições estéticas e funcionais favoráveis para habitação, utilização e organização dos ambientes. (DECORAFACIL, 2019)

"Arquitetura é antes de mais nada construção, mas, construção concebida com o propósito primordial de ordenar e organizar o espaço para determinada finalidade e visando a determinada intenção". (COSTA, 1940)

Percebe-se que arquitetos e designers possuem características semelhantes, porém, embora semelhantes, suas funções são diferentes e, por isso, é necessário ambos na construção e planejamento do ambiente e saber diferenciá-los.

3.3 Arquiteto e Designer

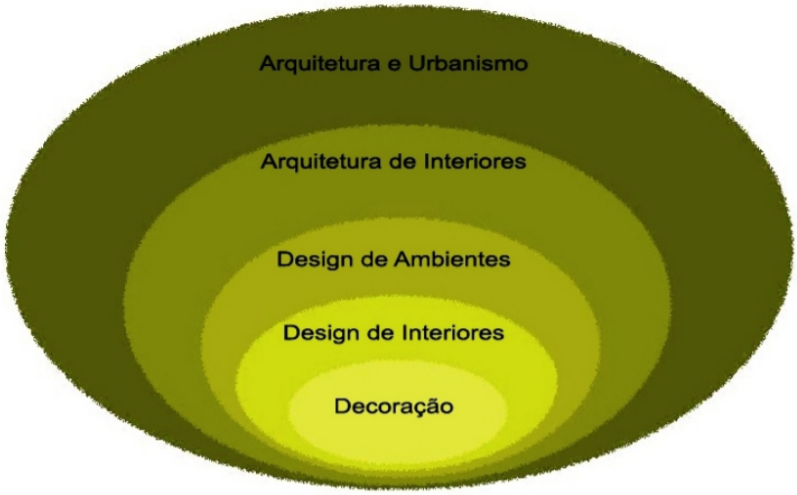
Embora ambas as áreas possuem características em comum, os profissionais da arquitetura, ou melhor, a área em si é muito mais abrangente. Segundo a lei n° 13.369/12, que reconhece o profissional de design fala o seguinte no Art. 2°, "Designer de interiores e ambientes é o profissional que planeja e projeta espaços internos, visando ao conforto, à estética, à saúde e à segurança dos usuários, respeitadas as atribuições privativas de outras profissões regulamentadas em lei".

A lei n° 5.194/24 que reconhece o arquiteto cita o seguinte, "As profissões de engenheiro, arquiteto e engenheiro-agrônomo são caracterizadas pelas realizações de interesse social e humano". Dentre as citações da lei, as realizações dos interesses nos meios de locomoção, edificações, aspectos técnicos e artísticos entre outros.

Portanto, percebe-se que o profissional da arquitetura possui uma área de atuação muito mais abrangente do que o designer. Na Figura 9 pode-se notar a tamanha diferença entre as duas áreas com diversas subáreas entre elas.

Por sua vez, percebe-se que designer tem seu foco voltado para o interior do ambiente, não se limitando só a isso, e o arquiteto tem seu foco na parte externa de uma construção.

Figura 9 – Comparativo entre subdivisões da Arquitetura



Fonte: (BARROS, 2015)

3.4 Realidade Aumentada no Design de Interiores

Segundo Romao (2013), a realidade aumentada torna-se cada vez mais uma ferramenta de muita produtividade para o design. Seu campo de aplicações e o seu uso tem se mostrado cada vez mais valorizado, pois permite que o designer visualize e interaja com os seus projetos de uma forma mais intuitiva, permitindo também mostrar o layout de um ambiente para o cliente.

> Um uso interessante de tecnologia que é especialmente relevante para a indústria de design de interiores e arquitetura é a realidade aumentada, onde podemos testar cores, texturas, móveis e objetos nas nossas salas antes de realizarmos as compras. A realidade aumentada já é possível em grandes varejistas internacionais do setor, como Ikea, e deve se expandir cada vez mais facilitando as escolhas, aumentando o universo de possibilidades e também estreitando a relação entre os profissionais e clientes (GRIMBERG, 2017).

Cyrela (2020) cita benefícios que a realidade aumentada traz aos projetos de design tais como:

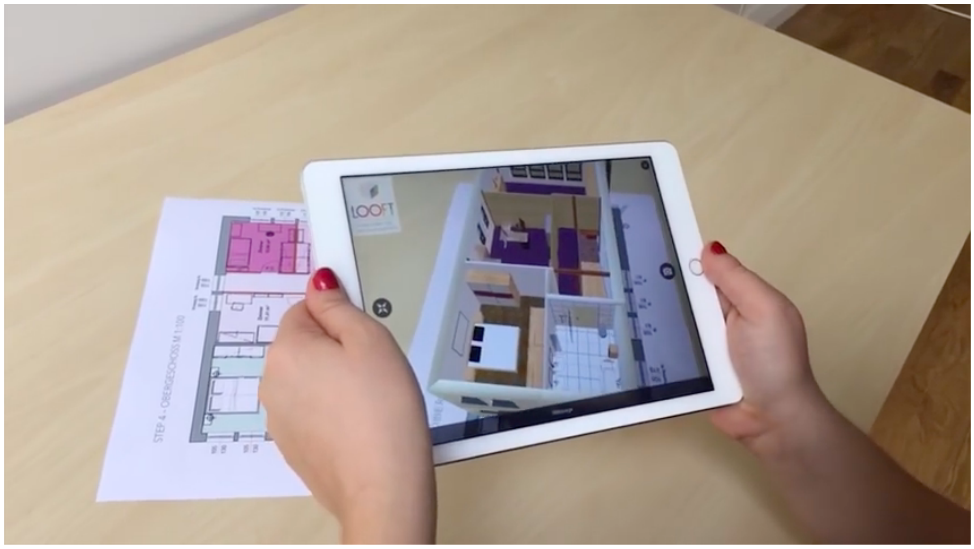
- Conhecer possibilidades decorativas;
- Economia de recursos;
- Segurança nas escolhas;
- Conclusão mais rápida de projetos.

Segundo o site da Archademy (2020), com a tecnologia da realidade aumentada, é possível antecipar o resultado final a partir de uma experiência altamente inovadora, agregando muito mais credibilidade e redução de erros ou dúvidas por parte de clientes e de outros responsáveis pelo projeto.

3.4.1 Identificadores ou Marcadores

Alguns aplicativos de realidade aumentada para design de interiores utilizam marcadores ou identificadores para posicionamento dos objetos a serem representados. Na Figura 10 podemos visualizar os desenhos da planta baixa servindo como identificadores para reendenização dos objetos no tablet.

Figura 10 – Projeção da planta baixa



Fonte: (HALO, 2020)

4 FERRAMENTAS E SDKs

Neste capítulo são apresentadas as ferramentas e tecnologias que foram utilizadas no desenvolvimento da aplicação.

4.1 Unity 3D

Unity é uma das principais plataformas de desenvolvimento de jogos nos tempos atuais. A plataforma tem seu funcionamento em tempo real, potencializada por ferramentas e serviços integrados na plataforma, oferecendo possibilidades incríveis para desenvolvedores de jogos e criadores de conteúdo (UNITY, 2020b).

A Unity3D abstrai do desenvolvedor de jogos a necessidade de utilizar diretamente DirectX ou OpenGL (apesar de ainda ser possível, caso necessário), suportando a criação de Shaders complexos com a linguagem Cg da NVidia. Internamente, o subsistema de simulação física é o popular PhysX, também da NVidia. Para a execução de *scripts*, a Unity usa uma versão de alto desempenho da biblioteca Mono, uma implementação de código aberto do framework .Net da Microsoft (PASSOS et al., 2009).

Um grande ponto forte da Unity é a sua importação de arquivos. Segundo Kucera (2013), ela consegue importar arquivos de vários formatos, tanto 2D como 3D, aceitando os arquivos de diversos programas de modelagem como o Blender, Autodesk 3DS *Max*, Autodesk Maya, Cheetah3D, Maxon CINEMA 4D, Luxology Modo 3D, entre outros.

O motor gráfico da Unity 3D usa Direct3D, OpenGL, OpenGL ES e APIs proprietárias. Há suporte para mapeamento de relevo, mapeamento de reflexão, mapeamento de parallax, ambient occlusion, sombras dinâmicas usando mapas de sombra, render-to-texture e efeitos de pós-processamento (KUCERA, 2013).

Os games do Unity são baseados em cenas, os *Game Objects* são praticamente todos os objetos dentro da cena. São todos os elementos posicionados dentro da cena através de um sistema de coordenadas, seja em 2 ou 3 dimensões. Câmeras, modelos, luzes, sistemas de partículas: todos são *Game Objects*. Esses elementos são a unidade fundamental dentro de qualquer cena de *game* dentro do Unity (GASPAROTTO, 2014).

4.2 Vuforia

Vuforia Engine é a plataforma mais amplamente usada para o desenvolvimento de RA, com suporte para os principais telefones, tablets e óculos. Os desenvolvedores podem adicionar facilmente funcionalidades avançadas de visão computacional aos aplicativos Android, iOS e UWP, para criar experiências de RA que interagem de forma realista com objetos e o ambiente (VUFORIA, 2020).

4.2.1 Recursos do Vuforia Engine

Os recursos de reconhecimento e rastreamento do Vuforia Engine podem ser usados em uma variedade de imagens e objetos. A seguir está uma lista dos recursos fornecidos pelo Vuforia segundo a sua documentação:

- **Alvos de modelo**: permitem reconhecer objetos por forma usando modelos 3D pré-existentes. Permite colocar conteúdos de RA em uma ampla variedade de itens, como equipamentos industriais, veículos, brinquedos e eletrodomésticos;
- **Alvos de área**: aumenta ambientes reais digitalizados usando um scanner 3D disponível comercialmente. Permite criar conteúdos persistente alinhado com precisão em uma ampla variedade de lugares comerciais, públicos ou divertidos para enriquecer os espaços com experiências aprimorada;
- **Alvos de imagem**: anexe conteúdo a imagens planas, como mídia de impressão e embalagens de produtos;
- **Alvos de objeto**: são criados verificando um objeto. Eles são uma boa opção para brinquedos e outros produtos com detalhes de superfície ricos e um formato consistente;
- **Múltiplos alvos**: são criados usando mais de um alvo de imagem e podem ser organizados em formas geométricas regulares (por exemplo, caixas) ou em qualquer arranjo arbitrário de superfícies planas;
- **Alvos de cilindro**: reconhece imagens embrulhadas em objetos de formato aproximadamente cilíndrico (por exemplo, garrafas de bebidas, xícaras de café, latas de refrigerante);
- **VuMarks**: são marcadores personalizados que podem codificar uma variedade de formatos de dados. Eles suportam identificação e rastreamento exclusivos para aplicativos de RA;
- **Câmera externa**: acesse os dados de vídeo de uma câmera fora da câmera de um telefone ou tablet ao criar experiências de RA;
- **Plano de chão**: permite colocar conteúdo em superfícies horizontais no ambiente, como mesas e pisos;

O Vuforia Engine também pode reconhecer e rastrear objetos 3D. O reconhecimento de objeto permite que os *Object Targets* (Alvos de Objeto) sejam criados verificando objetos físicos.

O Vuforia Fusion é um conjunto de tecnologias desenvolvidas para resolver o problema de fragmentação das tecnologias que permitem RA, incluindo câmeras, sensores, chipsets e estruturas de software como ARKit e ARCore. Fundindo os recursos do dispositivo subjacente com os recursos do Vuforia Engine, permite que os desenvolvedores consigam uma experiência com RA indiferente do modelo do dispositivo, seja ele, Android ou iOS. O Vuforia Fusion é usado por vários recursos do Vuforia Engine, como:

- **Rastreador de dispositivo**: oferecendo postura de dispositivo de seis graus de liberdade;
- **Plano de chão**: permitindo que o conteúdo virtual seja colocado em planos horizontais

no ambiente;

- **Rastreamento estendido**: permitindo rastreamento estendido para todos os tipos de alvo Vuforia;

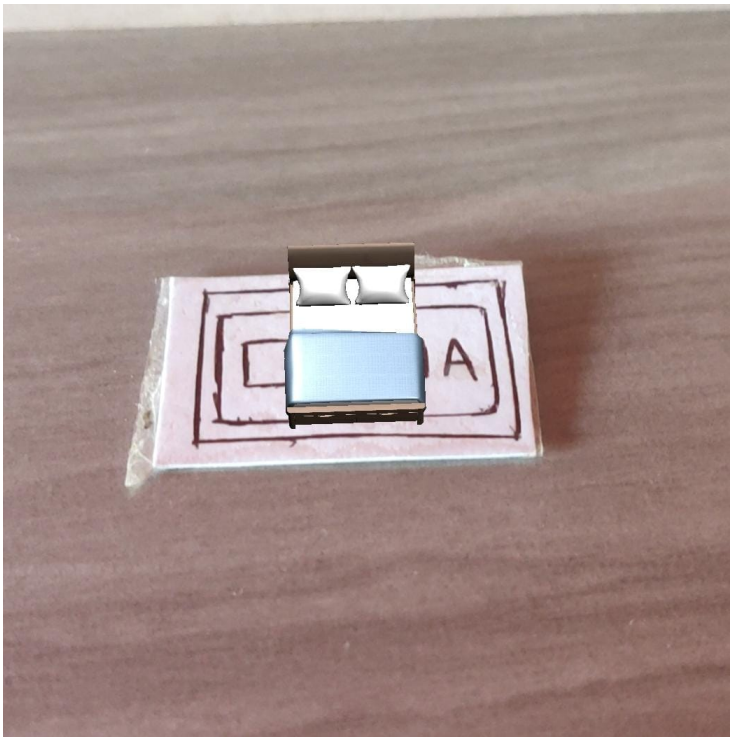
Os Serviços da Web Vuforia API permite que você gerencie esses grandes bancos de dados de imagens na nuvem de forma eficiente e permite que você automatize seus fluxos de trabalho por integração direta em seus sistemas de gerenciamento de conteúdo.

