

AQUÁRIO VIRTUAL: MULTIPLAYER E REALIDADE VIRTUAL

Matheus Waltrich da Silva, Dalton Solano dos Reis – Orientador

Curso de Bacharel em Ciência da Computação

Departamento de Sistemas e Computação

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

matheuswaltrich@gmail.com, dalton@furb.br

Resumo: Este artigo apresenta o desenvolvimento do projeto de extensão do projeto Aquário Virtual (LOSADA, 2019), adicionando as funcionalidades de multiplayer e realidade virtual. Este projeto foi desenvolvimento na plataforma Unity3D, utilizando a biblioteca mirror, para o multiplayer. Os objetivos de projeto foram atingidos, sendo que o multiplayer e a realidade virtual funcionam corretamente em conjunto. Com um dispositivo executando o ambiente do aquário sendo o host e os demais dispositivos executando o modo de realidade virtual como clientes. Neste projeto, também foi realizado a sincronização de uma câmera do dispositivo host para os dispositivos clientes, sendo necessário fazer conversão da imagem do host em um array de bytes para sincronização deste array para os clientes. Extensões também são possíveis, uma vez que foi desenvolvido a funcionalidade multiplayer e realidade virtual, sendo possível incrementar novas funcionalidades e melhorias nestas duas funcionalidades.

Palavras-chave: Aquário virtual. Multiplayer. Realidade Virtual.

1 INTRODUÇÃO

Para Almeida (2004, p.4) a inserção das Tecnologias da Informação e Comunicação (TICs) na educação oportuniza romper com as paredes da sala de aula e da escola, integrando-a à comunidade que a cerca, à sociedade da informação e a outros espaços produtores de conhecimento. Tomé (2013) salienta que as tecnologias são imprescindíveis no aprendizado infantil, pois desenvolvem diferenciadas e ricas estratégias, fazendo com que os educandos aprendam de forma lúdica, dinâmica e prazerosa, respeitando suas limitações e individualidades. Um destes recursos da tecnologia da informação que vem se tornando cada vez mais presente na sala de aula é a realidade virtual. Conforme Queiroz (2017) “A tecnologia de Realidade Virtual (RV) vem se tornando bastante popular e acessível sendo natural, dado o seu potencial como ferramenta didática, sua crescente introdução nas salas de aula tradicionais e virtuais.”

Uma forma de se utilizar a realidade virtual é através de simulações, que é bastante utilizada no âmbito educacional. Também é possível utilizar a realidade virtual com Interface de Usuário Tangível (IUT - do Inglês, TUI - Tangible User Interface), como pode-se observar no trabalho Aquário Virtual: Simulador de Ecossistema (LOSADA, 2019). Neste trabalho, o usuário pode utilizar sensores para realizar a alteração de parâmetros do ecossistema do aquário, como temperatura, luminosidade e a alimentação dos peixes. O comportamento dos peixes do simulador do aquário é realizado de forma autônoma respeitando algumas regras básicas da simulação do ecossistema.

Neste trabalho, se usa a realidade virtual, que é uma tecnologia de interface, utilizada para realizar uma imersão maior para o usuário no ambiente virtual. Para isso será utilizado o Head Mounted Display (HMD), que é um suporte para a cabeça, na qual vai inserido o smartphone, com executa a simulação. Para a utilização HMD na Unity, tem-se o Cardboard. O Cardboard é um aplicativo de realidade virtual, desenvolvido pela Google, que pode ser utilizado com um suporte de cabeça para o smartphone.

Este trabalho também irá utilizar a função de multiplayer, fazendo com que um smartphone android irá executar o aquário como host e outros dispositivos poderão ser conectar neste host e compartilhar o mesmo ambiente de integração com a realidade virtual. Para o desenvolvimento do multiplayer, foi utilizada a biblioteca Mirror, que é uma biblioteca gratuita, para o Unity, para desenvolvimento de jogos multiplayer.

A presente proposta tem como objetivo estender o projeto Aquário Virtual: Simulador de Ecossistema (LOSADA, 2019) para que o usuário tenha a percepção do ecossistema pela visão do peixe. Os objetivos específicos são: a) criar uma extensão em realidade virtual com a utilização do cardboard; b) tornar o simulador multijogador; c) utilizar o conceito de avatar para o usuário ter a visão do peixe de dentro do aquário. Será utilizada a tecnologia de cardboard, uma plataforma de realidade virtual desenvolvida pela Google que permite acoplar um smartphone na cabeça do usuário. Fazendo com que o usuário tenha uma percepção maior de como as ações no ambiente impactam na vida dos seres do aquário. Terá disponível em conjunto a tecnologia de cardboard a opção de multiplayer, possibilitando que vários usuários possam utilizar o mesmo aquário.

Mirror

cabeça, no qual se insere um smartphone para fechar todo o campo de visão do usuário com o resultado da visualização da simulação.

Uma das formas de usar o HMD é com o Cardboard, que é um aplicativo de realidade virtual desenvolvido pela Google para este fim.

também usou a função

multiplayer, e assim executar o aquário virtual como

host. E ter outros aquários virtuais conectados neste host, e assim compartilhar o mesmo ambiente virtual usando o conceito de realidade virtual.

multiplayer se utilizou a biblioteca

gratuita para o Unity que permite o desenvolvimento de

Desta forma, este trabalho tem como

Itálico.

usuário possa ter uma

E assim, ter disponível em conjunto com a tecnologia de cardboard a opção de multiplayer

2 FUNDAMENTAÇÃO TEÓRICA

Neste capítulo serão apresentadas as seguintes fundamentações: realidade virtual, multiplayer, o trabalho anterior e os trabalhos correlatos.

