

**CENTRO DE EDUCAÇÃO PROFISSIONAL DE TIMBÓ**

**CURSO TÉCNICO EM INFORMÁTICA**

**PROTÓTIPO DE APLICATIVO DE MODELAGEM DE  
AMBIENTES EM REALIDADE VIRTUAL PARA O  
METAVERSO**

**GUSTAVO HENRIQUE CAMPESTRINI  
MICHEL NIENOW DE BARROS**

**TIMBÓ  
2022**

# **PROTÓTIPO DE APLICATIVO DE MODELAGEM DE AMBIENTES EM REALIDADE VIRTUAL PARA O METAVERSO**

Por

**GUSTAVO HENRIQUE CAMPESTRINI  
MICHEL NIENOW DE BARROS**

Trabalho aprovado em sua forma final pelo  
Curso Técnico de Informática do Centro de  
Educação Profissional de Timbó - CEDUP

---

Prof. Roberto Luiz Debarba – Orientador

---

Prof. Edésio Marcos Slomp, Me.

---

Prof. André Luiz Gretter

---

Prof. Rutineia Luciene Bell

Timbó, 18 de julho de 2022

## **AGRADECIMENTOS**

Primeiramente, gostaríamos de expressar nossos sinceros agradecimentos a todos que contribuíram para o nosso trabalho.

Em segundo lugar, gostaríamos de agradecer nossos orientadores e professores, especialmente Prof. Roberto Luiz Debarba, Prof. Edésio Marcos Slomp, Prof. André Luiz Gretter e Prof. Rutineia Luciene Bell por seu tempo, orientação e apoio.

Por fim, mas não menos importante, gostaríamos de agradecer a todos os nossos amigos e familiares que nos deram apoio e acreditaram em nosso projeto durante todo o processo.

## RESUMO

Já há algum tempo, é possível modelar projetos de ambientes de realidade virtual, porém o custo para contratar este serviço costuma ser muito elevado, e para aprender a fazê-lo, exige tempo e dedicação, o que obstaculiza o uso da tecnologia para os pequenos e médios empreendedores do mercado imobiliário, que geralmente dispõe dos próprios recursos para impulsionar seus resultados. Visando solucionar este problema foi desenvolvido um protótipo de aplicativo para modelagem de ambientes em realidade virtual para o metaverso. Com este protótipo será possível uma pessoa que mesmo que sem ter experiência e conhecimento em modelagem 3D, consiga recriar sua carteira de imóvel em realidade virtual, podendo carregar uma imagem deste imóvel para usar como referência enquanto modela-o imerso na realidade virtual. Também terá a possibilidade de deixar os projetos disponíveis em VR no metaverso para que seus potenciais clientes consigam visitá-los e desfrutar o potencial do imóvel a qualquer momento e de qualquer lugar do mundo, em uma experiência imersiva através de dispositivos de realidade virtual como o Meta Quest 2. Para o desenvolvimento deste protótipo utilizou-se as ferramentas: Unity como motor de jogo, as linguagens de programação C# e PHP, o banco de dados relacional MySQL. O protótipo desenvolvido se mostrou viável e funcional para o propósito para o qual foi criado. Ele conseguiu atender às necessidades dos usuários, permitindo a criação de projetos de ambientes em VR, a partir de uma biblioteca de objetos como paredes e móveis pré-moldados e pré-configurados, permitindo assim praticidade ao modelar projetos.

Palavras-chave: metaverso; realidade virtual; modelagem 3D.

# SUMÁRIO

<b>1 INTRODUÇÃO .....</b>	<b>6</b>
1.1 OBJETIVOS DO TRABALHO .....	7
1.1.1 Objetivo Geral .....	7
1.1.2 Objetivos Específicos .....	7
1.1.3 Estrutura do trabalho .....	8
<b>2 FUNDAMENTAÇÃO TEÓRICA .....</b>	<b>9</b>
2.1 REALIDADE VIRTUAL .....	9
2.2 META QUEST 2 .....	9
2.3 METAVERSO .....	10
2.4 UNITY .....	11
2.5 MYSQL .....	12
2.6 PROTÓTIPO .....	12
2.7 C# .....	13
2.8 GOOGLE SIGN IN API .....	14
2.9 CPANEL .....	14
2.10 PHP .....	14
2.11 HTML .....	15
2.12 CSS .....	15
2.13 MODELAGEM 3D .....	16
2.14 LINGUAGEM DE PROGRAMAÇÃO .....	16
2.15 API .....	17
2.16 CLOUD COMPUTING .....	17
<b>3 DESENVOLVIMENTO DO PROTÓTIPO .....</b>	<b>19</b>
3.1 REQUISITOS PRINCIPAIS DO PROBLEMA A SER TRABALHADO .....	19
3.1.1 Requisitos Funcionais .....	19
3.1.2 Requisitos Não Funcionais .....	20
3.2 ESPECIFICAÇÃO .....	20
3.2.1 Diagrama de Casos de Uso .....	20
3.2.2 Especificação dos Casos de Uso .....	21
3.2.3 Diagrama de atividades .....	22
3.2.4 Diagrama de Classes .....	24

3.2.5 Diagrama MER .....	24
3.2.5.1 Modelo Conceitual.....	25
3.2.5.2 Modelo Lógico.....	25
<b>3.3 IMPLEMENTAÇÃO .....</b>	<b>26</b>
3.3.1 Técnicas e ferramentas utilizadas.....	27
3.3.1.1 Interface VR.....	27
3.3.1.1.1 XR Interaction Toolkit .....	27
3.3.1.1.2 Métodos de Execução .....	29
3.3.1.1.3 Objetos de Cena .....	29
3.3.1.1.4 Métodos para Salvar e Carregar Projeto .....	30
3.3.1.2 Interface Web.....	31
3.3.1.2.1 Gerenciamento dos Projetos.....	32
3.3.1.2.2 API Google Sign-in.....	32
3.3.1.2.3 Token de Acesso .....	33
3.3.1.3 Banco de Dados .....	34
3.3.1.4 Serviços na Nuvem .....	35
3.3.1.4.1 Hospedagem.....	35
<b>4 CONCLUSÕES .....</b>	<b>36</b>
<b>REFERÊNCIAS .....</b>	<b>37</b>
<b>ANEXO A – Questionário utilizado para o levantamento de requisitos .....</b>	<b>38</b>

## 1 INTRODUÇÃO

Que a realidade virtual<sup>1</sup> pode proporcionar inúmeras possibilidades não é exatamente uma novidade, mas sua crescente popularidade fez com que grandes empresas investissem em seu potencial, como a *Meta*, antigo *Facebook*, que em 2021, anunciou a mudança oficial do nome, fazendo referência ao termo metaverso, um conceito de tecnologia que busca não apenas diminuir as fronteiras entre o que é real do que é virtual, mas transformar ambos em uma só coisa. Este movimento despertou um grande interesse do mercado, e aumentou exponencialmente as vendas de dispositivos de realidade virtual, em especial do *Meta Quest 2*, um dispositivo de realidade virtual da própria *Meta* que oferece ao público uma das melhores, mais acessíveis e imersivas experiências de realidade virtual, chegando a custar menos que os smartphones mais cobiçados do momento. Ainda sobre a realidade virtual, vale ressaltar que existem dispositivos menos avançados que conseguem proporcionar algumas experiências de realidade virtual, por exemplo, headsets<sup>2</sup>, que funcionam como um suporte para transformar o smartphone em um dispositivo de realidade virtual.

Visando o potencial destas tecnologias, e o interesse do mercado no tema, foi pensado um protótipo de aplicação em realidade virtual que solucione o alto custo e a alta complexidade de modelagem 3D de interiores para os pequenos empreendedores do mercado imobiliário e a disponibilização destes modelos no metaverso, o **VRPlaces**. Utilizando este protótipo, um corretor pode por exemplo modelar os imóveis da sua carteira em poucos passos na realidade virtual e disponibilizá-la para visita de clientes a partir de qualquer lugar do mundo (metaverso).

Atualmente, é possível modelar diversos projetos em 3D, e até mesmo torná-los acessíveis na realidade virtual, porém o custo para contratar este serviço ainda é muito elevado, e para aprender a fazê-lo, exige algum tempo e dedicação, o que obstaculiza o uso da tecnologia para o empreendedor da ponta, que geralmente dispõe dos próprios recursos para impulsionar seus resultados.

Uma vez que o usuário tenha em mãos a planta baixa do imóvel que deseja apresentar na realidade virtual, basta realizar o carregamento deste arquivo na nuvem e seguir o

---

<sup>1</sup> Termo utilizado para referir-se ao ambiente gerado por computação gráfica, onde o usuário pode interagir simulando uma realidade

<sup>2</sup> São dispositivos que se ajustam à cabeça do usuário.

passo a passo para disponibilizar a planta dentro do protótipo. Agora, utilizando o dispositivo de realidade virtual *Meta Quest 2*, o usuário pode iniciar o software e começar a modelar este imóvel com as ferramentas disponíveis para criar as paredes, definir as portas e janelas, personalizar as cores e texturas do chão, do teto e das paredes, como também mobiliar e decorar o ambiente, tudo, sem a necessidade de um computador de alto desempenho, sem contratar profissionais especializados, ou nem mesmo estudar ferramentas de modelagem 3D e desenvolvimento de *software*.

Para modelar este projeto serão utilizadas ferramentas como *Unity 3D* para a parte visual, linguagem C# para a parte funcional, o banco de dados será *MySQL*, *Meta Quest 2* como dispositivo compatível com o projeto de software.

## 1.1 OBJETIVOS DO TRABALHO

Os objetivos apresentado a seguir foram desenvolvidos com base nas respostas do questionário, de acordo com APÊNDICE A. E expressam de forma amplas quais os objetivos almejados com o protótipo.