O Vuforia Engine *Driver Framework* permite que os desenvolvedores forneçam e consumam dados de sistemas externos por meio do Vuforia Engine. Esta estrutura fornece acesso aos recursos da câmera externa. A câmera externa define todas as especificações exigidas para que o Vuforia Engine acesse fontes externas de imagem.

4.2.2 *Image Targets*

Image Targets (Alvos de imagem) representam imagens que o Vuforia Engine pode detectar e rastrear. O mecanismo detecta e rastreia a imagem, comparando os recursos naturais extraídos da imagem da câmera com um banco de dados de recursos de destino conhecido (VUFORIA, 2020). Quando a câmera do dispositivo reconhece esses *targets* no mundo real, isso aciona a exibição de conteúdo virtual sobre a posição do marcador na visualização da câmera. Na Figura 11 está um exemplo de um marcador e a projeção do objeto 3D sobre o mesmo.

Figura 11 – Projeção de uma cama utilizando *Image Target*



Fonte: AUTOR

4.3 Linguagem de programação C#

Segundo a documentação da Microsoft (2020), o C# é uma linguagem de programação orientação à objetos com sua sintaxe parecida com as linguagens C/C++, oferece suporte aos conceitos de encapsulamento, herança e polimorfismo. Na figura Figura 12 está um exemplo da sintaxe do C#.

Figura 12 – Sintaxe de código em C#

```
1   public class Teste
2   {
3       public static void Main()
4       {
5           System.Console.WriteLine("Realidade Aumentada");
6       }
7   }
```

Fonte: AUTOR

> A sintaxe C# é altamente expressiva, mas também é simples e fácil de aprender. A sintaxe de chave do C# será reconhecível instantaneamente para qualquer pessoa que esteja familiarizada com C, C++, Java ou JavaScript. Os desenvolvedores que conhecem qualquer uma dessas linguagens geralmente podem trabalhar de maneira produtiva em C# em um curto período de tempo. O C# fornece recursos avançados, como tipos anuláveis, delegados, expressões lambda, correspondência de padrões e acesso de memória direta seguro. O C# oferece suporte a tipos e métodos genéricos, que fornecem aumento na segurança e no desempenho do tipo. O C# fornece iteradores, que habilitam implementadores de classes de coleção para definir comportamentos personalizados para o código do cliente. As expressões de consulta de Language-Integrated (LINQ) fazem com que a consulta fortemente tipada seja uma construção de linguagem de primeira classe (MICROSOFT, 2020).

O C# é uma linguagem de programação orientada a objetos e orientada a componentes. O C# fornece construções de linguagem para dar suporte direto a esses conceitos, tornando o C# uma linguagem natural para criar e usar componentes de software (MICROSOFT, 2020).

4.4 3DS *Max*

O 3ds *Max* é um programa de modelagem tridimensional que permite renderização de imagens e animações através de um conjunto de ferramentas flexíveis e repleto de recursos para a criação de seus projetos.

Com o 3ds *Max*, é possível criar lugares e personagens 3D, objetos e sujeitos de todos os tipos. É possível organizá-los em configurações e ambientes para criar as cenas do seu filme, jogo ou visualização. É possível animar os personagens, colocá-los em movimento, fazê-los falar, cantar e dançar, ou chutar e lutar. E, em seguida, é possível fazer filmes de toda a ação virtual (AUTODESK, 2015).

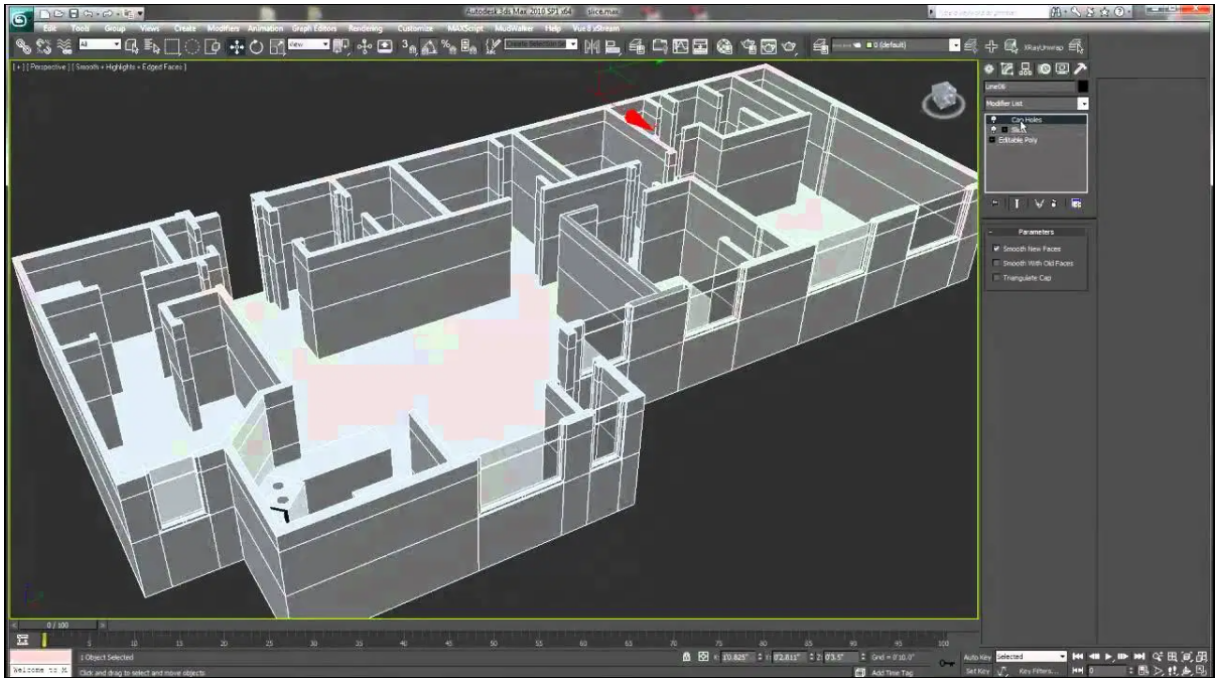
4.4.1 Modelagem

Segundo Almeida (2007), modelagem tridimensional é uma área da computação gráfica que tem como finalidade criar objetos em três dimensões, fazer renderizações e criar animações.

> Uma montagem de personagens é um tipo especial de grupo para objetos específicos de uma configuração de personagem: a malha, os ossos, a cinemática inversa, as cadeias, os objetos auxiliares, os controladores e outros objetos do personagem usados para a animação de personagens. Depois que os objetos são agrupados (montados), é possível executar várias funções no grupo como um todo, como salvar e carregar uma animação para todo o conjunto de ossos/malhas (AUTODESK, 2020).

É possível modelar objetos na sua cena criando objetos padrões, como geometria 3D e formas 2D, e aplicar modificadores a esses objetos. O 3ds *Max* inclui uma ampla gama de objetos padrões e modificadores. A Figura 13 mostra a interface do 3ds *Max* e a modelagem de uma planta de uma casa.

Figura 13 – Modelagem da planta de uma casa



Fonte: (AZEVEDO, 2003)

4.5 SDK Android

Segundo a documentação do Android Studio (ANDROID, 2020), ele é o *Integrated Development Environment* (IDE) oficial para o desenvolvimento de aplicativos Android, baseado no IntelliJ IDEA. Além do poderoso editor de código e ferramentas de desenvolvedor do

IntelliJ, o Android Studio oferece ainda mais recursos que aumentam sua produtividade ao criar aplicativos Android, como:

- Um sistema de compilação baseado em Gradle flexível;
- Um emulador rápido e rico em recursos;
- Um ambiente unificado onde você pode desenvolver para todos os dispositivos Android;
- Aplicar alterações para enviar alterações de código e recursos para seu aplicativo em execução sem reiniciá-lo;
- Modelos de código e integração com o GitHub para ajudá-lo a criar recursos de aplicativo comuns e importar códigos de amostra;
- Extensas ferramentas e estruturas de teste;
- Ferramentas Lint para detectar desempenho, usabilidade, compatibilidade de versão e outros problemas;
- Suporte a C ++ e NDK;
- Suporte integrado para Google Cloud Platform , facilitando a integração do Google Cloud Messaging e do App Engine;

Segundo (VAATI, 2020), o Android SDK é uma coleção de ferramentas de desenvolvimento de software e bibliotecas necessárias para desenvolver aplicativos Android. Cada vez que o Google lança uma nova versão do Android ou uma atualização, um SDK correspondente também é lançado, o qual os desenvolvedores devem baixar e instalar. O SDK do Android é formado pelos seguintes componentes:

- **Android SDK *Tools***: inclui um conjunto completo de ferramentas de desenvolvimento e depuração para Android e está incluído no Android Studio. As ferramentas SDK também consistem em ferramentas de teste e outros utilitários necessários para desenvolver um aplicativo;
- **SDK *Build Tools***: são as ferramentas de construção necessárias para construir componentes para construir os binários reais para seu aplicativo Android;
- **SDK Platform-Tools**: as ferramentas da plataforma Android são usadas para oferecer suporte aos recursos da plataforma Android atual e são necessárias para o desenvolvimento de aplicativos Android. Elas incluem:
  - **Android *Debug Bridge* (adb)**: esta é uma ferramenta de linha de comando útil que permite que você se comunique com um dispositivo. O comando adb permite que você execute ações do dispositivo, como instalar e depurar aplicativos. Ele também fornece acesso a um *shell* Unix que você pode usar para executar uma variedade de comandos em um dispositivo;
  - ***Fastboot***: permite atualizar um dispositivo com uma nova imagem do sistema;
  - ***Systrace***: esta ferramenta ajuda a coletar e inspecionar informações de tempo em todos os processos em execução no seu dispositivo no nível do sistema. É crucial para depurar o desempenho do aplicativo;
- **Plataforma SDK**: para cada versão do Android, há uma plataforma SDK disponível. Eles

são numerados de acordo com a versão do Android e uma versão da API. As versões mais recentes da plataforma SDK têm mais recursos para desenvolvedores, mas os dispositivos mais antigos podem não ser compatíveis com as versões mais recentes da plataforma;

- **APIs do Google**: o Google fornece uma série de APIs exclusivas Google para facilitar o desenvolvimento de seu aplicativo. Eles também oferecem uma imagem do sistema para o emulador para que você possa testar seu aplicativo usando as suas APIs;
- **Android Emulator**: é uma ferramenta de emulação de dispositivo baseada em QEMU que simula dispositivos Android em seu computador, permitindo que os desenvolvedores testem aplicativos em diferentes dispositivos e níveis de API do Android, sem a necessidade de dispositivos físicos para cada um;

5 DESENVOLVIMENTO DA APLICAÇÃO

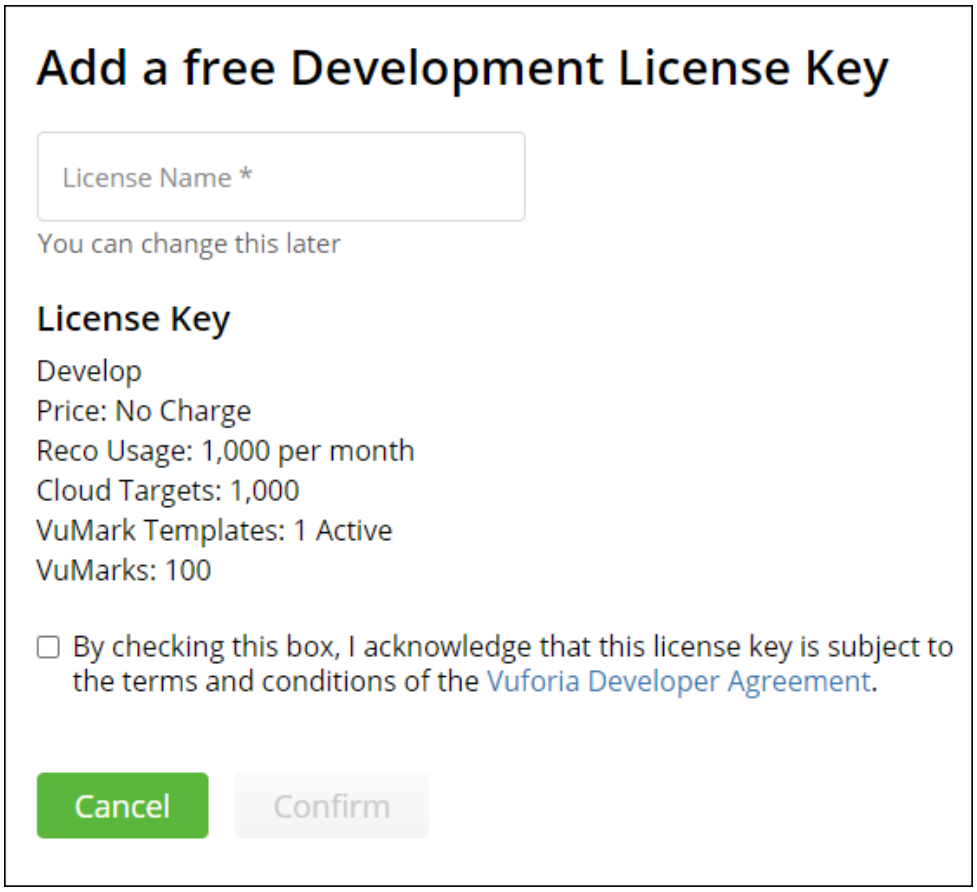
O desenvolvimento da aplicação foi dividido em três grandes etapas:

1. Criação dos *image targets* com os nomes dos objetos a serem renderizados pela aplicação;
2. Modelagem da planta baixa;
3. Desenvolvimento da cena na Unity integrado com o SDK do Vuforia, juntamente com as duas etapa anteriores resultando na aplicação final.