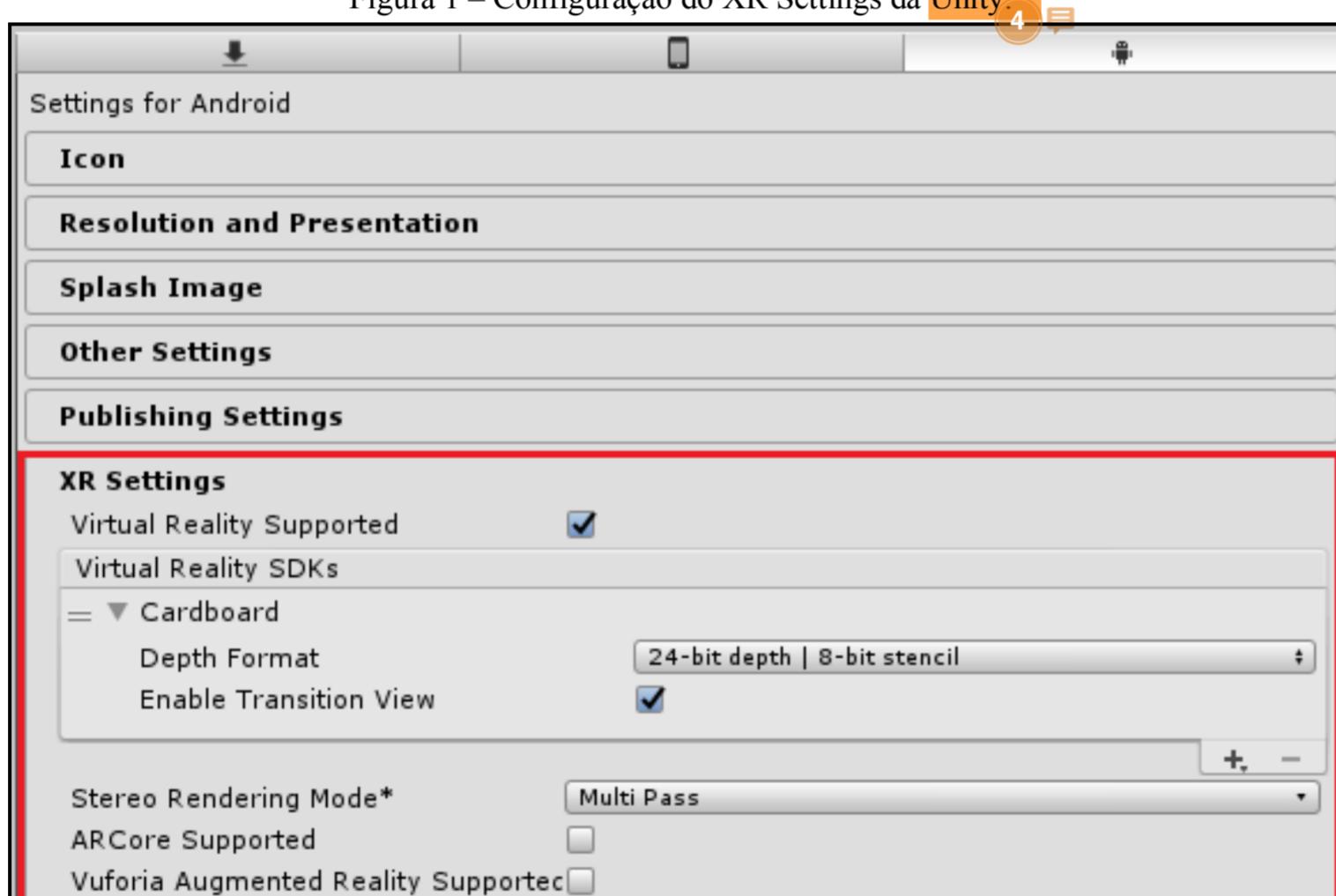
2.1 REALIDADE VIRTUAL

A realidade virtual é uma tecnologia de interface que engana os sentidos do usuário, através de um ambiente virtual criado induzindo efeitos visuais e sonoros. Conforme Queiroz (2017) o potencial de aplicação da realidade virtual na educação vem sendo exaltado desde sua pré-história quando nem mesmo tinha essa denominação. A realidade virtual possibilita ao usuário o sentimento de estar presente no ambiente virtual. Para Baierle e Gluz (2017) a evolução que levou ao surgimento de tecnologias como a Realidade Virtual, leva a cada dia mais pessoas em diferentes áreas de estudo a experimentar simulações tridimensionais devido a sua semelhança física com o mundo real. Conforme Sgobbi et. al (2014), a utilização de tecnologias imersivas permite uma presença virtual na situação de ensino-aprendizagem, garantindo um maior aproveitamento nas atividades propostas.

Uma das formas de utilizar a realidade virtual é através do Head Mounted Display (HMD). O HMD é um dispositivo utilizado em um suporte na cabeça ou como parte de um capacete, que possui um pequeno display óptico em frente aos olhos do usuário (Monocular) ou de cada olho (Binocular). Uma forma de criar uma aplicação no Unity utilizando o HMD é com o Cardboard, conforme Google (2019).

Cardboard é um aplicativo de realidade virtual, desenvolvido pela Google, que pode ser utilizado com um suporte de cabeça para o smartphone, para que o usuário tenha a visão do ambiente virtual de forma imersiva. Nele o usuário pode montar o seu próprio visor com base nas instruções contidas no site da Google ou comprar um visor pronto. Uma das formas de utilizar realidade virtual é através do motor de jogos Unity. O Unity possui funções nativas para configurar as propriedades do XRSettings (Figura 1, Virtual Reality Supported), dentro das propriedades do projeto. E com scripts permite usar realidade virtual nas cenas gráficas definidas no projeto.