### 1.1.1 Objetivo Geral

Permitir a modelagem de imóveis em VR através de dispositivos de realidade virtual, criando assim, projetos imersivos que possam ser visitados de qualquer lugar do mundo, e então desfrutar do potencial do imóvel.

### 1.1.2 Objetivos Específicos

- Solucionar o alto custo de modelagem de ambientes VR, que atualmente necessitam de profissionais específicos.
- Disponibilizar uma ferramenta imersiva, com objetos pré-moldados e pré-configurados.
- Viabilizar showroom em VR para dispositivos como, smartphones e dispositivos de realidade virtual.



### 1.1.3 Estrutura do trabalho

Este trabalho está estruturado em quatro capítulos, sendo o primeiro a introdução e os objetivos do tema proposto, além de conter a estrutura do trabalho.

No segundo capítulo refere-se à fundamentação teórica, apresentando os principais conceitos e ferramentas de acordo com o tema escolhido.

O terceiro capítulo trata do desenvolvimento do trabalho, onde será retratado a especificação, análise e detalhes da implementação. Além de ser apresentado os diagramas de casos de uso e de atividades.

Por fim, no quarto capítulo, se encontram as conclusões feitas acerca do desenvolvimento do protótipo, expondo os resultados obtidos, e apresentado as dificuldades encontradas, e sugestões para a continuidade deste projeto.

## 2 FUNDAMENTAÇÃO TEÓRICA

Neste capítulo será abordado a fundamentação teórica que se trata de um conjunto de princípios que sustentam ou embasam a realização de um projeto de pesquisa.

### 2.1 REALIDADE VIRTUAL

O termo "realidade virtual"(RV) é creditado a Jaron Lanier, e como o próprio nome já diz é um ambiente virtual gerado por meio de computação gráfica, com objetos e cenas que pareçam reais, onde os usuários possam se movimentar, ver por diferentes ângulos, ver objetos, agarrá-los e até mesmo remodelá-los. Assim, o usuário tem uma percepção de imersão nessa realidade, que pode ser acessada através de dispositivos como o Meta Quest 2.

Latta (1994) define a Realidade Virtual como uma interface homem-máquina que simula um mundo virtual, permitindo que os usuários interajam com ele. Esta interface segundo ela é considerada a mais avançada tecnologia disponível atualmente, pois busca imergir o usuário dando-lhes sensações e informações sobre o mundo virtual como se ele realmente existisse.

Em essência, a realidade virtual é literalmente o que o nome sugere, uma realidade, porém virtual, seja esta realidade uma representação do que existe ou um devaneio, uma ideia, do que poderia existir, ainda que essa existência seja restrita ao meio virtual.

### 2.2 META QUEST 2

O Meta Quest 2 é um dispositivo eletrônico de realidade virtual desenvolvido e distribuído pela empresa Meta. O dispositivo acompanha dois controles remotos rastreáveis, para simular as mãos do usuário dentro da realidade virtual.

Com quatro câmeras distribuídas, o Meta Quest 2 rastreia o ambiente de maneira precisa, conseguindo identificar a posição e a altura onde se encontra, além de atualizar com fidelidade a movimentação através da imagem capturada por estas câmeras. Além de rastrear o ambiente desta forma, ele usa o mesmo sistema para

rastrear os controles, uma vez que estejam no campo de visão, serão exibidos na posição equivalente na realidade virtual e caso prefira, o usuário pode usar diretamente as mãos, que também serão mapeadas e inseridas na realidade virtual.

O Meta Quest 2 é principalmente um console de realidade virtual, sendo amplamente usado para jogos, mas também como uma ferramenta de treinamento e de trabalho, já que ele consegue traduzir com precisão a movimentação do usuário dentro da realidade virtual, permitindo que ferramentas antes utilizadas através de uma tela de duas dimensões, agora possam ser manipuladas dentro desta realidade.

Com essa gama de recursos e possibilidades, o Meta Quest 2 se mostrou a opção ideal para o desenvolvimento desta ferramenta, garantindo a qualidade e precisão dos projetos, e também o desempenho esperado. Atualmente seu preço segundo o site oficial da fabricante sua versão de entrada custa US\$299, convertendo para real este valor fica R\$1616,93 na cotação de 16 de julho de 2022. Na Figura 01 está ilustrado o dispositivo.

Figura 01 - Meta Quest 2



Fonte: Site oficial do Meta: <https://store.facebook.com/help/quest/>

## 2.3 METAVERSO

Entender a origem e etimologia da palavra "metaverso", é mais simples do que seu conceito. Segundo Zompero (2022), "meta" vem do grego, que significa "além de", "no meio de", "entre", representando também, a ideia de mudança e transcendência,

enquanto “verso”, se refere a universo, que pode ser explicado pelas leis da física, incluindo o espaço e o tempo.

Quanto ao conceito, é onde encontramos mais dificuldade na compreensão, pois trata-se de algo que ainda está surgindo e se transformando, porém podemos considerar alguns de seus aspectos mais proeminentes, como o estreitamento entre a nossa realidade, e as tecnologias que usamos para acessar o mundo virtual.

Hoje quando se pensa em alugar ou comprar um imóvel, o método mais comum é acessar um site e visualizar as fotos e vídeos do imóvel de nosso interesse, mas e se ao invés de acessar um site, fosse possível visitar um condomínio virtual, e conhecer os ambientes que estão disponíveis para compra como se estivesse no próprio local? Conhecer a vizinhança, experimentar a vista do lugar, e tudo isso sem estar fisicamente ali, mas virtualmente. Esta possibilidade pode existir com o metaverso, pois é essa uma de suas características, nos aproximar do que está fisicamente distante, através da tecnologia, e deixar a experiência virtual cada vez mais próxima da experiência real.

O metaverso não é um lugar, não é um site, não é um produto que se possa tocar, mas um conceito, que busca englobar todos os aspectos da vida cotidiana e transformar as nossas interações com o mundo real e virtual em algo cada vez mais dinâmico e acessível, onde visitar um lugar, conversar com alguém, acessar experiências, seja algo tão casual quanto olhar para o relógio no pulso e ver as horas, porém tão imersivo e incrível quanto ir no show da sua banda favorita, ainda que esta esteja no outro lado do mundo.

## 2.4 UNITY

Lançado em 2005 pela Unity Technologies, o Unity é um dos mais conhecidos e completos motores de desenvolvimento de jogo, que permite o desenvolvimento de jogos 2D e 3D, para mais de dez plataformas, dentre elas Windows, Linux, consoles, VR e AR.

Chama-se motor de desenvolvimento este tipo específico de ferramenta que acopla consigo uma série de funcionalidades, como a capacidade de criar e operar de maneira interativa, ambiente de jogos, filmes, animações e outros produtos digitais,

sistemas de som e iluminação, programação de inteligência artificial e de scripts, diversos tipos de simulações entre outros.

Com sua ampla documentação, diversidades de plataformas suportadas e sendo uma ferramenta de fácil utilização, o motor de jogo Unity se tornou um dos mais populares motores de jogos [Petridis et al., 2010, Anderson et al., 2013].

O Unity Editor possui uma interface com o conceito de “What You See Is What You Get”, que permite que o usuário desenvolva seus jogos através do "arrastar e soltar"(drag-and-drop). Com sua vasta documentação e sua grande comunidade desenvolvidora que auxilia novos usuários por meios de fóruns e páginas online, a Unity torna-se uma ferramenta de fácil utilização.

## 2.5 MYSQL

MySQL é um sistema de gerenciamento de banco de dados relacional (SGBDR), desenvolvido originalmente em 1994 pela empresa sueca MySQL AB, fundada por, David Axmark, Allan Larsson e Michael Widenius. Atualmente é mantido pela companhia de software Oracle.

Uma das grandes vantagens do MySQL está na licença de uso. Trata-se de uma ferramenta de código fonte aberto (Open Source), basicamente ela possui duas formas de uso: um software livre, que é baseada nas cláusulas GNU-PL (General Public Licence) e outra para uso comercial. Sobre a licença comercial, existe para algumas situações em que é necessário embutir MySQL em aplicações comerciais, ou suporte específico, ou mesmo aquisição de outras ferramentas de apoio. O MySQL é um SGBD portátil a diversos Sistemas Operacionais, onde se destacam Linux, Unix, FreeBSD, Mac OS X Server, Windows e linguagens de programação como Java, Python, PHP, Perl, C, C++, entre outras. (ALMEIDA, 2007).

Sendo assim o MySql permite a persistência de dados e implementação de funcionalidades que são utilizadas para armazenar informações solicitadas pelo software.

## 2.6 PROTÓTIPO

O termo protótipo refere-se a uma técnica de desenvolvimento de software baseada em modelos, em que se cria um modelo, para simular, exemplificar, o comportamento principal do sistema projetado. Essa abordagem permite que os

desenvolvedores explorem e testem diferentes soluções, e levantem os requisitos necessários para o funcionamento do projeto antes de implementarem o código final.

Segundo o livro *Design Thinking – Inovação em negócios* 1, “o protótipo é a tangibilização de uma ideia, a passagem do abstrato para o físico de forma a representar a realidade – mesmo que simplificada – e propiciar validações. ”

Existem diversas maneiras para criar um protótipo, utilizando ferramentas, como papel e caneta, programas de computador ou até mesmo prototipagem rápida (impressão 3D).

No geral, protótipos são um tipo de ferramenta usada para criar e testar novos produtos antes de investir tempo e dinheiro em sua versão final. Eles são úteis para testar funcionalidades, recolher feedbacks, avaliar seu potencial de sucesso no mercado, além de permitir que clientes e desenvolvedores visualizem e explorem o sistema em construção, de forma a ajustá-lo de acordo com suas necessidades.

## 2.7 C#

Quando se fala em projetos computacionais, não há como não falar também em linguagem de programação, e é disto que se trata o C# (C sharp), uma linguagem de programação. Fortemente tipada, e orientada a objetos, o C# é uma das mais conhecidas linguagens de programação do mercado, além de fácil aprendizado, sua aplicação é ampla, graças ao fato da mesma ser desenvolvida pela empresa Microsoft, mesma empresa do popular sistema operacional Windows.