5.1 Configuração e Criação dos *image targets*

A criação dos *image targets* foi realizado através Vuforia. Para poder utilizar o Vuforia foi necessário criar uma conta na página do desenvolvedor e criar uma chave de licença (essa chave é adicionada no Prefab ARCamera). Para criar a chave é necessário acessar o menu *"Develop"*, submenu *"License Manager"* e clicar em *"Get Development Key"*. Na Figura 14 é possível visualizar a tela de cadastro da chave.

Figura 14 – Cadastro da chave do Vuforia

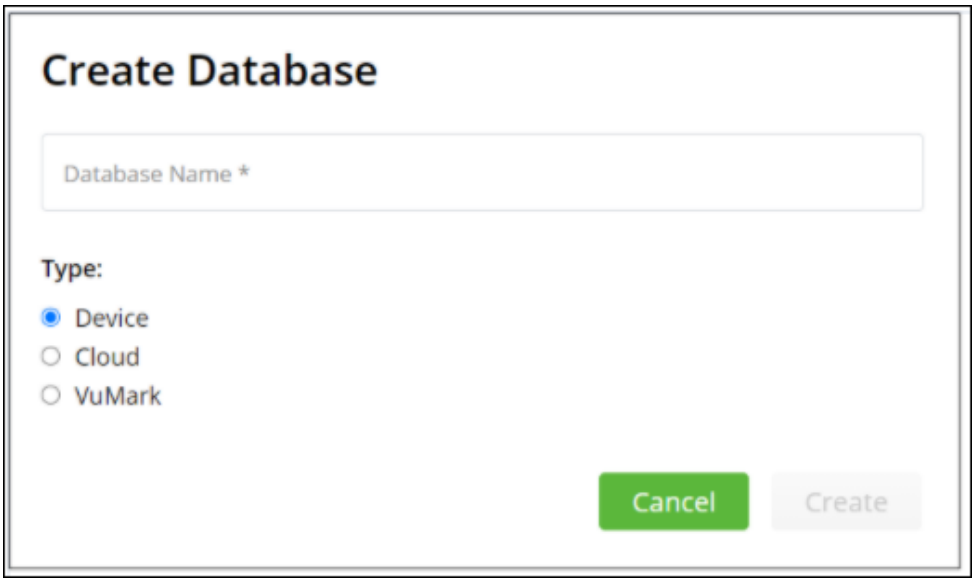


Fonte: AUTOR

Após a criação da chave foi necessário criar um banco de dados para armazenar e processar as imagens. O Vuforia suporta três tipo de banco de dados. Na Figura 15, é possível ver os tipos de bancos que podem ser criados e sua tela de cadastro. A seguir, as suas descrições:

- **Banco de dados Device**: são banco locais de destinos de imagem ou objeto armazenados no dispositivo do usuário;
- **Bancos do tipo VuMark**: são banco de dados locais de VuMarks possuindo seu armazenamento localizado nos dispositivos de usuários;
- **Banco de dados em nuvem**: são bancos com destinos para armazenamento de imagens em um ambiente online para serem consultados pela Internet;

Figura 15 – Tipos de banco do Vuforia



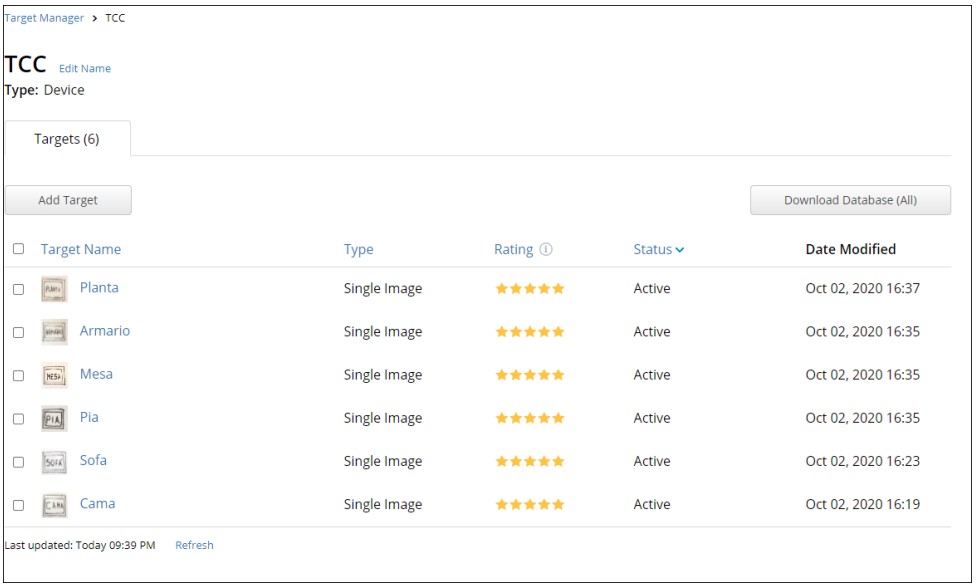
Fonte: AUTOR

5.1.1 Inserindo *targets* no banco de dados

Com o banco de dados criado torna-se possível a inserção de registros. O banco do Vuforia é muito simples de ser manuseado pois é totalmente visual. Após a criação do banco, direciona-se para a etapa de inserção e listagem das *targets*.

A listagem dos dados é composta por *Target Name* (Nome do Alvo); *Type* (Tipo), que variam entre *Single Plane, Cuboid, Cylinder* e 3D *Object*; *Rating* (Classificação), determina a qualidade e velocidade que o banco de dados irá detectar o alvo; *Status* (Estado), determina se a target está ativa ou não; E a última coluna é a data de modificação como mostra a Figura 16.

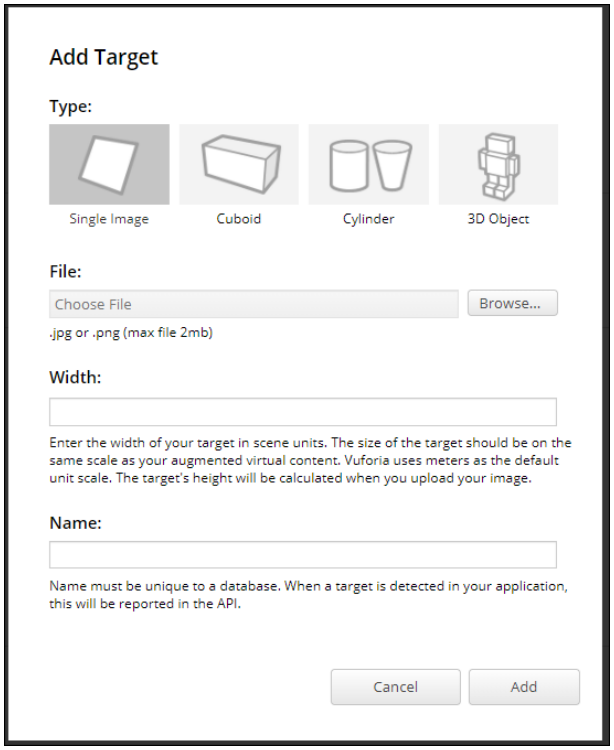
Figura 16 – Banco de dados Vuforia



Fonte: AUTOR

Para inserir as *targets* basta clicar no botão *"Add Target"*, selecionar o tipo do alvo, que neste caso foi *Single Image*, escolher a imagem desejada, seu tamanho e um nome conforme a Figura 17. As *image targets* foram confecções artesanais demonstradas na Figura 18.

Figura 17 – Cadastro de *image target*



Fonte: AUTOR

Figura 18 – Marcadores artesãos

Fonte: AUTOR

Quando feito o cadastro da *target*, é iniciado o processo de classificação da mesma, onde uma nota de zero a cinco estrelas é atribuída. Caso a nota seja zero, o alvo não será detectado pelo Vuforia na aplicação, quanto maior a classificação mais fácil e rápida é a detecção do alvo durante a execução. É possível alterar a imagem do *target* e alterar seu nome, conforme mostra a Figura 19:

Figura 19 – Edição de um *image target*

Fonte: AUTOR

Na tela de edição ainda é possível visualizar as características (*Features*) da classificação da imagem. Na Figura 20 pode-se visualizar marcações amarelas que são os pontos de reconhecimento do Vuforia para identificação do alvo. Quanto mais pontos, melhor é sua classificação, tornando mais fácil sua detecção. Para acessar essa classificação basta clicar na opção *"Show Features"*, presente na Figura 19.

Figura 20 – Features de um *image target*

Fonte: AUTOR

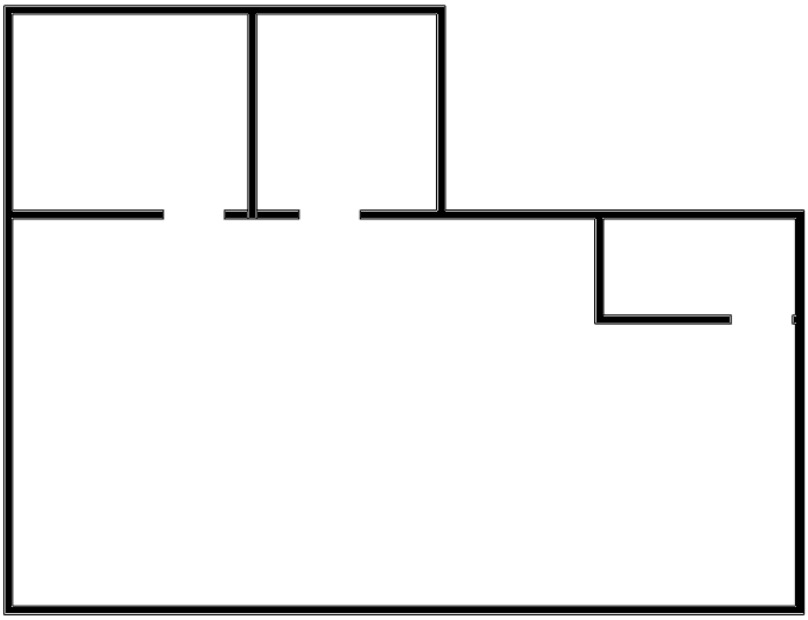
5.2 Modelagem da planta baixa

A modelagem da planta baixa consiste em duas etapas. A primeira etapa é feita pela Unity lendo uma imagem de uma planta baixa criando cubos para cada pixel preto indicando que são paredes. A segunda etapa é a junção de todos os cubos gerados em um único objeto através do 3D *Max*.

5.2.1 Criação da planta baixa

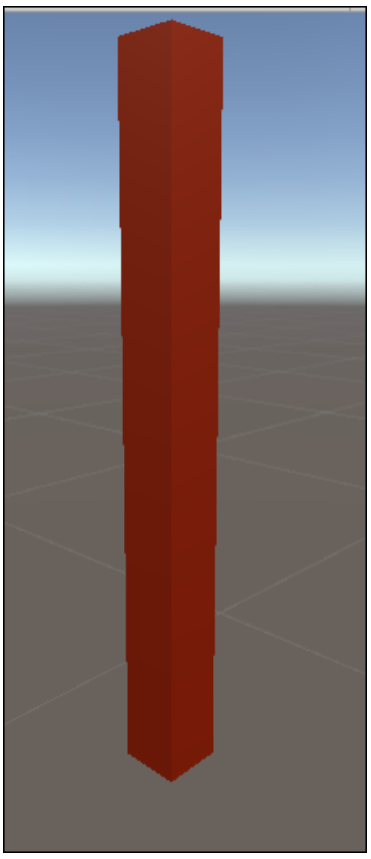
Para a criação da planta baixa foi importada a Figura 21 para a Unity para servir de modelo de planta baixa. A Figura 22 foi utilizada como bloco para desenhar as paredes da planta.