Figura 1 – Configuração do XR Settings da Unity.



Fonte: elaborado pelo autor.

Conforme Baierle e Gluz (2017), a RV imersiva tem o potencial de ser uma tecnologia revolucionária, talvez até disruptiva. Porém esse potencial depende de uma série de fatores envolvendo facilidade de uso, custo acessível e uma aplicação popularizadora (*killer application*). Isto é ainda mais importante nas aplicações educacionais, fortemente sensíveis a estes três fatores.

A Realidade Virtual, permite que o aluno ⁶ experimente atividades de forma prática ⁷ sem sair da sala de aula. Conforme Silva e Pio (2017), A tecnologia de realidade virtual tem favorecido novas formas de ensinar. A imersão do aluno em um ambiente virtual proporciona a ele uma experiência inovadora e interativa, com sensações reais de vivência na interação com um cenário virtual, mas próximo da realidade.

Citação não encontrada nas referências bibliográficas.

Itálico.

Itálico.

Remover ponto final.

As legendas das figuras não tem pontuação final (ABNT).

Itálico

aluno

prática

(2017), a tecnologia

(UNITY, 2019).

Itálico.

uma Application Programming Interface (API) de

com que o

tempo. Esta

Itálico.

nesta

Itálico.

Itálico.

Itálico.

Itálico.

Não itálico.

Fonte courier.

Não itálico.

Fonte courier.

Itálico.

funcionalidades (Figura 2).

Remover ponto final.

Mirror (2020).

ter

Itálico.

Fonte courier.

Fonte courier.

Itálico.

Itálico.

Fonte courier.

Itálico.

Itálico.

Itálico.

Itálico.

Protocolo de Controle de Transmissão (Transmission Control Protocol - TCP).

2.2 MULTIPLAYER

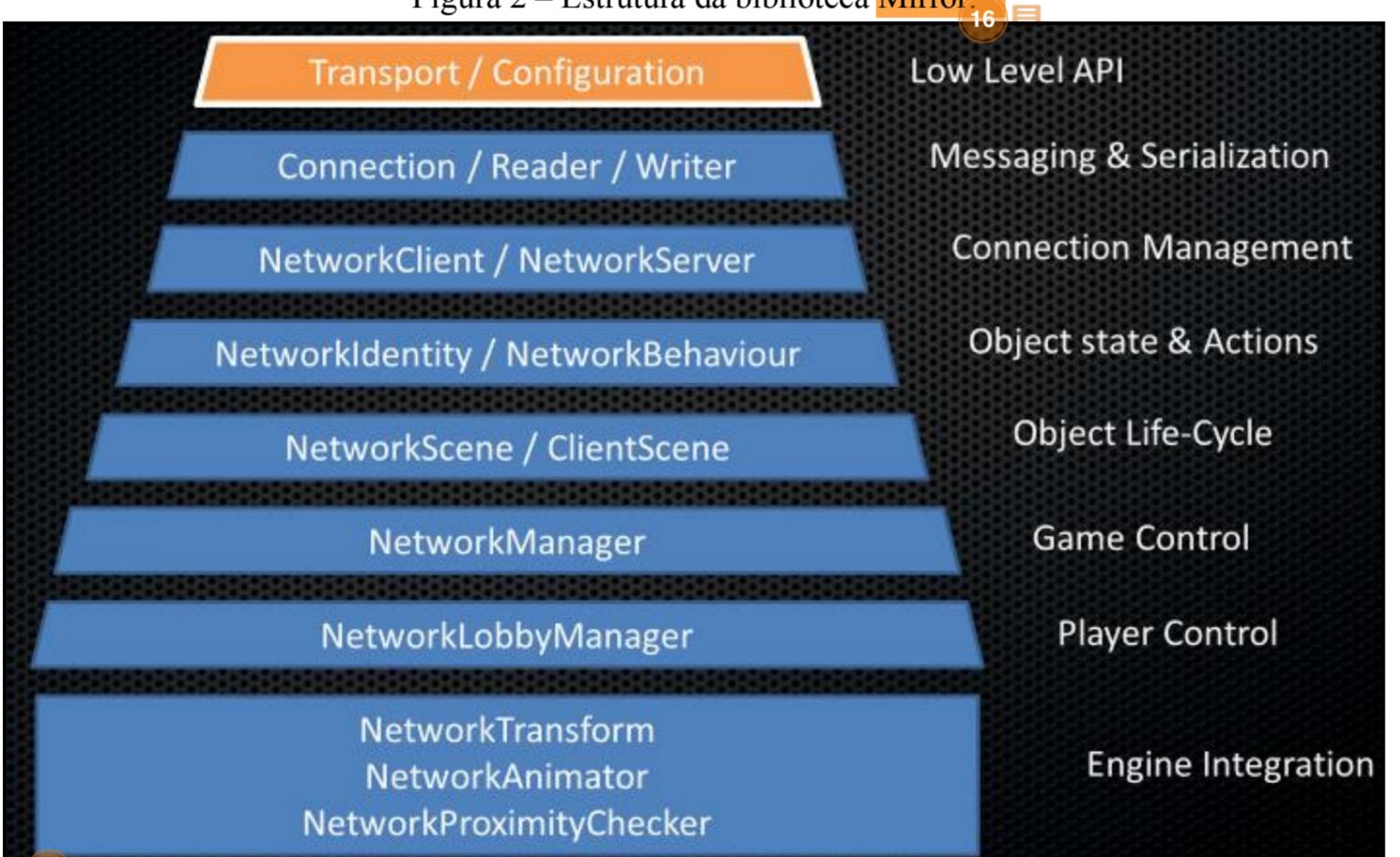
A utilização multiplayer é permitir que vários usuários participem simultaneamente de uma mesma partida em um mesmo ambiente virtual simulado. Segundo Luis et al (2016) a utilização do recurso de multiplayer pode auxiliar a interação entre os alunos e assim facilitar a relação deles com as atividades. Possibilita assim uma forma diferente de uso onde todos os alunos possam participar simultaneamente. No motor de jogos Unity é possível desenvolver jogos multiplayer, com recursos que auxiliam o desenvolvimento do jogo. O Unity permite desenvolver jogos com baixa latência, prevenção de trapaças e criação de partidas multiplayer (UNITY (2019)).