Com uma vasta biblioteca, o C# se conecta com vários recursos necessários para o desenvolvimento de aplicações para o metaverso, como por exemplo uma biblioteca chamada Unity XR, que permite a integração do ambiente virtual construído no Unity com o dispositivo VR Meta Quest 2, e o banco de dados MySQL, ambos presentes neste projeto.

Por fim, o C# também conta com uma grande comunidade de desenvolvedores, o que garante maior suporte para o desenvolvimento de aplicações nesta linguagem além de soluções muito mais robustas.

## 2.8 GOOGLE SIGN IN API

O Google Sign In é uma API (Application Programming Interface) que permite aos usuários se cadastrarem e se autenticarem a sites ou aplicativos usando suas credenciais do Google.

Ao escolher o Google como método de cadastro, o usuário insere seu e-mail e senha no formulário de acesso Google e a plataforma viabiliza o resto do processo, poupando o usuário de escrever todos os seus dados, pois os mesmos serão automaticamente inseridos pela API.

Outro ponto forte é que fornece uma maneira conveniente e segura de acessar aplicativos e sites com apenas um clique, já que uma vez cadastrado, a API autentica a solicitação de acesso sem pedir a senha a cada vez.

## 2.9 CPANEL

O cPanel é um painel de controle, usado para gerenciar hospedagem de forma simplificada de diversos tipos de produtos digitais, por exemplo servidores de sites, banco de dados, e-mails e etc.

Por meio dele é possível instalar aplicações, monitorar o desempenho das páginas e serviços que estão hospedados, além de disponibilizar a configuração de todos os recursos em um só ambiente.

Em concordância com L. Andrei, o cPanel está se tornando o mais popular Painel de Controle, rodando em milhões de websites espalhados pelo mundo, mostrando-se uma excelente opção pela sua versatilidade e fácil utilização.

## 2.10 PHP

Lançado em 1994, PHP é hoje uma conhecida linguagem de programação utilizada principalmente, mas não exclusivamente para criação de sites. Além da sua compatibilidade com HTML, essa tecnologia funciona nos servidores, tirando a responsabilidade do processamento da máquina do usuário.

Embora originalmente PHP significasse “Personal Home Page Tools”, sua evolução funcional levou a mudança do significado de seu nome, passando a ser um

acrônimo para “Hypertext Preprocessor”, que descreve melhor as possibilidades da tecnologia, pois processa os scripts no servidor antes de enviar os retornos para a tela do usuário.

Graças ao processamento dos comandos ser executado pelo servidor, as páginas de internet desenvolvidas em PHP podem ser mais dinâmicas, entregar funções mais complexas, parecidas com programas de computador, e também mais protegida, uma vez que os as linhas de comandos ficam exclusivamente no servidor, ficando visível, apenas aquilo que é pertinente ao visitante da página.

## 2.11 HTML

HTML (acrônimo para HyperText Markup Language) é uma linguagem de marcação usada para marcar e estruturar elementos de uma página da internet, que são interpretados pelos navegadores (web browser).

Fundada em 1990 por um cientista chamado Tim Berners-Lee, a Linguagem de Marcação de Hipertexto (HTML) foi criada com finalidade inicial de tornar possível o acesso e a troca de informações e de documentação de pesquisas, entre cientistas de diferentes universidades.

De fácil estruturação, o HTML permite o desenvolvimento de páginas de internet usando poucas linhas, separando cada elemento da estrutura em TAGs, que são lidas pelo navegador linha por linha, e traduzida para o usuário no que ele conhece como um website.

## 2.12 CSS

Quando se fala em CSS, é justo afirmar que este é parte fundamental na construção de uma página de internet, não por ser responsável pela codificação da página, mas por ser a tecnologia que define como toda a visualização de uma página deverá ser exibida.

O CSS foi desenvolvido em 1996, por um consórcio de empresas chamado W3C. Devido às limitações do HTML no que diz respeito a estilização, o CSS aparece como um complemento praticamente inseparável, pois se comparar o HTML a uma casa recém construída, mas sem acabamento, pode-se comparar o CSS ao próprio



acabamento, ditando cada detalhe visual desta casa. Ainda comparando o HTML com uma casa, nesta casa já construída, e sem acabamentos, cada cômodo já tem seus móveis e seus objetos distribuídos respectivamente. Neste cenário, o CSS é quem vai dizer as cores da parede, o piso de cada cômodo, a disposição de cada móvel e de cada objeto.

O CSS não é parte intrínseca do HTML, ou seja, não há a obrigatoriedade de usá-lo, mas como visto, sua implementação é de grande importância no desenvolvimento de páginas de internet e por isso fazê-lo é quase sempre uma unanimidade em projetos voltados para a internet.

### 2.13 MODELAGEM 3D

Modelo, modelar, forjar, dar forma à, são algumas definições semelhantes que podem introduzir o conceito de modelagem. Neste caso, dar forma a um objeto do mundo real, em um ambiente virtual, é o que se chama de modelagem 3D, pois se cria uma representação fidedigna a algo que existe, em todas as suas três dimensões.

Na computação gráfica, a modelagem 3D é amplamente utilizada nas mais diversas áreas, como na prototipagem de produtos, jogos, filmes, vestuário, estudos geográficos, astronomia, arquitetura, engenharia e muito mais.

Essa versatilidade nem sempre se restringe às formas de um modelo, mas também ao seu comportamento, simulando movimentação, reação, estrutura e aparência como um todo, o que faz da modelagem 3D um recurso extremamente útil e de aplicação abrangente.

### 2.14 LINGUAGEM DE PROGRAMAÇÃO

Uma linguagem de programação é um conjunto de regras sintáticas e semânticas que definem uma representação de um programa de computador. Uma linguagem de programação permite que um programador escreva um programa em uma forma que possa ser traduzida para um formato de execução pelo computador.

Uma linguagem de programação é um método padronizado que usamos para expressar as instruções de um programa a um computador programável. Ela segue um conjunto de regras sintáticas e semânticas para definir um

programa de computador. Regras sintáticas dizem respeito à forma de escrita e regras semânticas ao conteúdo. (REGINALDO GOTARDO, 2015)

Existem diversas linguagens de programação atualmente, algumas a sintaxe se aproxima a uma linguagem humana, por isso entram na categoria das linguagens de alto nível, como o C# e Python. Por outro lado, existem as linguagens que possuem sintaxe e semântica próximas ao código de máquina, portanto, classificadas como linguagens de baixo nível, por exemplo o Assembly.

## 2.15 API

API (Application Programming Interface) é um conjunto de rotinas e padrões de programação para acesso a um aplicativo de software ou plataforma web. Ela fornece aos desenvolvedores acesso a funcionalidades e dados para criar aplicativos compatíveis com um determinado sistema.

As APIs são úteis quando você precisa permitir que outras pessoas ou companhias acessem seus dados, mas não quer que elas tenham acesso total ao seu código-fonte. As APIs também podem ser usadas para simplificar o processo de desenvolvimento, permitindo que os desenvolvedores reutilizem código já existente e evitando a necessidade de reimplementar as mesmas funcionalidades em vários aplicativos.

## 2.16 CLOUD COMPUTING

O termo cloud computing surge da junção de dois conceitos: a computação em nuvem e a evolução atual do mundo digital. A computação em nuvem é uma tecnologia em que o usuário é capaz de obter acesso às informações e aos aplicativos em qualquer lugar do mundo. A segunda parte do termo, “computing”, é uma tradução literal da expressão em inglês “computação”, que pode ser interpretada como um conjunto de operações de processamento de dados, que refere a entrega de serviços de computação, incluindo servidores, banco de dados, rede, software, análise e inteligência.

Existem diversas formas de Cloud Computing, algumas mais seguras que outras, mas todas elas permitem o acesso à informação numa base de dados. A

maioria das formas de Cloud Computing é baseada em plataformas de software, como o Windows Azure ou o Amazon EC2.

### 3 DESENVOLVIMENTO DO PROTÓTIPO

Este capítulo difunde os conceitos e passos para o desenvolvimento do protótipo, através dos diagramas de atividade, diagrama de casos de uso e dos requisitos.

#### 3.1 REQUISITOS PRINCIPAIS DO PROBLEMA A SER TRABALHADO

Para levantamento dos requisitos, foi utilizada a técnica de pesquisa através de questionário (APÊNDICE A), sendo perguntas de múltipla escolha, possibilitando melhor visão sobre prioridades para este protótipo.

##### 3.1.1 Requisitos Funcionais

Os requisitos Funcionais descrevem as principais funcionalidades que o protótipo deve executar e como deve se comportar. A seguir no Quadro 01, estão representados os requisitos seguidos de seus respectivos casos de uso.

Quadro 01 – Requisitos Funcionais

<b>Requisito</b>	<b>Descrição</b>	<b>UC</b>
RF01	O protótipo deve permitir cadastrar usuários através de um website usando a API de login do Google	UC01
RF02	O protótipo deve permitir login no aplicativo VR usando um sistema de token.	UC02
RF03	O protótipo deve permitir a importação de imagem de referência, em formato PNG ou JPG via website.	UC03
RF04	O protótipo deve permitir a modelagem de ambientes VR em realidade virtual.	UC06
RF05	O protótipo deve permitir a aquisição de pacotes avulsos (móveis, texturas, itens) para inserir no projeto.	UC04 e UC08
RF06	O protótipo deve manter projetos dos imóveis	UC05

RF07	O protótipo deve disponibilizar os projetos dos imóveis para visitação, onde o usuário poderá enviar para seus clientes visitarem os imóveis.	UC07
------	---	------

Fonte: Elaborado pelos autores.