Figura 21 – Imagem para criação da planta baixa



Fonte: AUTOR

Figura 22 – Cubo para desenhar as paredes



Fonte: AUTOR

Para que fosse possível desenhar o cubo mostrado na Figura 22 em cada pixel preto da imagem, foi necessária a implementação de um *script* capaz de detectar esses pixels e renderizar o cubo. O *script* foi nomeado de *"GenerateMap"* e pode ser visto nas Figura 23 e Figura 24. O resultado da planta baixa na Figura 25

Figura 23 – *Script* para renderização de cubos parte 1

```
using System.Collections;
using System.Collections.Generic;
using System;
using UnityEngine;
using Random=UnityEngine.Random;

public class GenerateMap : MonoBehaviour
{
    [SerializeField]
    private Texture2D[] images;
    private Texture2D image;

    [SerializeField]
    private GameObject wallObject;

    [SerializeField]
    private GameObject imageRender;

    void Start()
    {
        float increment = 0.0015f;

        image = images[Random.Range(0, images.Length)];
        Color[] pix = image.GetPixels();

        int worldX = image.width;
        int worldZ = image.height;

        Vector3[] spawnPositions = new Vector3[pix.Length];
        Vector3 startingSpawnPositions = new Vector3(-Mathf.Round(worldX / 2 * increment), 0, -Mathf.Round(worldZ / 2 * increment) );
        Vector3 currentSpawnPos = startingSpawnPositions;

```

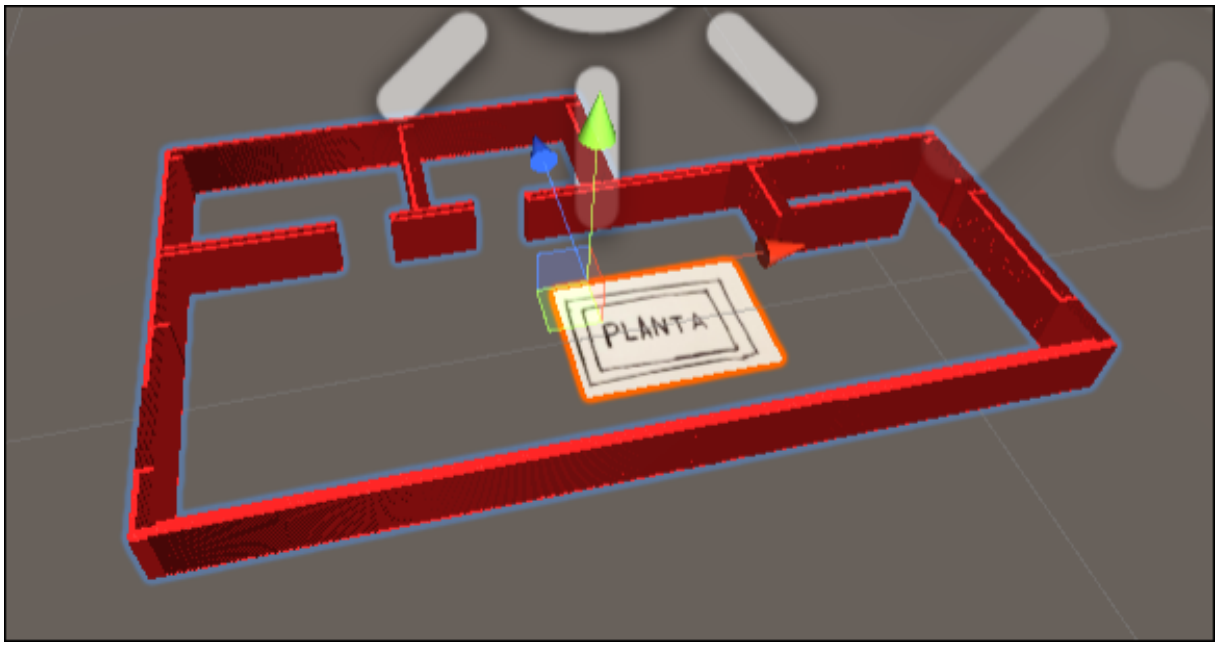
Fonte: AUTOR

Figura 24 – *Script* para renderização de cubos parte 2

```
        int counter = 0;

        for (int z = 0; z < worldZ; z++) {
            for (int x = 0; x < worldX; x++) {
                spawnPositions[counter] = currentSpawnPos;
                counter++;
                currentSpawnPos.x+=increment;
            }
            currentSpawnPos.x = startingSpawnPositions.x;
            currentSpawnPos.z+=increment;
        }

        foreach (Vector3 pos in spawnPositions)
        {
            Color c = pix[counter];

            if (! c.Equals(Color.white)) {
                GameObject cube = Instantiate(wallObject, pos, Quaternion.identity);
                cube.transform.parent = imageRender.transform;
            }
        }
    }
}
```

Fonte: AUTOR

Figura 25 – Planta baixa gerada através do *script GenerateMap*



Fonte: AUTOR

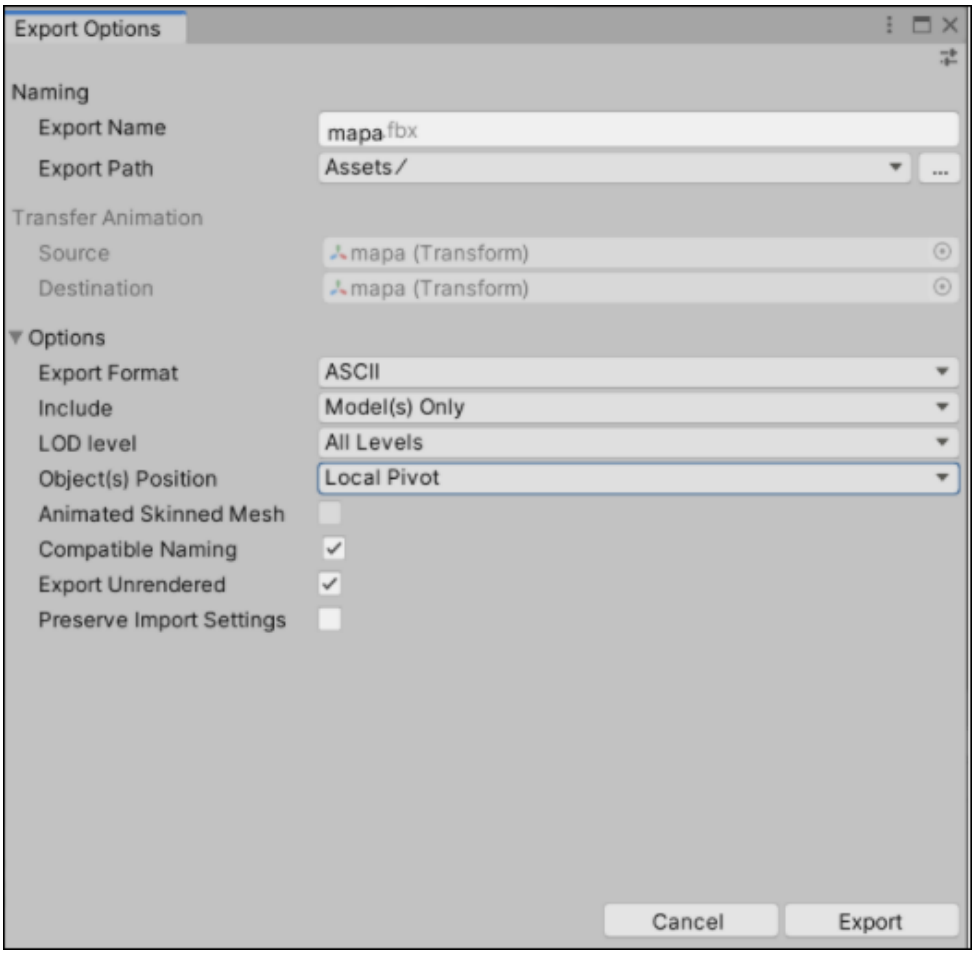
5.2.2 Exportação na Unity e modelagem no 3DS Max

Após a geração da planta baixa foi necessário exportá-la para o 3DS Max e mesclar os cubos devido a sua grande quantidade. Para importar no 3DS *Max* foi necessário exportar o modelo da planta baixa no formato FBX. Segundo a documentação da Unity é preciso adicionar o pacote "FBX *Explorer*", seguido os passos descritos a seguir:

1. Abra a janela do Gerenciador de pacotes e selecione *"All packages"* no menu suspenso do escopo do pacote;
2. Selecione o pacote que deseja instalar na lista de pacotes. As informações do pacote aparecem na exibição de detalhes. Observação: por padrão, o Gerenciador de pacotes não exibe pacotes de visualização. Se você não vir o pacote na lista de pacotes, pode ser um pacote de visualização. Certifique-se de que "Mostrar pacotes de visualização"esteja habilitado no menu suspenso "Avançado"na janela "Gerenciador de pacotes". Para obter mais informações, consulte Incluindo pacotes de visualização;
3. Selecione a versão para instalar;
4. Clique no botão Instalar (UNITY, 2020a).

Para exportar para o formato FBX, basta selecionar o objeto, clicar com o direito e ir na nova opção *"Export to FBX..."*. Em seguida, é exibida uma tela com os detalhes da exportação, onde pode-se escolher se irá ser exportado todos os níveis do objeto, suas animações, escolher seu formato, entre outros. Na Figura 26 podemos ver as opções de exportação.

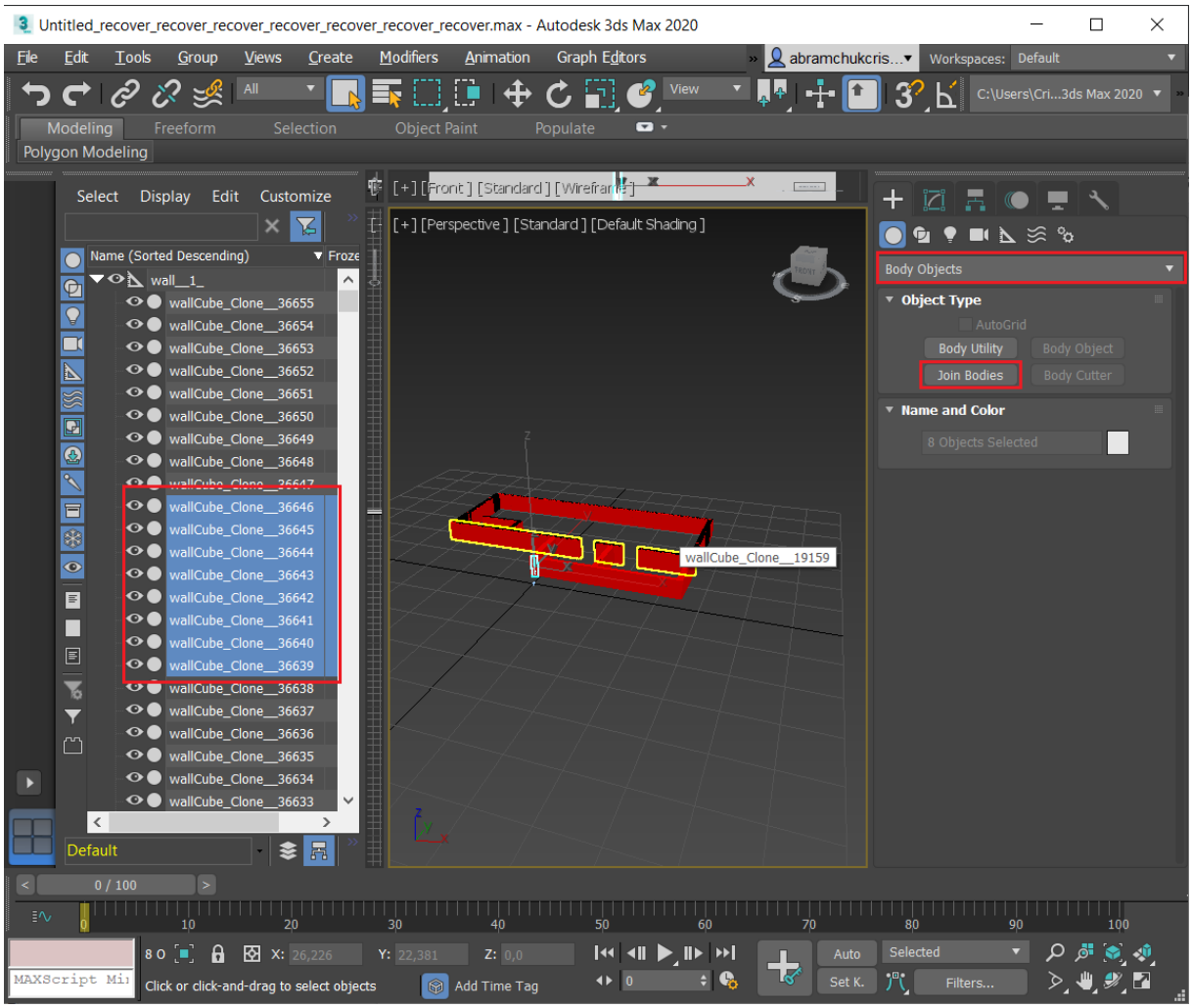
Figura 26 – Opções de exportação para FBX



Fonte: AUTOR

Depois de exportado e importado o objeto no 3ds *Max*, foi utilizada a geometria de *"Body Objects"* e a opção de *"Join Bodies"* para juntar os corpos dos cubos. Para utilizar essa opção é preciso selecionar os objetos na cena e escolher a opção de *"Join Bodies"*, como demonstrado e destacado na Figura 27.