Conforme Unity (2019), o motor de jogos Unity também fornece exemplos de jogos desenvolvidos para utilização multiplayer, para auxiliar o desenvolvedor no processo de implementação do jogo. Possui recursos para hospedagem de servidores para jogos online, recurso para comunicação de bate-papo tanto por voz quanto por texto e criação de partidas em grupo. Auxiliando o desenvolvimento de jogos multiplayer na ferramenta.

Para o desenvolvimento da funcionalidade multiplayer deste trabalho, foi utilizado uma biblioteca gratuita do Unity, chamada Mirror. Esta biblioteca utiliza os próprios recursos multiplayer da Unity, mas torna a utilização destes recursos mais amigável para o desenvolvedor. Mirror é uma API de alto nível para vários jogadores multiplayer. É um sistema para criar recursos para vários jogadores na plataforma Unity. O Mirror, permite fazer com o server seja também um cliente o tornando um host que é um cliente e server ao mesmo tempo, esta arquitetura é a que foi utilizada no projeto, tendo um dispositivo executando o aquário como host e clientes se conectando neste host utilizando a realidade virtual.

No desenvolvimento deste trabalho foram utilizados os scripts de sincronização de posição, com o nome de NetworkTransform e de animação com o nome NetworkAnimator da biblioteca Mirror. Estes scripts tem o objeto de sincronizar a localização (x,y,z) e a animação que está sendo executado pelo objeto que foi definido como player entre os clientes do host. A biblioteca Mirror é construída a partir de uma série de camadas que adicionam funcionalidade, que podemos observar na Figura 2 abaixo.

Figura 2 – Estrutura da biblioteca Mirror



Fonte: Mirror. (2020).

Para utilizar a biblioteca Mirror é necessário ter um objeto com o script NetworkManager, que a partir deste é chamado o NetworkServer, para instanciar um server ou host e o NetworkCliente, para instanciar um cliente deste servidor/host. O Mirror possui vários tipos de comunicação de rede para realizar a sincronização de informações entre o servidor/host e os clientes. Neste trabalho foi utilizada a comunicação via TCP.