### 3.1.2 Requisitos Não Funcionais

Requisito não funcional pode ser descrito como um requisito que atribui a propriedade do sistema ou uma funcionalidade do projeto. No Quadro 02, estão listados os requisitos não funcionais.

Quadro 02 – Requisitos Não Funcionais

Requisito	Descrição
RNF01	O protótipo necessita de acesso à internet.
RNF02	A interface 3D deve ser construída utilizando o motor de jogos Unity.
RNF03	O protótipo deve utilizar o MySQL como banco de dados.
RNF04	O protótipo deve ser compatível com o dispositivo Meta Quest 2.

Fonte: Elaborado pelos autores.

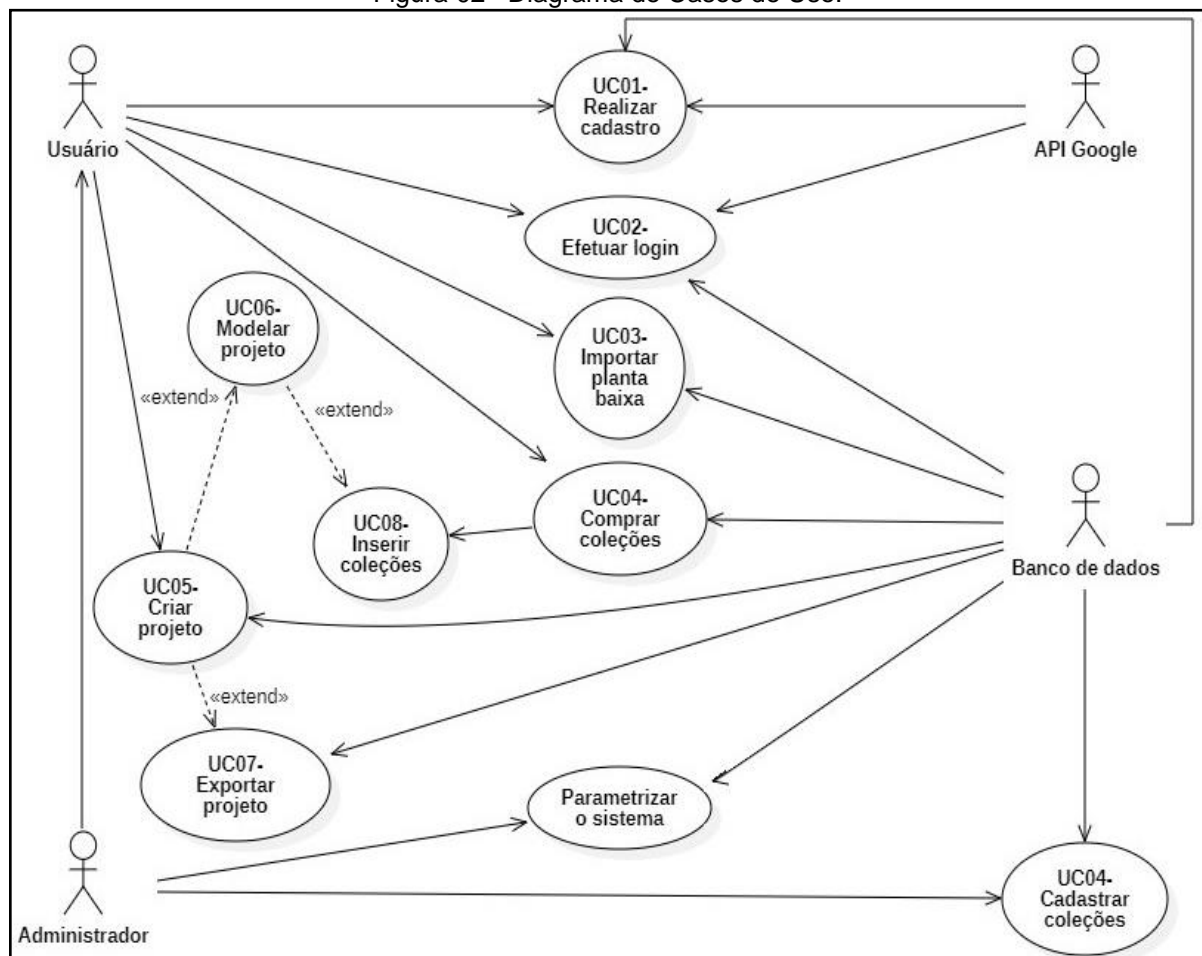
## 3.2 ESPECIFICAÇÃO

Através dos diagramas e notações será apresentada a especificação do protótipo, que é toda a modelagem envolvida para a produção de um projeto de desenvolvimento de software.

### 3.2.1 Diagrama de Casos de Uso

Segundo Pressman (2006, p. 137 e 138) [...] “um caso de uso conta uma história estilizada sobre como o usuário final (desempenhando um de uma série de papéis possíveis) interage com o sistema sob um conjunto de circunstâncias específicas.” Portanto, a Figura 02 representa a visão do usuário sobre o software.

Figura 02– Diagrama de Casos de Uso.



Fonte: Elaborado pelos autores.

### 3.2.2 Especificação dos Casos de Uso

Um modelo de caso de uso é um modelo que descreve como diferentes tipos de usuários interagem com o sistema, que se faz útil para um melhor entendimento das ações de cada caso de uso, assim mostrada no Quadro 03 e Quadro 03.

Quadro 03 - Notação de caso de uso.

UC01 - CRIAR PROJETO	
REQUISITOS ATENDIDOS	RF04 - O protótipo deve permitir que o usuário faça a modelagem do imóvel no aplicativo VR
PRÉ-CONDIÇÕES	Fazer login no website
ATORES ENVOLVIDOS	Usuário, banco de dados

CENÁRIO PRINCIPAL	No VrPlaces web o usuário criará um novo projeto, nomeando e definindo suas características como: tamanho, descrição, e realizar o upload da imagem de referência.
CENÁRIO ALTERNATIVO 01	Caso o usuário decida por não usar imagem de referência, o mesmo pode optar por criar um projeto com referência em branco.

Fonte: Elaborado pelos autores.

Quadro 01 - Notação de caso de uso

UC07 - EXPORTAR PROJETO	
REQUISITOS ATENDIDOS	RF08 – O protótipo deve disponibilizar os projetos dos imóveis para visitação, onde o usuário poderá desfrutar do potencial do imóvel, de maneira imersiva através de dispositivos de realidade virtual.
PRÉ-CONDIÇÕES	Projeto já criado pelo usuário
ATORES ENVOLVIDOS	Usuário
CENÁRIO PRINCIPAL	No VRplaces VR, após o usuário finalizar a modelagem do imóvel, através das opções no “painel de ferramentas” ele irá exportar este projeto para “ambientes”, espaço virtual de visitação, onde o imóvel modelado ficará visível para outros usuários e visitantes do protótipo.

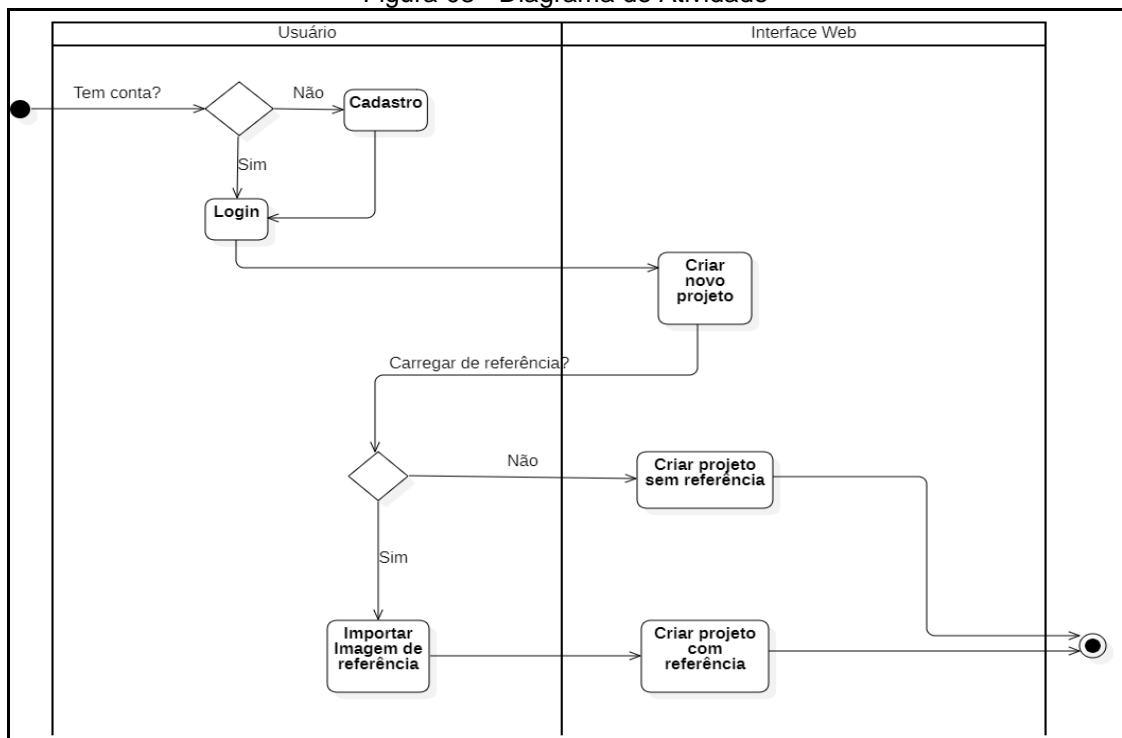
Fonte: Elaborado pelos autores.

### 3.2.3 Diagrama de atividades

O diagrama de atividades apresenta o fluxo de atividade de um único processo. Mostrando como cada atividade depende da outra.