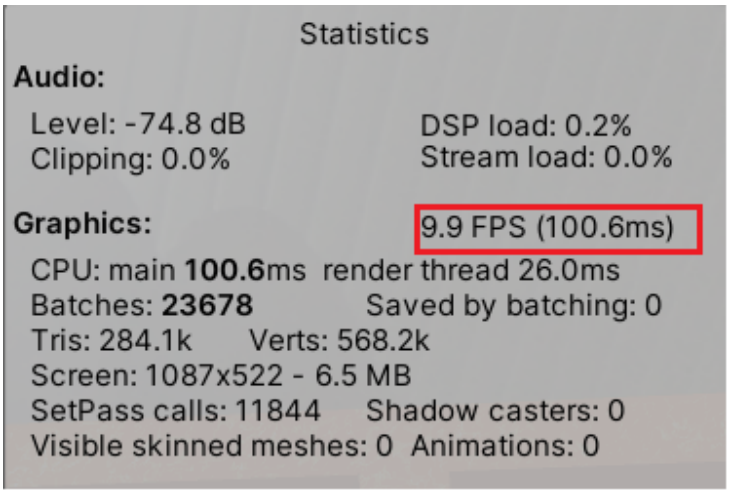
Figura 27 – Mesclando cubos no 3ds *Max*



Fonte: AUTOR

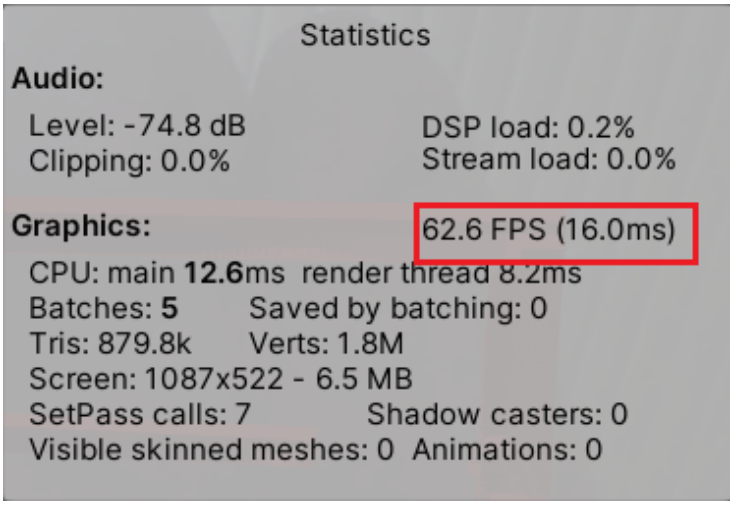
Depois de terminada a junção dos cubos, o objeto foi exportado e importado para Unity gerando a mesma imagem do objeto apresentado na Figura 27, porém agora como um objeto único ao invés de diversos cubos, o que resultou num grande ganho de performance na execução da aplicação pela diminuição da complexidade da geometria. Na Figura 28, podemos ver a quantidade de FPS (Quadros por segundo) antes da união dos cubos e na Figura 29 depois da união. Os FPS foram calculados pela Unity renderizando somente a planta baixa.

Figura 28 – FPS antes da união dos cubos



Fonte: AUTOR

Figura 29 – FPS depois da união dos cubos

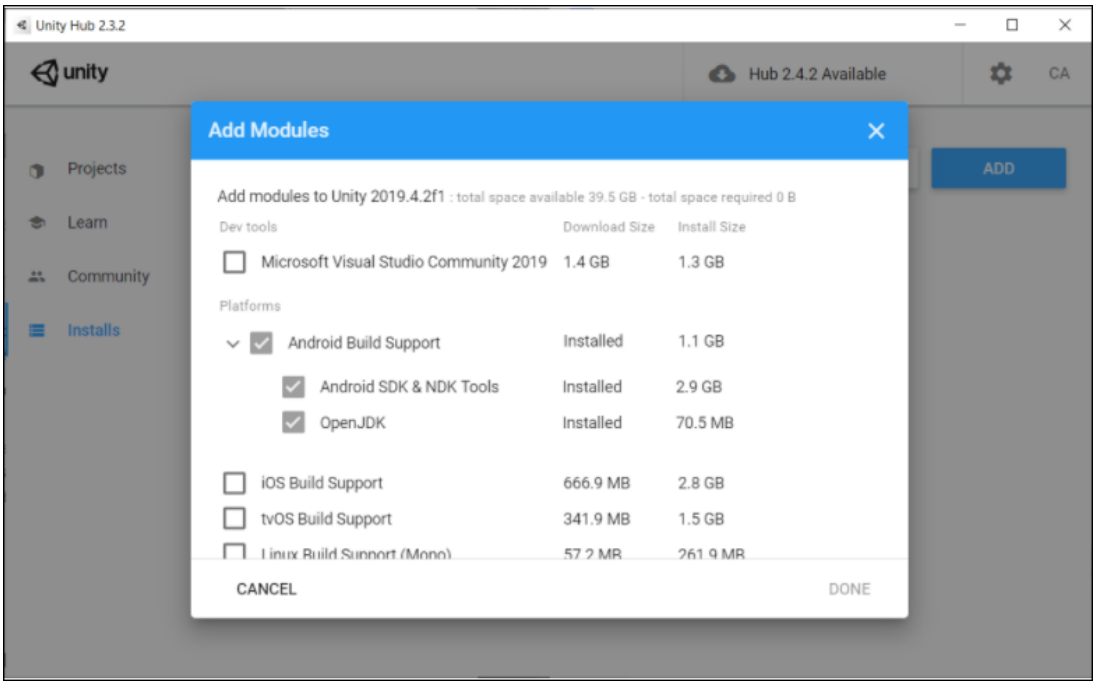


Fonte: AUTOR

5.3 Desenvolvimento da aplicação na Unity

Para o desenvolvimento da aplicação foi utilizada a versão 2019.2.4f1 da Unity. A criação e configuração do projeto Unity foi relativamente simples. Através do Unity Hub, foi possível adicionar módulos como o suporte para *build* em Android com uma simples marcação de *checkbox*. Na Figura 30 podemos ver o processo de adição de módulos no menu *"Installs"* do Unity Hub.

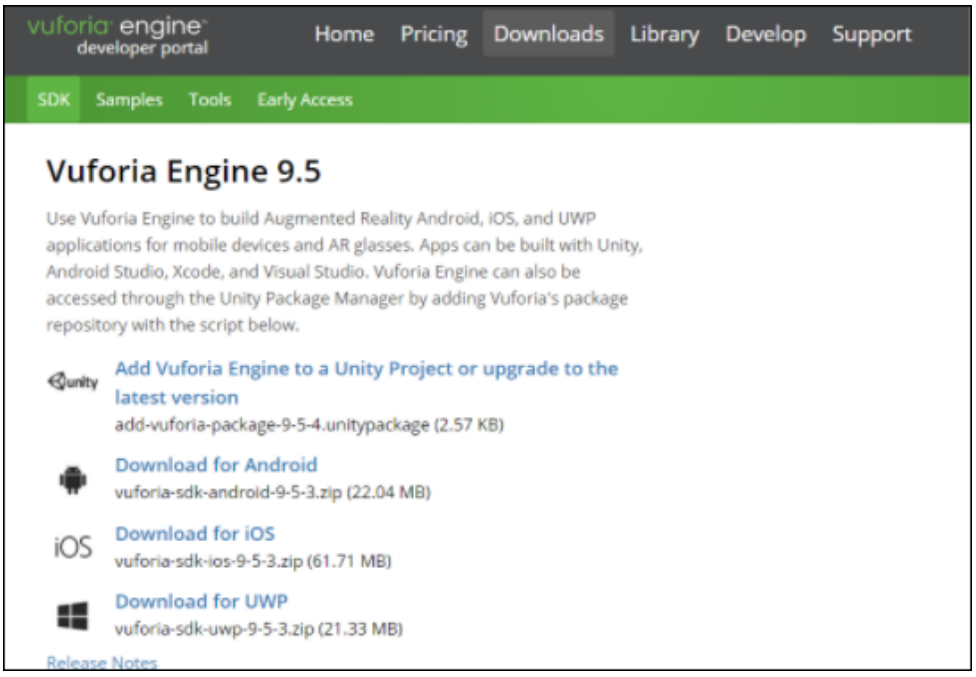
Figura 30 – Criando o projeto pelo Unity Hub



Fonte: AUTOR

Após a criação do projeto é importante adicionar o módulo do Vuforia na Unity. Para isso, é necessário fazer o *download* no site do Vuforia para desenvolvedores e, na aba *"Downloads"*, baixar a opção *"Add Vuforia Engine to a unity Project"* conforme demonstrado na Figura 31.

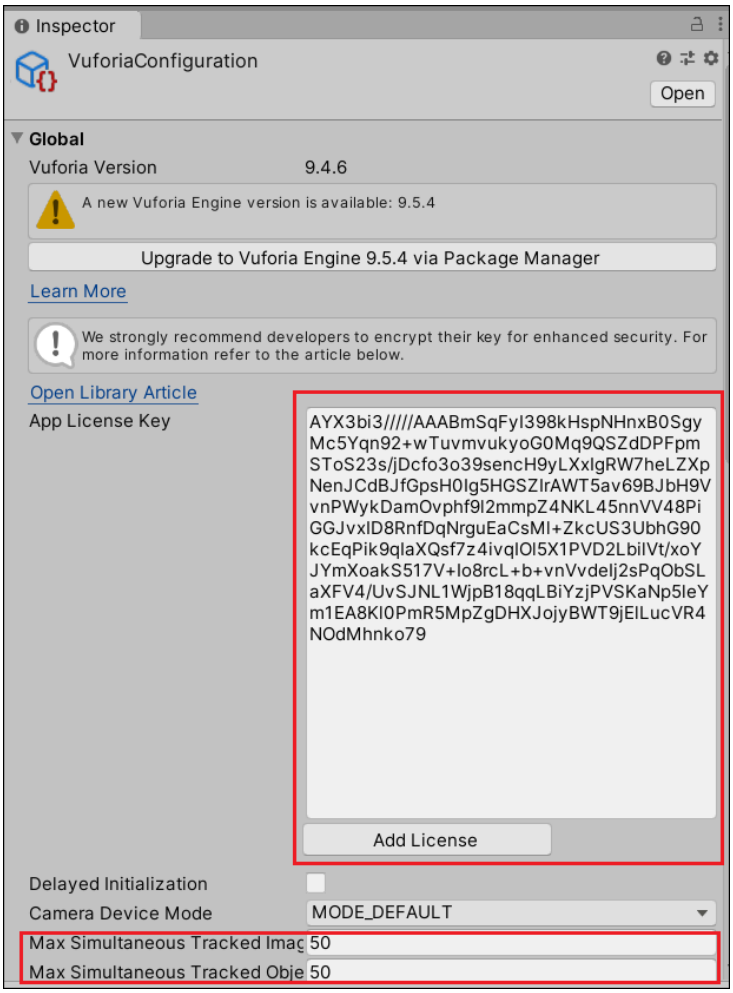
Figura 31 – Download do Vuforia para a Unity



Fonte: AUTOR

Terminado o *download*, basta executar o arquivo com o projeto aberto que aparecerá a opção “Vuforia *Engine*” no menu *“GameObject”*. Para utilizar o Vuforia é necessário adicionar uma ARCamera ao projeto e acessar a opção *“Open* Vuforia *Engine configuration”* para adição da chave de desenvolvedor e aumentar o número máximo de *targets* renderizados pelo Vuforia. Figura 32.

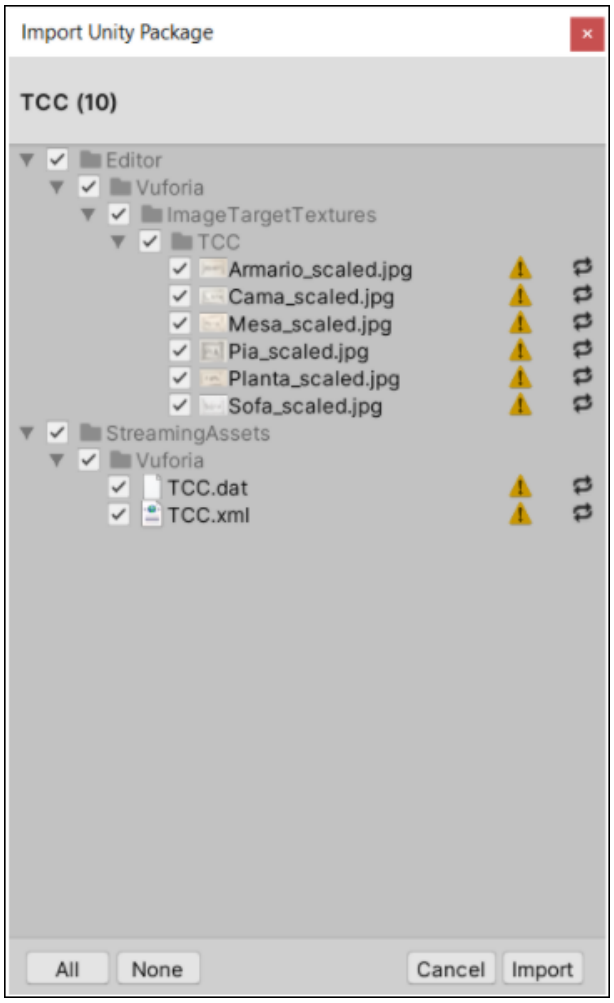
Figura 32 – Configuração do Vuforia



Fonte: AUTOR

Após a configuração do Vuforia é necessário fazer o *download* do banco de dados. Ao realizar o *download* deve-se executar o arquivo e selecionar quais *targets* serão importadas conforme mostra a Figura 33.