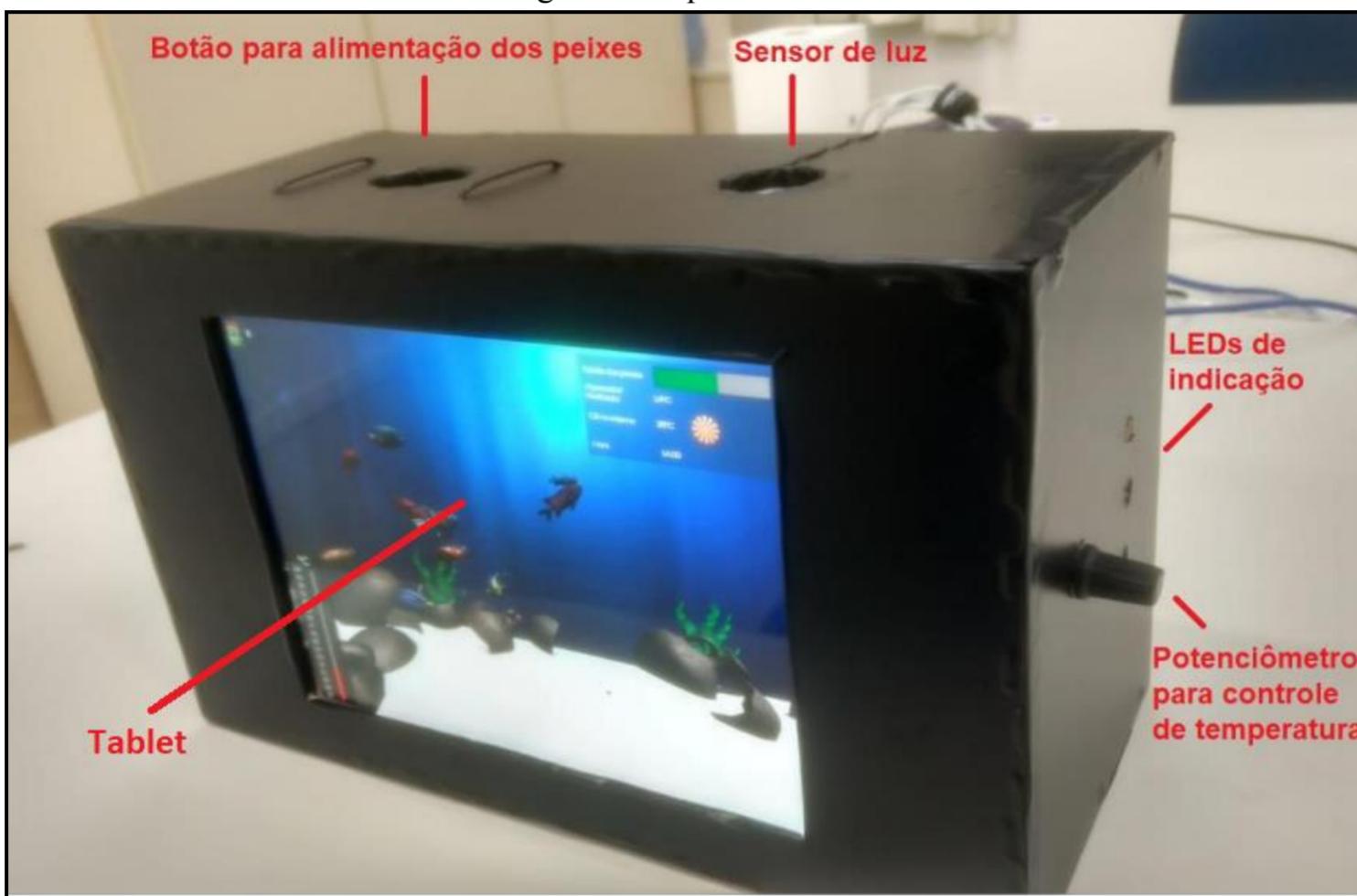
2.3 VERSÃO ANTERIOR DO SISTEMA

Na versão anterior do sistema tem-se o aquário virtual desenvolvido na plataforma Unity3D utilizando a linguagem de programação C# para implementação de comportamentos (LOSADA, 2019). O simulador foi desenvolvido utilizando o asset AIFishes (UNITY, 2019) como base, obtendo assim os peixes em 3D com alguns comportamentos padrões. Também utiliza o recurso de Interface de Usuário Tangível (IUT), para interação com o aquário virtual.

Na Figura 3 tem-se a caixa desenvolvida para portar o *tablet* e o IUT, consta um botão para realizar a alimentação dos peixes, um sensor para verificar a luz do ambiente externo,  leds de indicação e um potenciômetro para controle da temperatura.

1
Itálico.

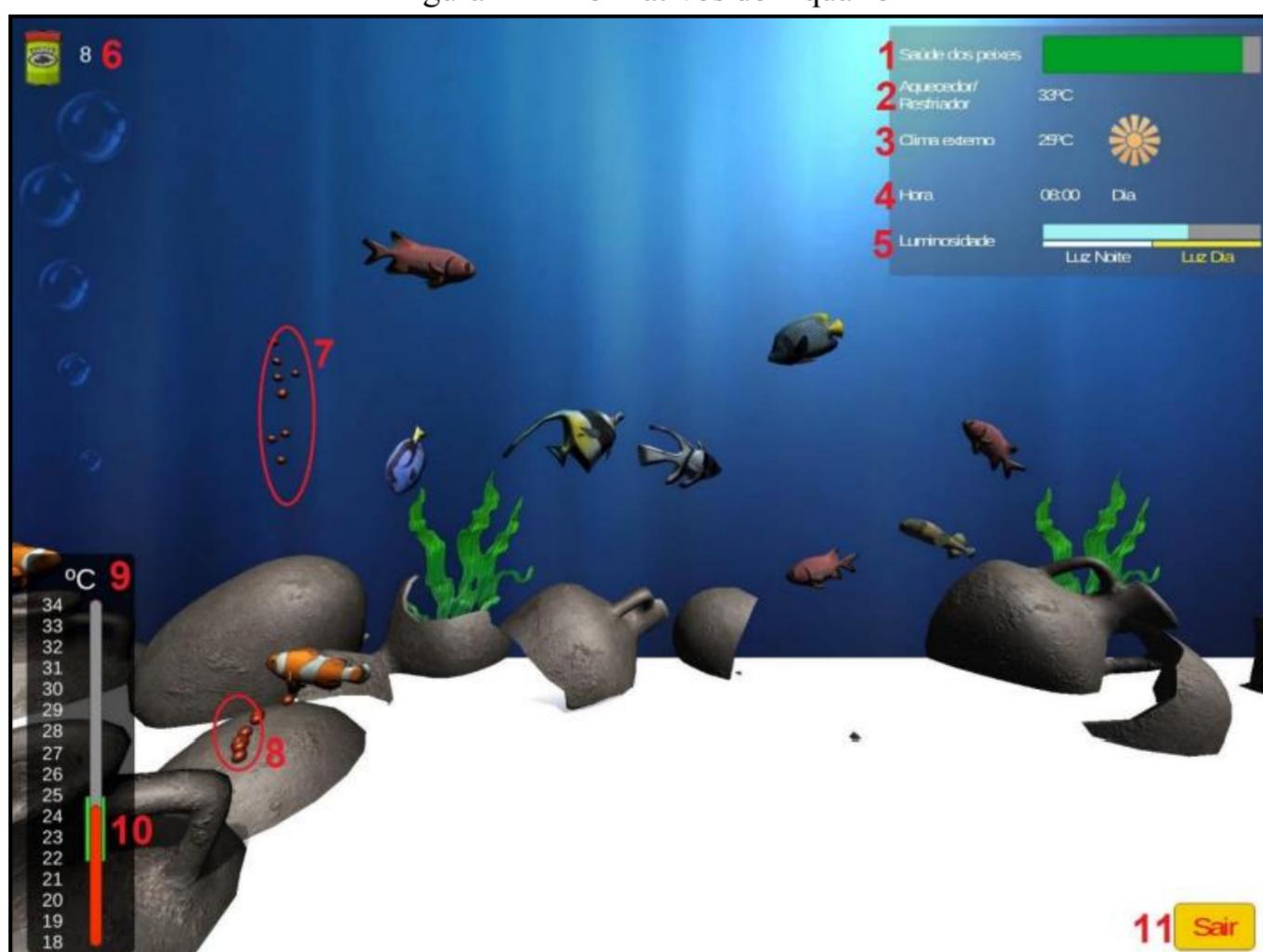
Figura 3 - Aquário Virtual



Fonte: Losada (2019).