Na Figura 03 o diagrama descreve a ação de criar um novo projeto dentro do protótipo. O ator usuário inicia a sequência, caso tenha uma conta irá fazer o login, caso contrário irá se cadastrar, na sequência decidirá se deseja inserir uma referência no projeto, caso a resposta seja positiva irá inserir a imagem, caso contrário o sistema VrPlaces Web irá finalizar o processo e criará o projeto, para ser acessado no VrPlaces VR.

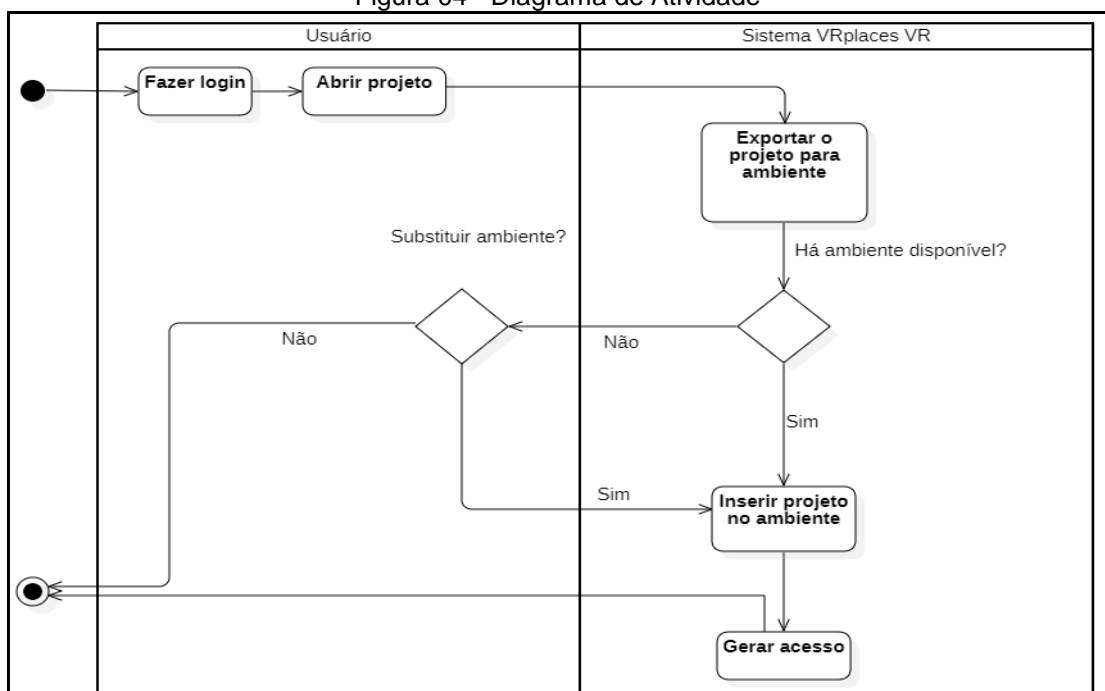
Figura 03 - Diagrama de Atividade



Fonte: Elaborado pelos Autores.

A Figura 04 demonstra o processo de exportar o projeto para ambiente, espaço virtual em que o imóvel modelado ficará disponível para o usuário compartilhar com seus clientes, para que possam visitar o imóvel em realidade virtual.

Figura 04 - Diagrama de Atividade



Fonte: Elaborado pelos Autores.

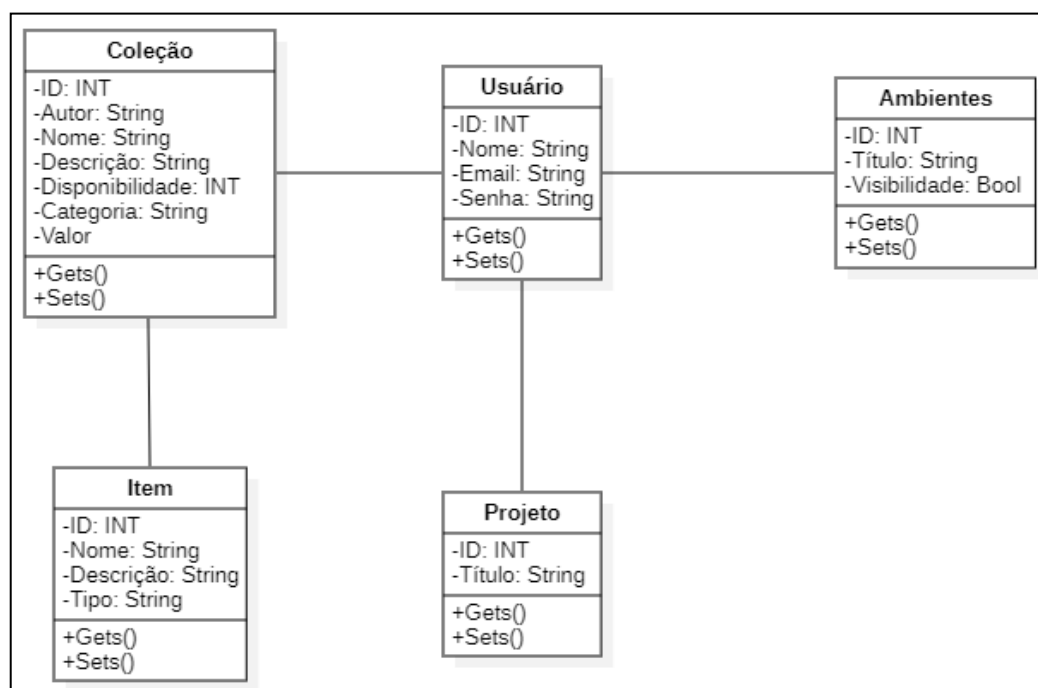


### 3.2.4 Diagrama de Classes

O diagrama de classes é fundamental para os processos de modelagem de um sistema, pois define todas as classes que o sistema precisa conter, além de ser base para a construção de outros diagramas.

Na Figura 05 está representado as principais classes do protótipo, como usuário, projeto e ambientes com seus respectivos atributos e métodos.

Figura 05 - Diagrama de Classes



Fonte: Elaborado pelos autores.

### 3.2.5 Diagrama MER

O Modelo de entidade relacionamento é uma representação gráfica utilizada para projetar banco de dados relacionais que ilustra como as entidades se relacionam entre si dentro do sistema, utilizando como base a relação de objetos reais, e sendo representado por meio de entidades e relacionamentos.

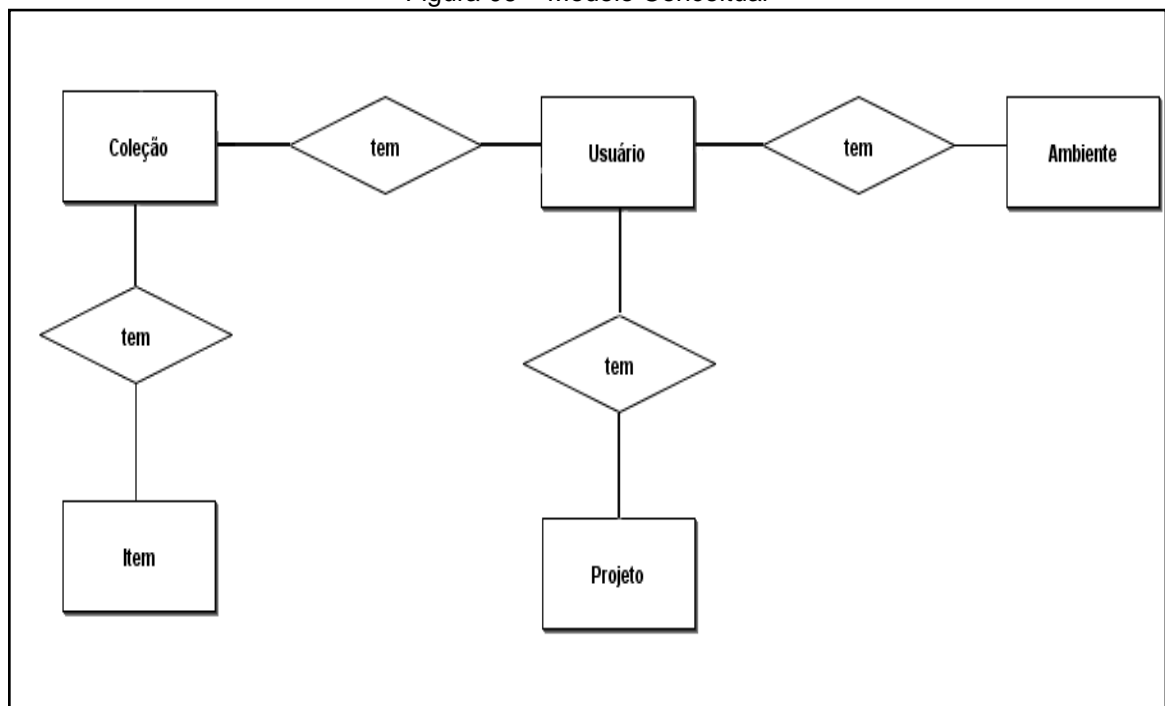
Para a criação dos diagramas desse protótipo foi utilizado a ferramenta brModelo, que oferece opções completas para a diagramação de um banco de dados.

### 3.2.5.1 Modelo Conceitual

A modelagem conceitual baseia-se no mais alto nível, pois seu foco é representar os aspectos do negócio sem se preocupar com a tecnologia subjacente.

Apresentado na Figura 06, o diagrama está representando cada entidade (tabela) do banco de dados.