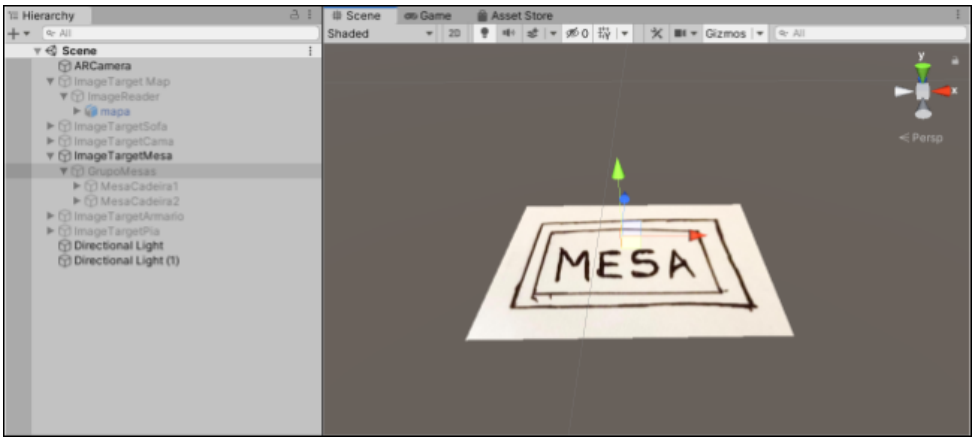
Figura 33 – Importação do banco de dados criado no Vuforia



Fonte: AUTOR

Efetuada a importação do banco de dados, é realizada a criação das *image targets*, para detecção da ARCamera, como mostra a Figura 34.

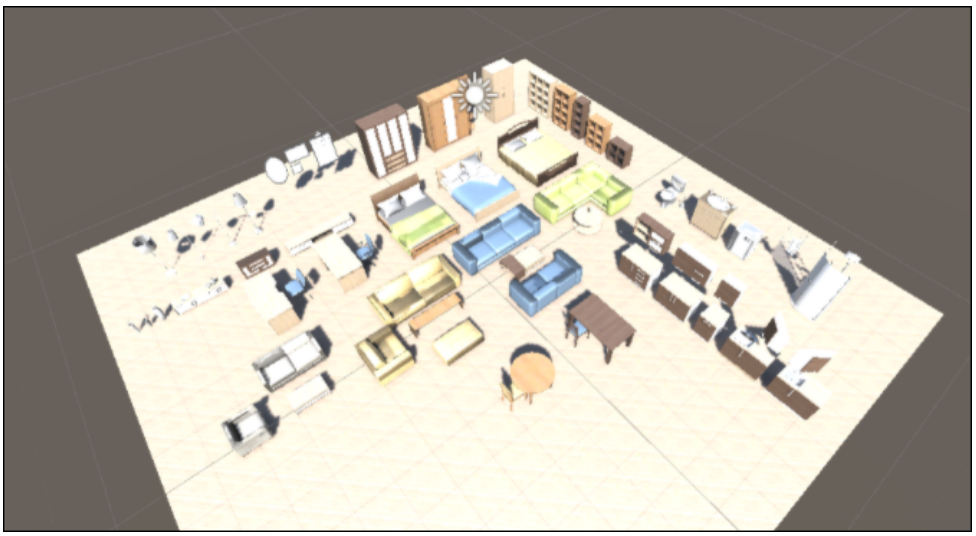
Figura 34 – Criação da *image target* da Mesa



Fonte: AUTOR

Os objetos renderizados na aplicação foram todos importados da Asset Store do pacote *"Big Furniture Pack"*. No pacote foram encontrados camas, mesas, armários, pias, entre outros. Figura 35.

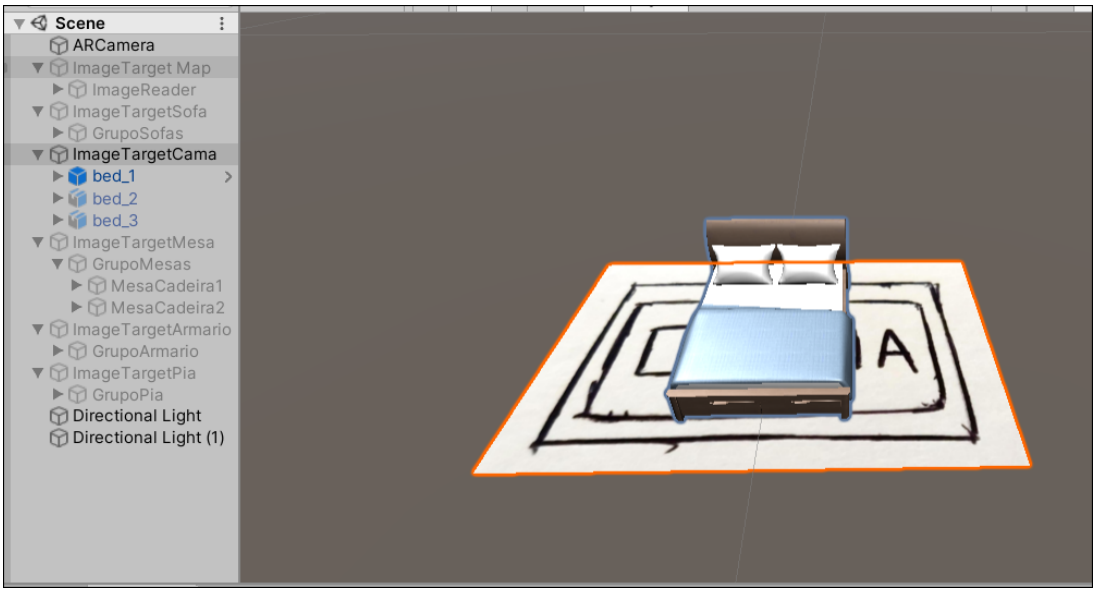
Figura 35 – Objetos do pacote *Big Furniture Pack*



Fonte: AUTOR

Foram selecionados alguns modelos do pacote *Big Furniture Pack* e adicionados à cena da aplicação. No caso da *image target* da cama foram três modelos, dos três modelos apenas um foi deixado ativado para que não houvesse conflito na hora da renderização dos mesmos. Figura 36.

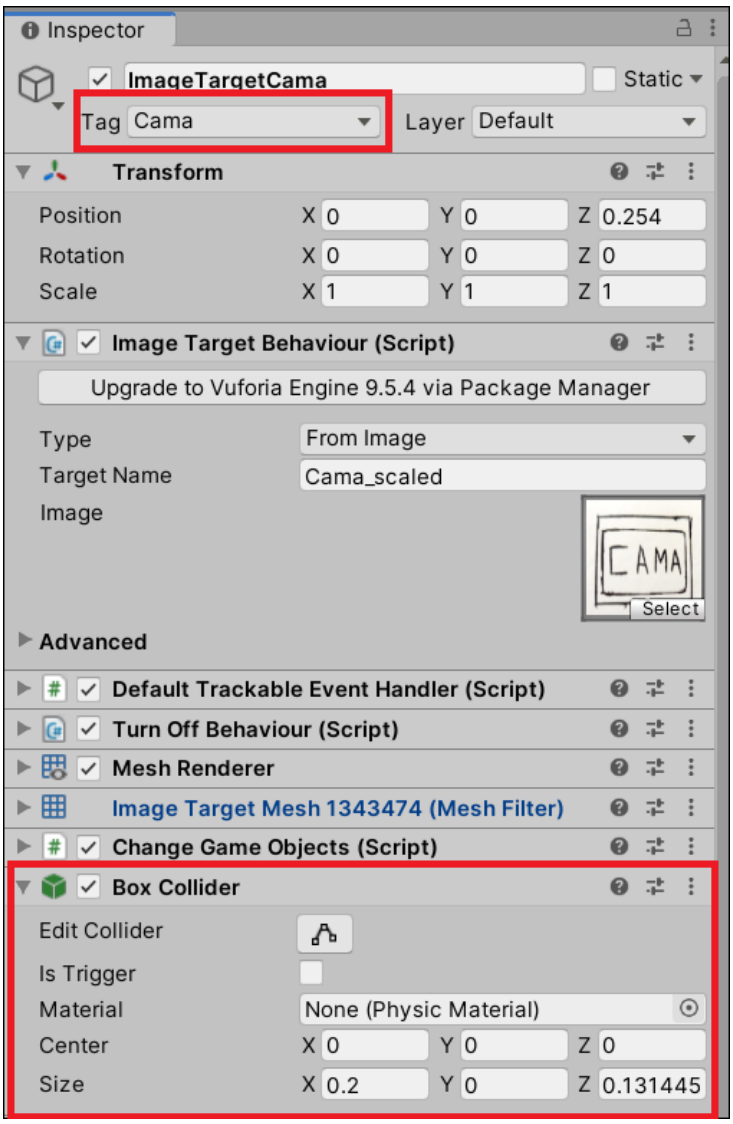
Figura 36 – Adicionando o modelo na *image target*



Fonte: AUTOR

Para que se tornasse possível a troca de objetos através do toque nos *image targets*, foi necessário adicionar um *Box Collider* e uma *tag* diferente em cada *image target* para que fosse possível identificar em qual *target* estaria ocorrendo o toque. Figura 37.

Figura 37 – Adição de *tag's e boxes colliders*

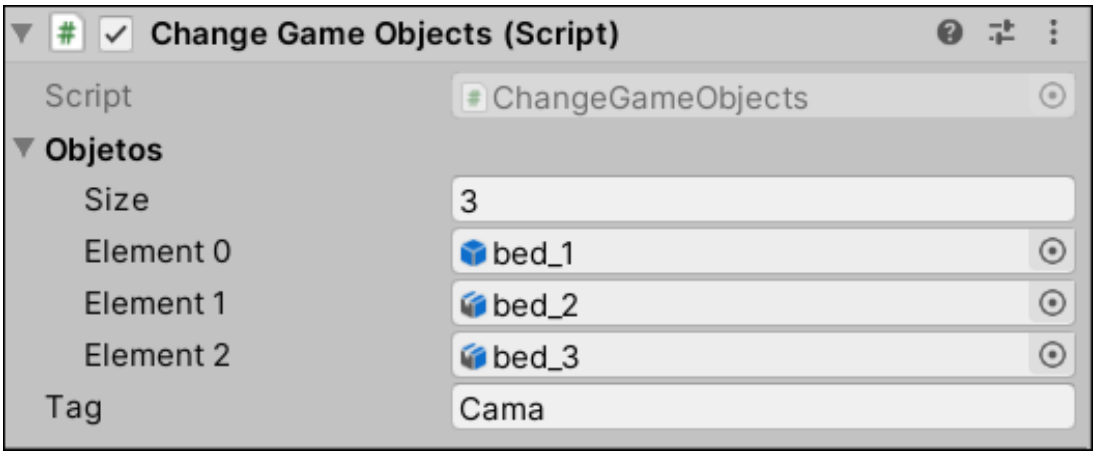


Fonte: AUTOR

Logo, foi necessário criar um *script* onde foi possível detectar qual alvo estava sendo tocado através do *smartphone* e trocar o objeto que se encontrava ativado, pelo próximo da lista que estava desativado.

O *script* foi adicionado em todos os alvos e dois campos foram parametrizados. O primeiro campo consiste em um *array* de *GameObjects*, denominado como "objetos", contendo os modelos de 3D das camas caso o alvo seja o da cama. O segundo, uma *string* contendo o nome da *tag* do alvo. Nas Figura 38 e Figura 39 pode-se verificar a parametrização e o *script* utilizado, respectivamente.