No simulador tem-se indicadores para informar a situação da saúde dos peixes do ambiente do aquário virtual, consta a saúde dos peixes (Figura 1-1), aquecedor/resfriador (Figura 1-2), clima externo (Figura 1-3), hora (Figura 1-4), luminosidade (Figura 1-5), comidas (Figura 1-6) e termômetro (Figura 1-9 e Figura 1-10). Na imagem também temos uma comida que foi inserida no aquário caindo e outra que já está no fundo do aquário (Figura 1-7 e Figura 1-8). Esses indicadores são utilizados para verificar a saúde dos peixes e o ambiente do aquário virtual, como pode-se observar na Figura 4. Por fim temos o botão para sair (Figura 1-11).

Figura 4 – Informativos do Aquário



Fonte: Losada (2019).

Quadro 3 – Sanches et al. - Vrcircuit: Realidade Virtual Aplicada ao Ensino de Circuitos Elétricos

Referência	Sanches et al. (2017).
Objetivos	Simular um ambiente de uma residência com problemas elétricos a serem resolvidos pelo usuário para o aprendizado de elétrica.
Principais funcionalidades	Utilizada a realidade virtual para o usuário poder visualizar o ambiente virtual da residência com problemas elétricos a serem resolvidos e com isso ter uma aprendizagem prática sobre o assunto ensinado.
Ferramentas de desenvolvimento	Para o desenvolvimento utilizada a ferramenta Unity 3D na versão 5.5.2.
Resultados e conclusões	Os objetivos do trabalho foram atingidos. Conforme Baierle e Gluz (2017), após testes realizados com alunos jogando o VRCircuit foi verificado que o mesmo pode vir a colaborar fortemente no processo de ensino/aprendizagem.

Fonte: elaborado pelo autor.

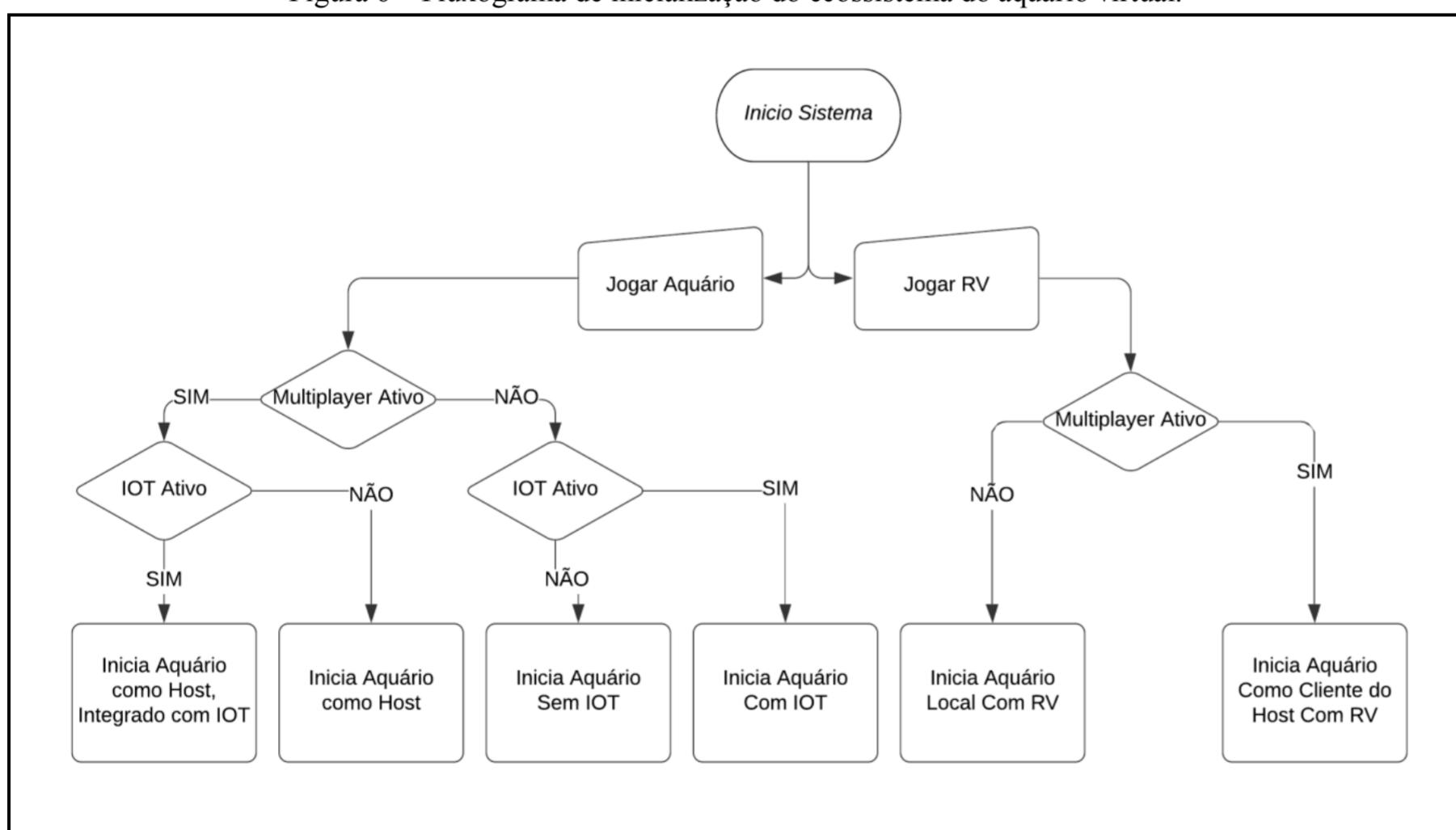
3 DESCRIÇÃO

Neste capítulo serão apresentadas algumas informações sobre o desenvolvimento do trabalho. Este projeto foi dividido em duas partes, primeiro a realidade virtual e depois o multiplayer.