Figura 06 – Modelo Conceitual

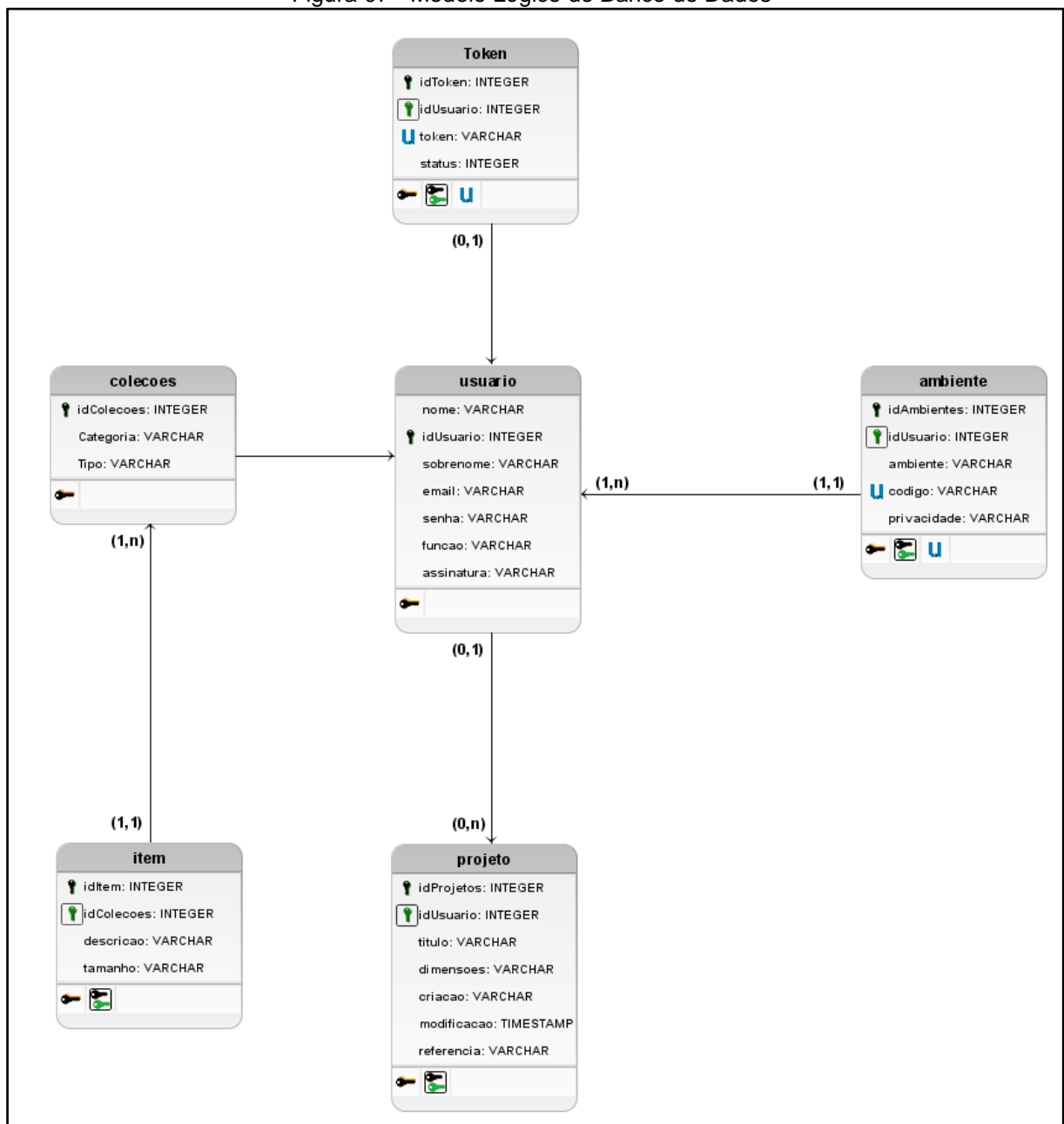


Fonte: Elaborado pelos autores.

### 3.2.5.2 Modelo Lógico

O Modelo Lógico de Dados é um modelo usado para descrever os dados de um banco de dados. Na figura 07 ele descreve os dados em termos de entidades, atributos e os relacionamentos entre as entidades do banco de dados.

Figura 07 - Modelo Lógico do Banco de Dados



Fonte: Elaborado pelos autores.

### 3.3 IMPLEMENTAÇÃO

Neste tópico será apresentado, detalhes e processos de implementação do protótipo, demonstrando as ferramentas, configurações e tecnologias utilizadas para o desenvolvimento do protótipo.

### 3.3.1 Técnicas e ferramentas utilizadas

Para o desenvolvimento da interface em realidade virtual, utilizou-se a ferramenta Unity 3D na versão 2021.2.19f1, associado a linguagem de programação C#. Para a interface Web, utilizou-se a linguagem de programação PHP, HTML e CSS. Para o gerenciamento da hospedagem do website e banco de dados utilizou-se o cPanel e o gerenciador de arquivos remoto utilizou-se o Filezilla.

Para a implementação e realização dos testes ao longo de todo processo de desenvolvimento, utilizou-se um computador com sistema operacional Windows 11, 16 GB de memória RAM e processador i5-10400F 2.9GHZ e uma placa de vídeo NVIDIA GeForce GTX 1650 e o dispositivo de realidade virtual Meta Quest 2.

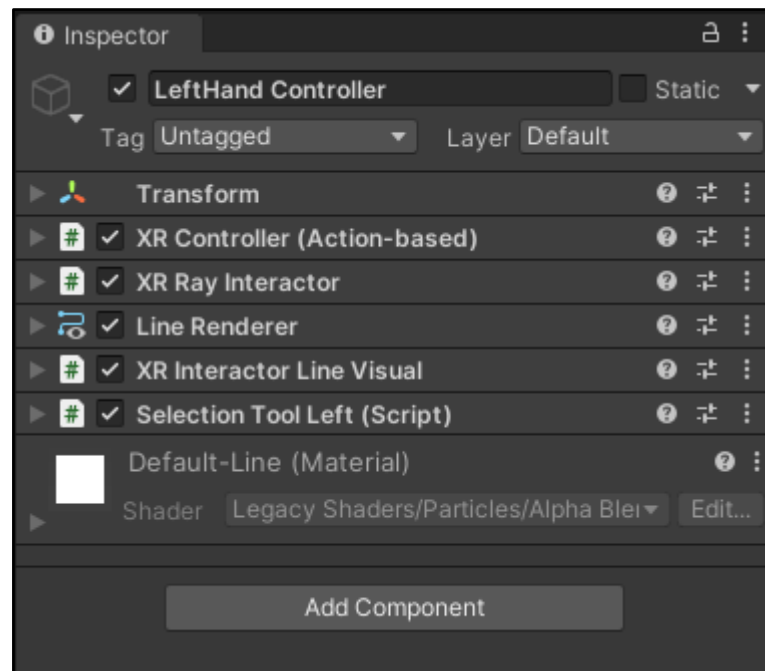
#### 3.3.1.1 Interface VR

Baseado nos resultados das pesquisas e levantamento de requisitos o Unity foi o framework mais adequado para implementação da interface em realidade virtual, pois além de um motor de jogos completo, possui uma série de plugins que permite sua adaptação para as mais diversas finalidades, além no ponto mais alto que é sua compatibilidade com a linguagem de programação C#, permitindo a personalização de funcionalidades e criação de ferramentas específicas que seriam necessárias.

##### 3.3.1.1.1 XR Interaction Toolkit

Parte fundamental para a execução deste projeto, é a comunicação eficiente entre a aplicação e o dispositivo Meta Quest 2, onde implementou-se o plugin XR Interaction Toolkit, que é um sistema de interação de alto nível baseado em componentes, que realiza esta comunicação. Apesar de permitir esta comunicação, a implementação deste plugin por si não resolve tudo, sendo necessário configurar e adequar através de código, a forma como esta comunicação deve ocorrer, sendo assim, foi necessário criar uma cena no Unity, que comportasse um conjunto de objetos chamados: XR Origin, Camera Offset, Main Camera, Controllers, XR Interaction Manager e Locomotion System, conforme FIGURA 08.

Figura 08 – Interface XR interaction Toolkit



Fonte: Elaborado pelos autores

Para gerenciar a forma como os controles se comportam, foram desenvolvidas duas classes, uma para representar o conjunto de botões e outra para representar o controle, desta forma foi possível desenvolver uma série de métodos universais que garantem o funcionamento de cada botão e cada controle individual e coletivo, como por exemplo descobrir os botões que estão sendo pressionados, quais estão trancados, quais estão com atraso de pressionamento ativo entre outros.

No QUADRO 04 é demonstrado parte do código responsável por verificar se os botões estão pressionados, e armazenar esta informação até nova chamada do método.

QUADRO 04 - código responsável por verificar se os botões estão pressionados.

```
public void SetButtonActive()
{
    joystick.TryGetFeatureValue(CommonUsages.menuButton, out jMenu.bActive);
    joystick.TryGetFeatureValue(CommonUsages.triggerButton, out jTrigger.bActive);
    joystick.TryGetFeatureValue(CommonUsages.gripButton, out jGrip.bActive);
    joystick.TryGetFeatureValue(CommonUsages.primaryButton, out jPrimaryButton.bActive);
    joystick.TryGetFeatureValue(CommonUsages.secondaryButton, out jSecondaryButton.bActive);
}
```

Fonte: Elaborado pelos autores.

Para garantir que pressionamentos duplos não interfiram na experiência do usuário, foi implementado um método de atraso no clique, onde uma vez que este seja chamado, a ação de pressionamento do botão deve ser bloqueada. No QUADRO 05, os métodos demonstrados aplicam o tempo de atraso padrão ao botão, e retornam o status do atraso do botão respectivamente.

QUADRO 05 - código responsável pelo método de atraso do botão.

```
public void SetDelay()
{
    this.bDelay = this.defaultDelay;
}

public bool DelayInactive()
{
    if (bDelay == 0)
    {
        return true;
    }
    return false;
}
```

Fonte: Elaborado pelos autores.