Figura 38 – Parametrização do *script* de trocas de *GameObjects*



Fonte: AUTOR

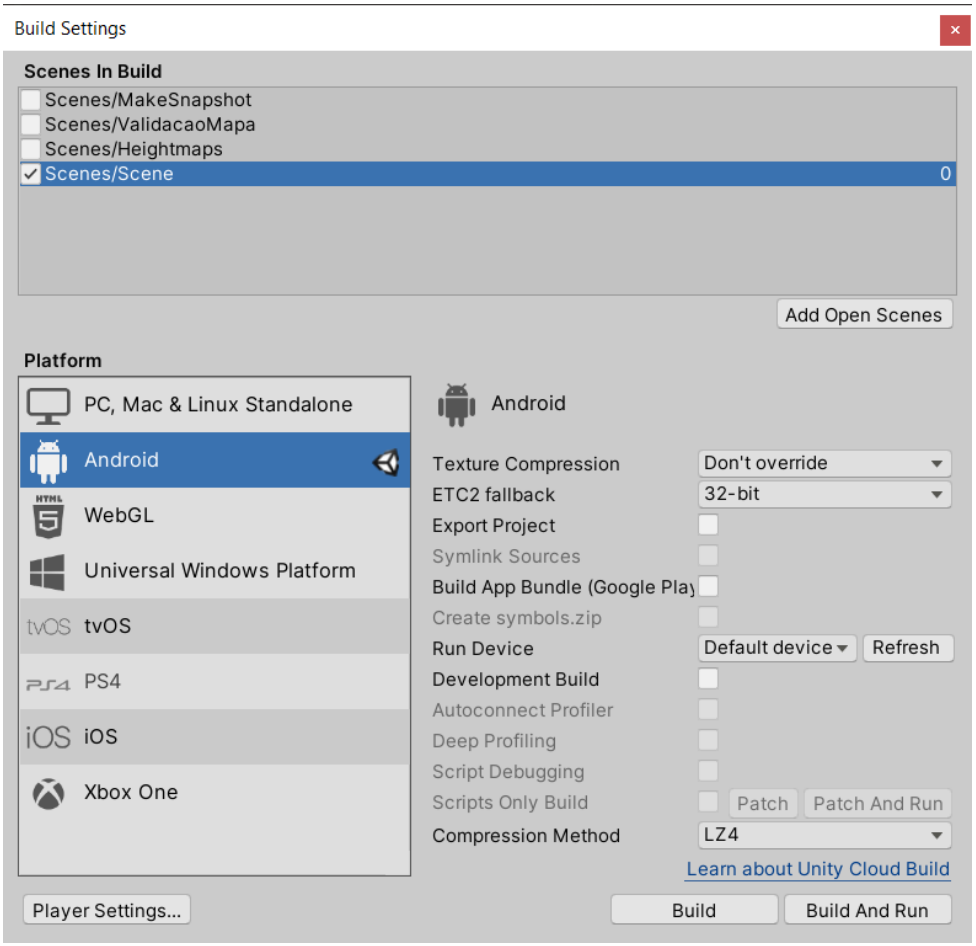
Figura 39 – *Script* para troca de *GameObjects* das *image targets*

```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;

public class ChangeGameObjects : MonoBehaviour
{
    [SerializeField]
    public GameObject[] objetos;
    public string tag;

    private GameObject objeto;

    void Update()
    {
        if (Input.touchCount > 0 && Input.touches[0].phase == TouchPhase.Began) {
            Ray ray = Camera.main.ScreenPointToRay(Input.GetTouch(0).position);
            RaycastHit Hit;

            if (Physics.Raycast(ray, out Hit)) {

                if (Hit.transform.tag == tag) {
                    int novoObjeto = 1;
                    int count = 0;
                    foreach (GameObject filho in objetos) {
                        if (filho.transform.gameObject.activeInHierarchy) {
                            novoObjeto = (count == objetos.Length -1) ? 0 : count + 1;
                        }
                        filho.transform.gameObject.SetActive(false);
                        count++;
                    }
                    objeto = objetos[novoObjeto];
                    objeto.transform.gameObject.SetActive(true);
                }
            }
        }
    }
}
```

Fonte: AUTOR

Com todas as *image targets* definidas e aplicadas na cena, dá-se o passo para o processo de compilação da aplicação. A Figura 40 demonstra as plataformas em que a Unity suporta para compilação de seus projetos e as cenas que serão carregadas na compilação. Para este projeto a plataforma escolhida foi Android.

Figura 40 – Configuração de *build* da aplicação

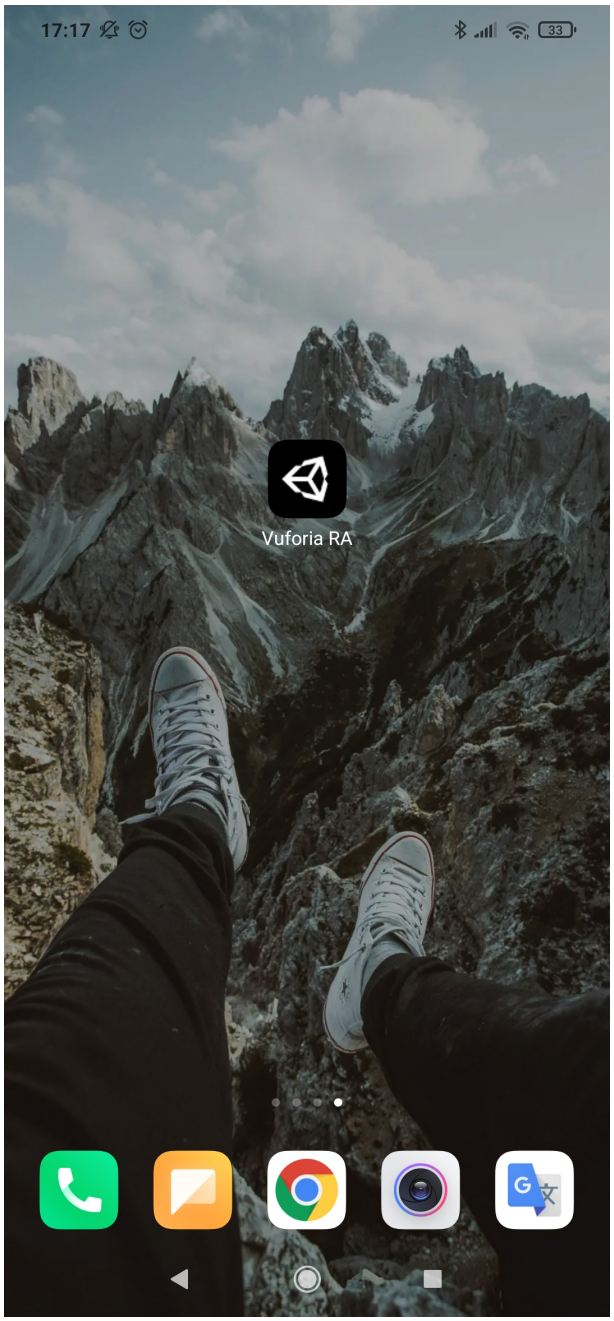


Fonte: AUTOR

5.4 Guia de Usabilidade do Aplicativo

Este subcapítulo descreve e demonstra como utilizar o sistema desenvolvido. Para execução do aplicativo, basta visualizar o ícone na tela do *smartphone* e selecioná-lo com um toque. O aplicativo é denominado “Vuforia RA” visto na Figura 41.

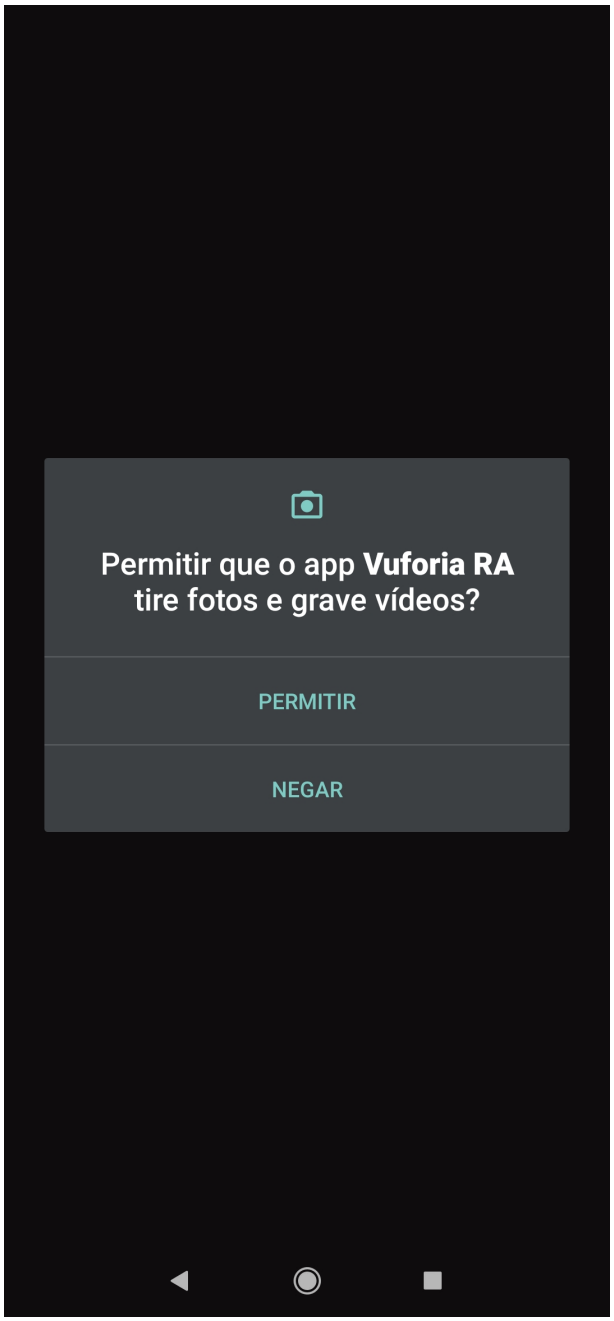
Figura 41 – Ícone do aplicativo instalado



Fonte: AUTOR

Após a execução do aplicativo, será solicitada a permissão para utilização da câmera do *smartphone* como mostra a Figura 42. Como o sistema é baseado em realidade aumentada é necessário conceder a permissão para que seja possível a execução da aplicação sem nenhum problema.

Figura 42 – Solicitação de permissão para uso da câmera



Fonte: AUTOR

Com a permissão da câmera efetuada, é possível identificar que a aplicação está em execução através do logo do SDK Vuforia, presente no canto inferior esquerdo da tela. Figura 43.

Figura 43 – Permissão de acesso a câmera ativa

Fonte: AUTOR

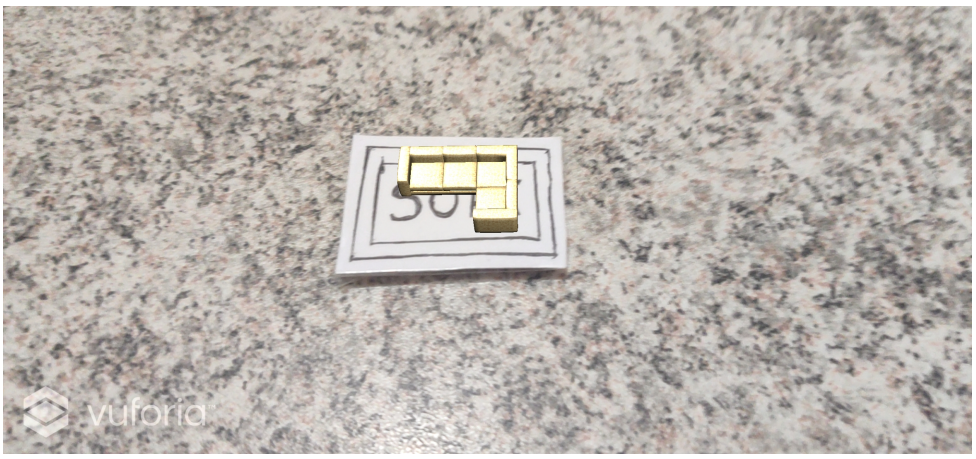
Em seguida, basta posicionar um dos marcadores criados em frente à câmera, conforme demonstra a Figura 44. Desta forma, a aplicação renderizará o objeto correspondente ao que está escrito no marcador.

Figura 44 – Renderização do objeto 3D do sofá

Fonte: AUTOR

Para fazer a troca dos modelos dos objetos, é necessário que o usuário toque nos marcadores através da aplicação. Após tocar no marcador a aplicação trocará o objeto como mostra a FIG

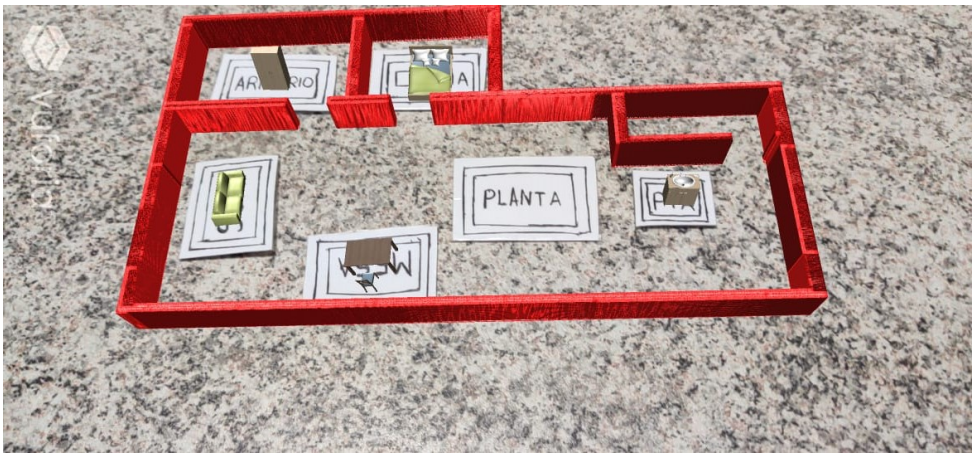
Figura 45 – Renderização do objeto 3D do sofá após o toque no *image target*



Fonte: AUTOR

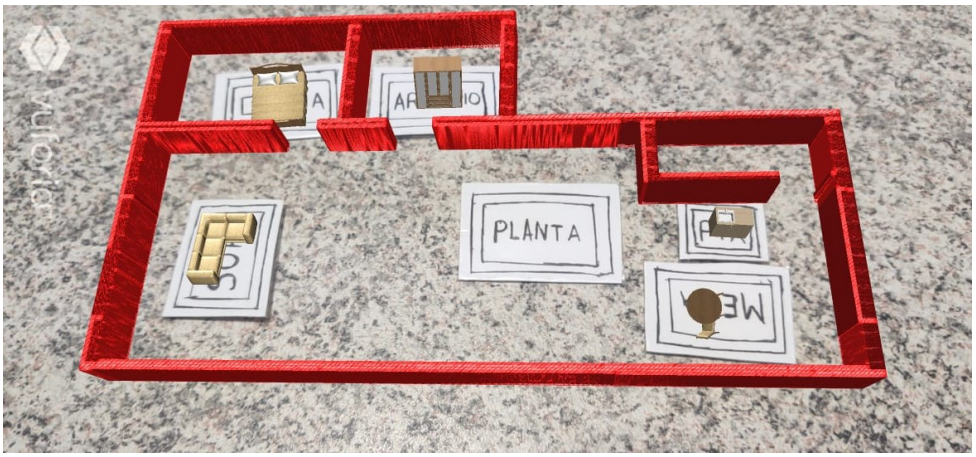
Com a aplicação podemos posicionar todos os objetos em frente à câmera e modelar o design da planta baixa de qualquer forma desejada. Na Figura 46, Figura 47 e Figura 48 é possível visualizar diferentes tipos de combinações possíveis com os marcadores.