Em relação ao desenvolvimento do trabalho algumas funcionalidades e regras do ambiente virtual ainda continuam com as regras definidas pelo projeto do Losada (2019), como se pode observar na Figura 3. Foi alterado o ciclo de chamada do aquário, para compatibilizar as funcionalidades já existentes do aquário, para ativar ou desativar a integração com o IOT e a utilização do multiplayer, sendo adicionado também um botão para executar o aquário em modo VR. Na Figura 6, abaixo podemos observar o novo fluxo de chamada do aquário no projeto, de forma que o aquário é ativo de formas diferentes conforme é configurado na tela de configuração do sistema.

Figura 6 – Fluxograma de inicialização do ecossistema do aquário virtual.

Figura 6 se pode observar



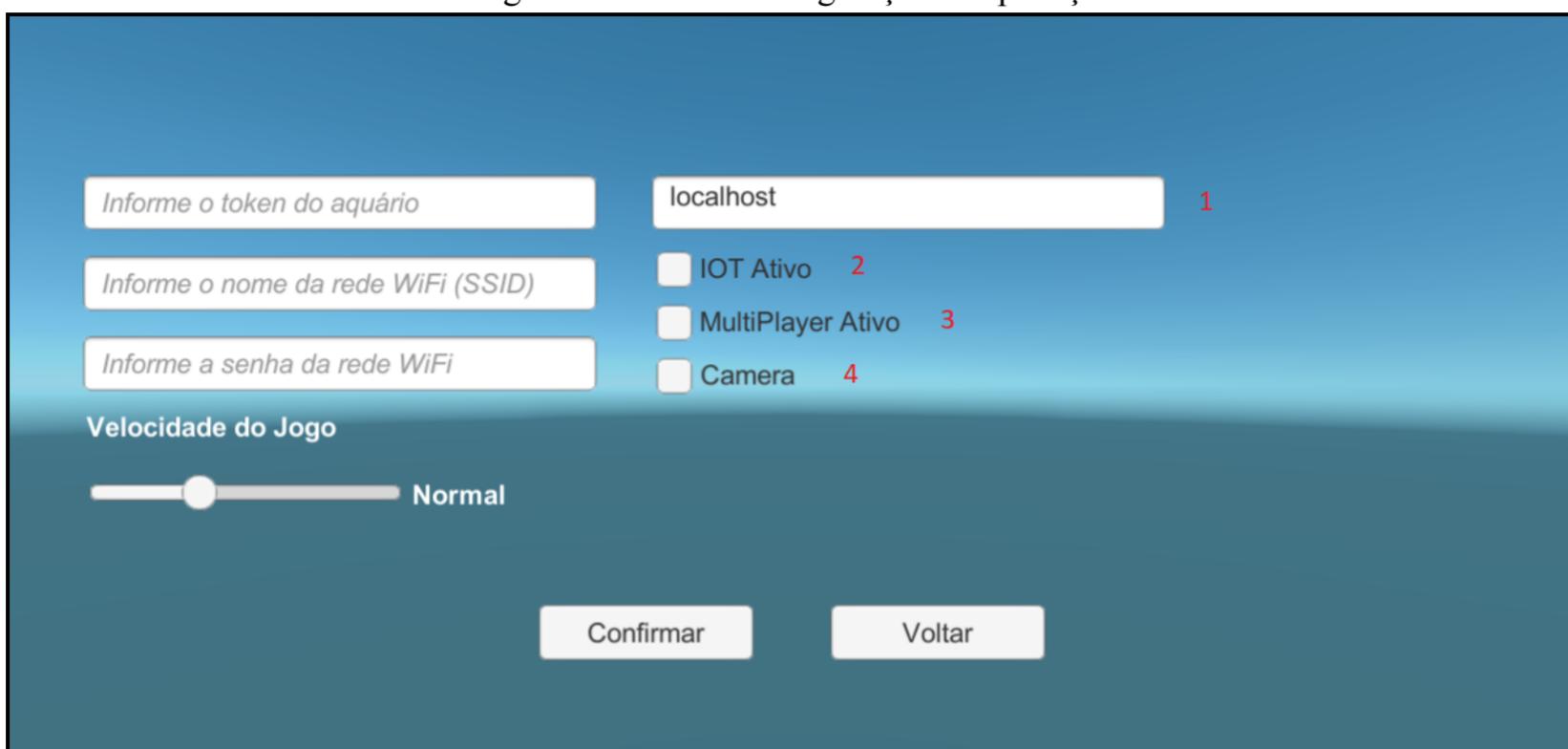
Fonte: Elaborado pelo autor.

Para poder compatibilizar o projeto anterior com as novas funcionalidades foram adicionadas algumas novas configurações. Na Figura 7 se pode observar as novas configurações que foram adicionadas ao projeto. O item 1 da figura é um campo que foi adicionado para configurar o ip do host dispositivo que vai executar o aquário. O item 2 da figura é uma configuração para ativar ou desativar a integração com o IOT, possibilitando ao usuário utilizar a aplicação, mesmo sem o dispositivo IOT. O item 3 da imagem é uma configuração para ativar ou desativar o multiplayer, possibilitando ao usuário utilizar a aplicação, sem o multiplayer. O item 4 da imagem, é uma configuração para ativar a câmera de desenvolvimento, permitindo inverter o lado da câmera que a câmera é apresentada. Esta configuração foi adicionada para facilitar os testes durante o desenvolvimento.