#### 3.3.1.1.2 Métodos de Execução

A execução de uma aplicação com interface de jogos utiliza um sistema de execução por taxa de quadros por segundo, isso significa que a cada quadro por segundo é executado uma determinada série de métodos. Para que os métodos não sejam executados todos ao mesmo tempo, conflitando com as ações realmente esperadas, a cena de edição foi separada em quatro modos de operação, sendo um modo: menu; e três modos de edição: construção, pintura e colocáveis.

Todos os modos são gerenciados por um controlador de cena, que informa qual o modo de operação deve ser executado. Cada modo de operação atua de maneira individual, possui suas próprias regras de negócio, definem o comportamento dos controles e interagem com os métodos correspondentes dos objetos em cena.

#### 3.3.1.1.3 Objetos de Cena

O que em programação chamamos de classes, para descrever um código que

representa uma base ou modelo, para criar novos objetos, no Unity temos sua versão visual, que chamamos de prefabs.

Um prefab é um agrupamento de objetos que podem ter as mais diversas características e formas, e quando instanciado na cena, torna-se um objeto único, levando consigo as características originais e todos os seus métodos.

Para assegurar que cada prefab criado para este protótipo se comporte como o esperado quando instanciado na cena, incluiu-se objetos de classes para determinar suas regras de negócio e métodos de acordo com suas especificidades e os modos de execução, garantindo que por exemplo um objeto de cena que deve se comportar como uma parede, não se comporte como uma porta, e nem que objeto que deve se comportar como uma mesa, receba as texturas e cores de um papel de parede.

#### 3.3.1.1.4 Métodos para Salvar e Carregar Projeto

Cada objeto instanciado e modificado, durante a execução da aplicação, é alocado na memória volátil, o que significa que durante esta execução tudo que foi instanciado, estará de alguma forma disponível para acesso, porém esta alocação é temporária, e após finalizar a execução, os objetos instanciados durante esta execução serão eliminados da memória, impossibilitando a recuperação posterior destes objetos, e para garantir que isso não aconteça, ou seja, tornar persistente os objetos produzidos durante a execução, é necessário selecionar na memória todos os dados que deverão ser persistentes, e convertê-los em algum formato armazenável, processo que chama-se serialização.

Conforme o QUADRO 06 o método implementado foi desenvolvido para o uso universal no protótipo, ou seja, sempre que for necessário salvar algum dado no armazenamento, este é salvo como um objeto, definido seu nome de arquivo, e o diretório de armazenamento, que após sua execução retorna o caminho do arquivo completo.

QUADRO 06 - código responsável pelo método de salvar.

```
public string SaveExecute(string filePathLocal, string fileNameComplete, object objToSave)
{
    DirectoryInfo di;
    string pathAndName;
    FileStream fileStream;
```

```

        BinaryFormatter bf = new BinaryFormatter();

        di = Directory.CreateDirectory(Application.persistentDataPath + checkPath(filePathLocal,
true));
        pathAndName = di + fileNameComplete;
        fileStream = File.Create(pathAndName);
        bf.Serialize(fileStream, objToSave);
        fileStream.Close();

        return pathAndName;
    }

```

Fonte: Elaborado pelos autores.

Uma vez serializado, é possível realizar o processo inverso, e carregar para a execução os dados armazenados como arquivo, porém serializar objetos de cena é uma ação proibida pela Unity, o que implica em ter métodos ainda mais específicos para serializar e salvar os objetos de cena, que façam a leitura dos objetos de cena que desejamos salvar, e converta seus valores em objetos de tipos serializáveis, possibilitando assim que estes objetos sejam de alguma forma salvos no armazenamento.

Se os objetos de cena têm seus valores convertidos para outros tipos de objeto, sabe-se que não são estes que serão carregados na cena. O que acontece aqui, é um novo instanciamento, que deverá ler os valores de cada objeto desserializado e utilizar como referência para instanciar novamente os prefabs como objetos de cena. Em resumo, novos objetos de cena são instanciados baseados nos valores dos objetos de cena que foram anteriormente serializados.

### 3.3.1.2 Interface Web

A interface web tem como objetivo facilitar o gerenciamento dos projetos e da conta pessoal do usuário, além de ser por meio dela que o usuário irá gerar um token de acesso a sua conta no dispositivo de realidade virtual. Para seu desenvolvimento utilizou-se a linguagem de marcação HTML, a linguagem de estilização CSS e a linguagem de programação PHP.



### 3.3.1.2.1 Gerenciamento dos Projetos

Por meio da interface web que a criação e gerenciamento dos projetos serão realizadas. Para isso, foi implementado uma tabela que lista todos os projetos criado pelo usuário, por meio dela será possível a criação e edição das informações dos projetos. Também será possível o carregamento da imagem de referência, com as extensões PNG ou JPG somente. A imagem será salva no servidor de arquivos e as informações dos projetos no banco de dados, ambas com o intuito de serem acessadas pela interface de realidade virtual.

### 3.3.1.2.2 API Google Sign-in

Para o acesso ao site VrPlaces será apresentado a tela de login onde o usuário terá duas opções, a primeira utilizando e-mail e senha e a segunda utilizando uma conta Google, em que o ao escolher esta opção, será acionada a API Google Sign-In. No Quadro 07 o código responsável por abrir a janela em que o usuário irá acessar com sua conta Google permitindo o protótipo se comunicar com os servidores do Google, para coletar as informações necessárias para o cadastro do usuário.

QUADRO 07 - código responsável pelo método de salvar.

```
window.onload = function() {  
  google.accounts.id.initialize({  
    client_id: "client_id",  
    callback: handleCredentialResponse  
  });
```

Fonte: Elaborado pelos autores.

Já no Quadro 08 é apresentado a função responsável por exibir e definir as características visuais do botão responsável por acionar a API.

QUADRO 08 - código responsável pelo método de salvar.

```
google.accounts.id.renderButton(  
  document.getElementById("buttonDiv"), {  
    theme: "outline",  
    size: "large",  
    type: "standard",  
    shape: "rectangular",  
    text: "$ {button.text}"  
  } // customization attributes  
);
```

Fonte: Elaborado pelos autores.

### 3.3.1.2.3 Token de Acesso

Para agilizar e facilitar o acesso à conta em dispositivos de realidade virtual, foi desenvolvido um sistema de token de 8 caracteres, que é gerado pela interface web quando o usuário solicita.

Cada token é único e só poderá ser gerado e atribuído a um usuário uma única vez. Para garantir isso, foi criada uma tabela no banco de dados que armazena o token e o status dele. O status representa a validade do token, caso o status seja “1” significa que é um token válido que irá conceder acesso a conta, caso seja “0” que não tem mais validade. Então na primeira vez em que o usuário gera um token ele é atribuído com o status de ativo, porém se deseja solicitar um novo, o sistema irá mudar no banco de dados o status do token velho para desativado “0”, e irá gerar um novo token com o status de ativo “1”, verificando primeiramente se ele nunca foi gerado anteriormente.

Para gerar um token foi necessário criar uma classe representando-o, nela foram criados métodos que geram aleatoriamente tokens e garantem que eles não nunca se repitam. O método que cria um token, gera 4 valores aleatórios de 0 a 9 representando 4 das dez listas que contém 26 elementos, cada um contendo 2 caracteres, podendo eles serem números ou letras. Após definidas aleatoriamente 4 das 10 listas, será novamente gerado um valor aleatório de 0 a 25, que representará as posições da lista, juntando cada elemento irá formar o token. Após gerado este token é verificado no banco de dados para garantir que ele é único, caso seja ele será salvo no banco de dados. No Quadro 09 está representado o trecho desta função escrita com a linguagem de programação PHP que gera o token.

QUADRO 09 - código responsável por gerar o token de acesso.

```

$pos1 = rand(0,9);
$pos2 = rand(0,9);
$pos3 = rand(0,9);
$pos4 = rand(0,9);
$token = $listaAle[$pos1][rand(0,25)] . $listaAle[$pos2][rand(0,25)]
. $listaAle[$pos3][rand(0,25)] . $listaAle[$pos4][rand(0,25)];

return $token;

```

Fonte: Elaborado pelos autores.

### 3.3.1.3 Banco de Dados

Para suprir as necessidades de armazenamento de dados do protótipo escolheu-se o banco de dados relacional MySQL, por sua compatibilidade com o PHP e C#, sua ampla documentação e por seu bom desempenho.

As conexões com o banco são realizadas tanto pela interface web, desenvolvida em PHP, quanto pela interface de realidade virtual desenvolvida em C#. Em ambas são realizadas conexões diretamente com o banco de dados.

No Quadro 10 é possível analisar um trecho exemplificando a operação de consulta ao banco de dados em C#, cuja função é listar todos os projetos criados pelo usuário. Já no Quadro 11, é possível analisar a função de cadastrar um novo usuário no banco de dados desenvolvida em PHP.

QUADRO 10 - código responsável consulta no banco de dados em C#.

```

public List<Projeto> GetProjects(string idusuario)
{
    MySqlConnection conn = new MySqlConnection(connStr);
    List<Projeto> result = new List<Projeto>();
    try
    {
        conn.Open();
        string sqlSelect = "SELECT * FROM projeto WHERE idusuario=" + idusuario + ",";

        MySqlCommand cmd = new MySqlCommand(sqlSelect, conn);
        MySqlDataReader reader = cmd.ExecuteReader();

        while (reader.Read())
        {
            string idprojeto = reader["idprojeto"].ToString();
            string idusuarioBD = reader["idusuario"].ToString();
            string titulo = reader["titulo"].ToString();
            string dimensao = reader["dimensao"].ToString();
            string criacao = reader["criacao"].ToString();
            string modificacao = reader["modificacao"].ToString();
            string referencia_tipo = reader["referencia_tipo"].ToString();

```

```

        result.Add(new Projeto(idprojeto, idusuarioBD, titulo, dimensao, criacao, modificacao,
referencia_tipo));
    }
    reader.Close();
    return result;
}

```

Fonte: Elaborado pelos autores.