Figura 46 – Exemplo de combinação 1



Fonte: AUTOR

Figura 47 – Exemplo de combinação 2



Fonte: AUTOR

Figura 48 – Exemplo de combinação 3

Fonte: AUTOR

6 CONSIDERAÇÕES FINAIS

Com a finalização do desenvolvimento deste trabalho, foi possível perceber o quanto a realidade aumentada pode ajudar designers de interiores em tomadas de decisões e de como ela torna a experiência interativa e satisfatória para o cliente.

A utilização do Vuforia para a aplicação da realidade aumentada foi de grande ajuda, devido a sua configuração ser simples, sua integração fácil com a Unity e seu banco de dados de fácil manipulação.

O 3ds *Max* foi de grande importância para o resultado final da aplicação. Sem ele não haveria desempenho para uma boa utilização da mesma. Com sua função de unir os corpos, foi possível simplificar a grande quantidade objetos que representavam a planta baixa em um único componente. Logo obteve-se um grande aumento desempenho, tornando-a assim, uma aplicação utilizável em tempo real.

A utilização da Unity na versão 2019.4.2.f1 juntamente com o Unity Hub possibilitou a configuração do SDK do Android com uma simples marcação de *checkbox*, o que facilitou ainda mais o desenvolvimento da aplicação.

O trabalho apresentado serve como uma referência para aprendizado e demonstração das ferramentas Unity e Vuforia. Trazendo também uma base com conhecimentos iniciais referentes à realidade aumentada.

A aplicação desenvolvida neste trabalho cumpriu com os resultados esperados, sendo capaz de facilitar o trabalho de designers, dando mais flexibilidade e visibilidade em seus projetos. Para esta aplicação foram criados seis marcadores para a utilizar em sua primeira versão.

Como trabalhos futuros, destaca-se a implementação da técnica de *heightmap* para criação das plantas baixas a partir de fotos das mesmas, fazendo com que a aplicação não dependa do 3ds *Max* para auxiliar em modelagens para ganho de desempenho, tornando a aplicação mais independente. A criação de marcadores através da aplicação utilizando a API do Vuforia. Salvar combinações dos modelos 3D representados pelos marcadores trazendo mais agilidade. Posteriormente, realizados os ajustes e melhorias propostas, o aplicativo terá plenas condições de publicação na loja da *Play Store* para acesso ao público.

REFERÊNCIAS

ALMEIDA, M. **Desvendando o 3ds Max**. [S.l.]: Universo dos Livros Editora, 2007. Citado na página 18.

ARCHADEMY. **Como a realidade virtual e aumentada tem moldado a arquitetura**. [S.l.], 2020. Disponível em: <https://www.archademy.com.br/blog/realidade-virtual-e-aumentada/>. Acesso em: 31 de outubro de 2020. Citado na página 12.

AUTODESK. **Introdução**. [S.l.], 2015. Disponível em: <http://help.autodesk.com/view/3DSMAX/2017/PTB/?guid=GUID-68FFEEDA-3801-444E-B450-942E03A7523D>. Acesso em: 31 de outubro de 2020. Citado na página 17.

AUTODESK. **Montagem de personagens**. [S.l.], 2020. Disponível em: <http://help.autodesk.com/view/3DSMAX/2020/PTB/?guid=GUID-96C4CC13-0135-4CBB-90D4-EC13153EE44E>. Acesso em: 31 de outubro de 2020. Citado na página 18.

AZEVEDO, E. **Computação Gráfica, Teoria e Prática.** [S.l.], 2003. Disponível em: <https://sobrinhodocliente.wordpress.com/2017/03/24/computacao-grafica-um-breve-resumo/>. Acesso em: 31 de outubro de 2020. Citado na página 18.

AZUMA, R.; BAILLOT, Y.; BEHRINGER, R. **IEEE Computer Graphics and Applications**. [S.l.], 2001. Citado na página 5.

BARROS, T. **Gear VR, óculos de realidade virtual da Samsung, ganha preço no Brasil**. [S.l.], 2015. Disponível em: <https://www.techtudo.com.br/noticias/noticia/2015/11/gear-vr-oculos-de-realidade-virtual-da-samsung-ganha-preco-no-brasil.html>. Acesso em: 10 de junho de 2020. Citado 2 vezes nas páginas 4 e 12.

BIMBER, O.; RASKAR, R. **Spatial augmented reality: merging real and virtual worlds**. [S.l.]: CRC press, 2005. Citado 4 vezes nas páginas 5, 6, 8 e 9.

BROLL, W. et al. An infrastructure for realizing custom-tailored augmented reality user interfaces. **IEEE transactions on visualization and computer graphics**, IEEE, v. 11, n. 6, p. 722–733, 2005. Citado na página 6.

CARDOSO, A. **Tecnologias para o desenvolvimento de sistemas de Realidade Virtual e Aumentada**. [S.l.], 2007. Citado na página 4.

COSTA, L. **Considerações sobre arte contemporânea**: Lúcio costa, registro de uma vivência. [S.l.], 1940. Citado na página 11.

CRAIG, E. **Meta 2 AR Glasses – A Preview of our Future?** [S.l.], 2016. Citado na página 7.

CUPERSCHMID, A.; FREITAS, M. d.; RUSCHEL, R. Tecnologias que suportam realidade aumentada empregadas em arquitetura e construção. **Cadernos do PROARQ**, p. 47–69, 2012. Citado 4 vezes nas páginas 6, 7, 8 e 9.

CYRELA. **COMO A REALIDADE AUMENTADA PODE AJUDAR NA DECORAÇÃO?** [S.l.], 2020. Disponível em: <https://blog.cyrela.com.br/realidade-aumentada/>. Acesso em: 16 de junho de 2020. Citado na página 12.

DECORAFACIL. **Arquitetura: o que é, conceito, estilos e breve história**. [S.l.], 2019. Disponível em: <https://www.decorfacil.com/arquitetura/>. Acesso em: 16 de junho de 2020. Citado na página 11.

ESTUDANTE, G. do. **Design de Interiores**. [s.n.], 2019. Disponível em: <https://guiadoestudante.abril.com.br/profissoes/design-de-interiores/>. Acesso em: 16 de junho de 2020. Citado na página 10.

FILHO, J. G. **Design do objeto: bases conceituais**. [S.l.], 2006. Citado na página 10.

FIORINI, C. P. V. V. G. C. **A atuação design de interiores em espaços de oficinas para centros de referência de assistência social CRAS**. [S.l.], 2020. Disponível em: <https://revistaintramuros.com.br/atuacao-design-de-interiores-em-espacos-de-oficina-edicao-01/>. Acesso em: 16 de junho de 2020. Citado na página 10.

GASPAROTTO, H. M. **Unity 3D: Introdução ao desenvolvimento de games**. [S.l.], 2014. Disponível em: <https://www.devmedia.com.br/unity-3d-introducao-ao-desenvolvimento-de-games/30653>. Acesso em: 31 de outubro de 2020. Citado na página 14.

GRIMBERG, J. **As novidades da tecnologia na arquitetura e no design de interiores**. [S.l.], 2017. Disponível em: <https://archtrends.com/blog/as-novidades-da-tecnologia-na-arquitetura-e-no-design-de-interiores/>. Acesso em: 16 de junho de 2020. Citado na página 12.

GUBERT, M. L. **Design de Interiores: a padronagem como elemento compositivo no ambiente contemporâneo**. [S.l.], 2011. Citado na página 10.

HALO. **Tecnologia: o que a realidade aumentada é capaz de fazer na construção civil?** [S.l.], 2020. Disponível em: <https://halonotoriedade.com.br/tecnologia-o-que-a-realidade-aumentada-e-capaz-de-fazer-na-construcao-civil/>. Acesso em: 16 de junho de 2020. Citado na página 13.

KIRNER, C.; ZORZAL, E. R. Aplicações educacionais em ambientes colaborativos com realidade aumentada. In: **Brazilian Symposium on Computers in Education (Simpósio Brasileiro de Informática na Educação-SBIE)**. [S.l.: s.n.], 2005. v. 1, n. 1, p. 114–124. Citado 2 vezes nas páginas 7 e 8.

KUCERA, J. **Desenvolva jogos com a Unity 3D**. [S.l.], 2013. Disponível em: <https://www.devmedia.com.br/desenvolva-jogos-com-a-unity-3d/29125>. Acesso em: 31 de outubro de 2020. Citado na página 14.

LAVIOLA, J. J. et al. **3D user interfaces: theory and practice**. [S.l.]: Addison-Wesley Professional, 2017. Citado na página 5.

MICROSOFT. **Introdução à linguagem C# e .NET**. [S.l.], 2020. Disponível em: <https://docs.microsoft.com/pt-br/dotnet/csharp/getting-started/>. Acesso em: 10 de novembro de 2020. Citado na página 17.

MILGRAM, P. **Augmented Reality: A Class of Displays on the Reality-Virtuality Continuum, Telemanipulator and Telepresence Technologies**. [S.l.], 1994. Citado na página 3.

NETTO, A. **Definições, Dispositivos e Aplicações**. [S.l.], 2002. Citado na página 4.

PASSOS, E. B. et al. Tutorial: Desenvolvimento de jogos com unity 3d. In: **VIII Brazilian Symposium on Games and Digital Entertainment**. [S.l.: s.n.], 2009. p. 1–30. Citado na página 14.

RODELLO, S. R. R. S. I. A. **Realidade Misturada: Conceitos, Ferramentas e Aplicações**. [S.l.], 2010. Citado na página 3.

ROMAO, M. M. G. V. P. A. **REALIDADE AUMENTADA: CONCEITOS E APLICAÇÕES NO DESIGN**. [S.l.], 2013. Disponível em: <https://archtrends.com/blog/as-novidades-da-tecnologia-na-arquitetura-e-no-design-de-interiores/>. Acesso em: 16 de junho de 2020. Citado na página 12.

SOUZA, E. de. **Pokémon Go ganhará nova realidade aumentada no Android e 4ª geração em breve**. [S.l.], 2018. Disponível em: <https://olhardigital.com.br/noticia/pokemon-go-ganhara-nova-realidade-aumentada-no-android-e-4-geracao-em-breve/79111>. Acesso em: 10 de junho de 2020. Citado na página 5.

TERRA. **Segundo dados, mercado de decoração teve crescimento de 23de faturamento no primeiro trimestre de 2017**. [S.l.], 2017. Disponível em: <https://www.terra.com.br/noticias/dino/segundo-dados-mercado-de-decoracao-teve-crescimento-de-23-de-faturamento-no-primeiro-/trimestre-de-2017,ffcea57a5edc204236f7d7b59699d5739diph57e.html>. Acesso em: 2 de julho de 2020. Citado na página 1.

UNITY. **Installing from a registry**. [S.l.], 2020. Disponível em: <https://docs.unity3d.com/Manual/upm-ui-install.html>. Acesso em: 31 de outubro de 2020. Citado na página 28.

UNITY. **Plataforma do Unity**. [S.l.], 2020. Disponível em: <https://unity.com/pt/products/unity-platform>. Citado na página 14.

VUFORIA. **Vuforia developer Library**. [S.l.], 2020. Disponível em: <https://library.vuforia.com/getting-started/overview.html>. Acesso em: 31 de outubro de 2020. Citado 2 vezes nas páginas 14 e 16.

ZAPP2PHOTO. **Industrial 4.0 Augmented reality concept. Hand holding tablet with AR service**. [S.l.], 2017. Disponível em: <shorturl.at/ijsBI>. Citado na página 9.