QUADRO 10 - função de cadastrar um novo usuário no banco de dados desenvolvida em PHP.

```

public function setUsuario($nome, $sobrenome, $email, $senha, $funcao, $assinatura)
{
    $sql = "INSERT INTO `usuario` (`nome`, `sobrenome`, `email`, `senha`, `funcao`, `assinatura`)
VALUES (?, ?, ?, ?, ?, ?)";
    $stmt = $this->mysqli->prepare($sql);
    $stmt->bind_param("ssssss", $nome, $sobrenome, $email, $senha, $funcao, $assinatura);
    if ($stmt->execute() == TRUE) {
        return true;
    } else {
        return false;
    }
}

```

Fonte: Elaborado pelos autores.

#### 3.3.1.4 Serviços na Nuvem

Para a implementação do banco de dados, servidor de arquivos e do website durante o desenvolvimento deste protótipo, foram utilizados serviços de hospedados em um servidor na nuvem, contratado na empresa Unifique Telecomunicações que oferece aos seus clientes, o painel de controle cPanel, ferramenta qual permite gerenciar todos os aspectos da conta de hospedagem, como adicionar ou remover sites, ver o uso de recursos e muito mais.

##### 3.3.1.4.1 Hospedagem

Para hospedar o website seguiu-se 2 etapas, a primeira adquirir um domínio, neste caso foi adquirido através do Registro.br o “vrplaces.com.br” que é o endereço do site na internet e a segunda publicar o site e para isso, basta enviar os arquivos para o servidor usando um software FTP (File Transfer Protocol) que neste projeto utilizou-se o FileZilla.

## 4 CONCLUSÕES

A experiência neste trabalho demonstrou a viabilidade para desenvolver uma ferramenta de modelar ambientes para a realidade virtual na própria realidade virtual, pois foi possível provar os principais conceitos abordados por trás de toda a ideia.

A atividade mais destacada de todo o trabalho foi provavelmente a criação e modelagem de paredes, que foi possível graças ao uso de formas pré-moldadas e pré configuradas. Embora as peças no protótipo sejam básicas, as possibilidades experimentadas podem se aplicar a muitas outras formas.

Para uma experiência mais imersiva, o usuário pode visualizar uma imagem de referência do imóvel que deseja modelar, dentro da realidade virtual, precisamente no chão. Foi possível atingir esse resultado. Uma vez que o arquivo se encontra no dispositivo, ele é carregado como textura dentro da interface de realidade virtual.

Embora no protótipo nenhuma cena dedicada apenas à visita tenha sido criada, a execução desta ideia se baseia na mesma que a cena de edição: carregar um arquivo de projeto na memória e instanciar na cena. Uma vez carregado, como na cena de edição, é só se locomover pelo ambiente e desfrutar a visita.

Como a compatibilidade com outros dispositivos é uma característica de configuração do Unity, foram realizados testes através de simuladores na própria engine, que apresentou compatibilidade com vários dispositivos para pelo menos a visualização.

Outro objetivo levantado durante a pesquisa, é disponibilizar o serviço a preços menores que o praticado no mercado. Como não foi possível mensurar o valor final do serviço, qualquer afirmação seria imprecisa, ficando em aberto este objetivo.

No que diz respeito às possibilidades comerciais do protótipo, é necessária uma pesquisa mais específica sobre o assunto, com tudo, o desenvolvimento se mostrou viável e tangível, do ponto de vista técnico e econômico.

## REFERÊNCIAS

ALMEIDA, Israel Bastos de Souza; FRANÇA, Ivo Chaves de. Ferramenta de captura de comandos DML em aplicações web com PHP e MySQL. Salvador – BA: UNIVERSIDADE CATÓLICA DO SALVADOR, 2007.

GOTARDO, Reginaldo. Linguagem de programação. Rio de Janeiro: Sesi, p. 34, 2015. Interamericana do Brasil Ltda., 2006.

L., Andrei. Tutorial cPanel Completo: Todos os Recursos do Painel de Hospedagem. 2022. Disponível em: <https://www.hostinger.com.br/tutoriais/tutorial-cpanel-completo>. Acesso em: 12 jun. 2022.

LATTA, J.N. & OBERG, D.J.: "A conceptual virtual reality model", IEEE Computer Graphics & Applications, pp. 23-29, Jan, 1994.

META Quest 2. 2022. Disponível em: <https://store.facebook.com/quest/products/quest-2>. Acesso em: 16 jul. 2022.

PRESSMAN, Roger S. Engenharia de Software. 6. ed. São Paulo: McGraw-Hill

VIANNA, Maurício et al. Design Thinking: Inovação em negócios. Rio de Janeiro: Mjv Press, 2014. 195 p.

ZOMPERO, Eric. Explicando o Metaverso: Simples e Direto (Notas de Aula). São Paulo: [S.N.], 2022. 59 p.

## **ANEXO A – Questionário utilizado para o levantamento de requisitos**

Olá, somos uma equipe do curso técnico em desenvolvimento de software do CEDUP - Timbó e estamos recolhendo dados que vão auxiliar o desenvolvimento do nosso TCC (trabalho de conclusão de curso).

Não se preocupe, pois, a pesquisa é curta e anônima e nenhuma resposta será relacionada com a pessoa que respondeu ao questionário.

Atenciosamente, Gustavo Henrique Campestrini e Michel Nienow de Barros

### **Modelador de Imóveis 3D**

O projeto é um protótipo de aplicativo modelador de ambientes para realidade virtual, que através de um dispositivo de realidade virtual permite que o usuário construa uma imóvel em 3D, utilizando uma planta baixa como referência, mobílie, decore, e disponibilize para ser visitado na realidade virtual.

- 1) Você prefere realizar o cadastro como?
  - ☐ Em um site
  - ☐ No dispositivo de realidade virtual
  
- 2) Para se cadastrar na plataforma você prefere qual métodos de cadastro?
  - ☐ Facebook
  - ☐ Google
  - ☐ E-mail
  
- 3) Para usar uma planta baixa de um imóvel como referência dentro da realidade virtual para modelá-lo, você prefere importar esta imagem através de qual método?
  - ☐ Website
  - ☐ Aplicativo
  - ☐ Envio como anexo por e-mail
  
- 4) Ao entrar pela primeira vez, você prefere qual tipo de tutorial?

- ☐ Tutorial em vídeo
- ☐ Tutorial guiado/prático

5) Ao usar a ferramenta no ambiente de realidade virtual você acha importante que tenha a opção do modo escuro?

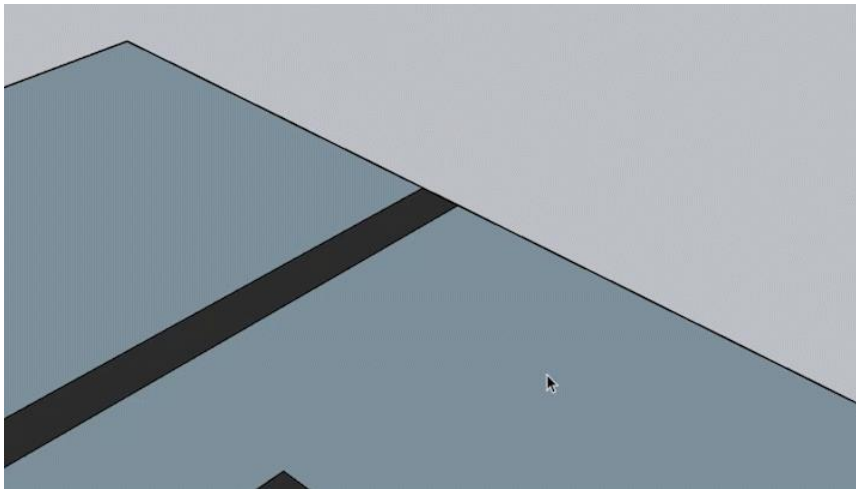
- ☐ Sim
- ☐ Não

6) Para desenhar as paredes e colocar os objetos, você acha que deveria existir um sistema de alinhamento automático?

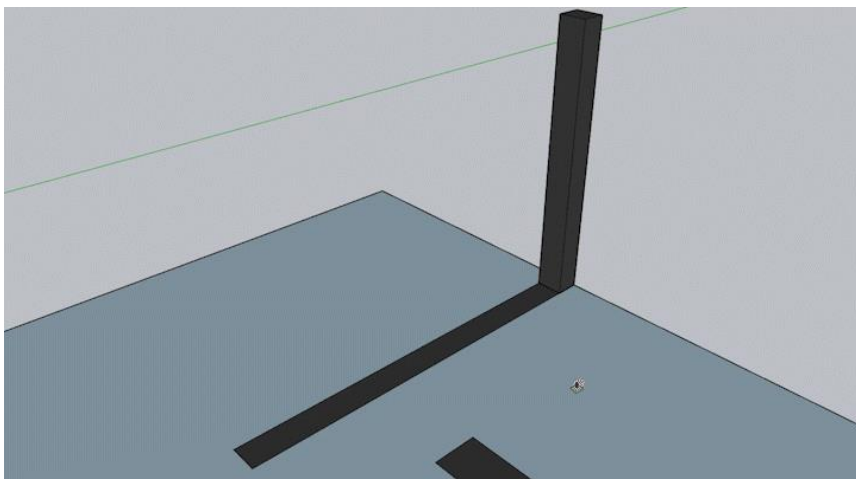
- ☐ Sim
- ☐ Não

7) Para desenhar as paredes, qual método parece mais fácil e intuitivo?

- ☐ Método de 2 pontos: definir começo da parede e final da parede



- ☐ Método extrusão: puxar a parede do começo até seu ponto final





8) Para utilizar esta ferramenta você prefere:

( ) Pagar um plano mensal que contemple uma série de pacotes (mobílias, texturas, itens).

( ) Pagar uma ativação única e comprar os pacotes avulsos (mobílias, texturas, itens).

9) Sugestões

Campo de texto para resposta longa