Itálico.

host do dispositivo

Figura 7 – Tela de configuração da aplicação



Fonte: Elaborado pelo autor.

Na tela inicial da aplicação se adicionou um novo botão chamado **Jogar RV** e alterado o nome do botão original, para **Jogar Aquário** (Figura 8). Ao se pressionar o botão **Jogar Aquário** com a opção multiplayer desmarcada, se inicia a mesma aplicação do projeto original, ou seja, sem o multiplayer. Já ao se pressionar o botão **Jogar RV** com a opção multiplayer desmarcada, se inicia a aplicação no modo de realidade virtual, sem o multiplayer. Com a opção multiplayer ativa, o botão **Jogar Aquário** inicia um host, já com um modelo do peixe na cena virtual do aquário. E por fim, tem o botão **Jogar RV**, que inicia um cliente no IP que está configurado no campo server. Este cliente entra no modo de realidade virtual para ser usado com o **cardboard**.

Figura 8 – Tela inicial da Aplicação

1 cardboard.



Fonte: Elaborado pelo autor.

Para o desenvolvimento da realidade virtual no projeto foi adicionado uma câmera no peixe que o segue e foi convertida **esta** câmera para que seja apresentada utilizando a funcionalidade de realidade virtual. Foi realizada a criação de um **script** (Quadro 4) para que a câmera acompanhasse o peixe durante o seu percurso no aquário virtual. Para ativar a câmera em modo de realidade virtual foi utilizada a propriedade **XRSettings**, da Unity que controla a ativação de desativação da realidade virtual.

2 Itálico.

Quadro 4 – Script para que a câmera acompanhe o peixe

```
public GameObject alvo;
public GameObject cameraposicao;

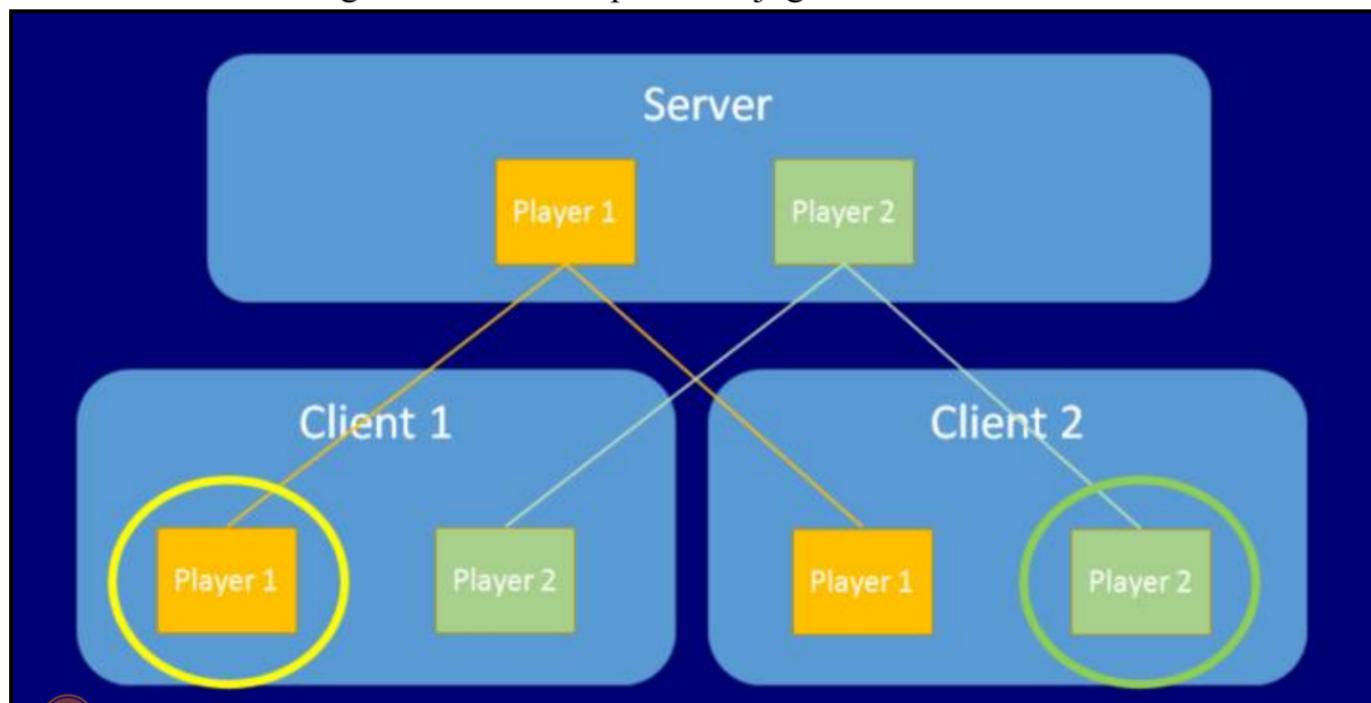
// Update is called once per frame
void Update()
{
    transform.LookAt(alvo.transform);
    transform.position = cameraposicao.transform.position;
}
```

Fonte: Elaborado pelo autor.

Já a biblioteca Mirror realiza o espelhamento dos *players* entre os vários clientes. Sendo controlado pelo servidor host.¹ Na Figura 9 se observa como esta arquitetura funciona com o server espelhando os dois *players* nos clientes.

Itálico.

Figura 9 – Server espelhando jogadores nos clientes

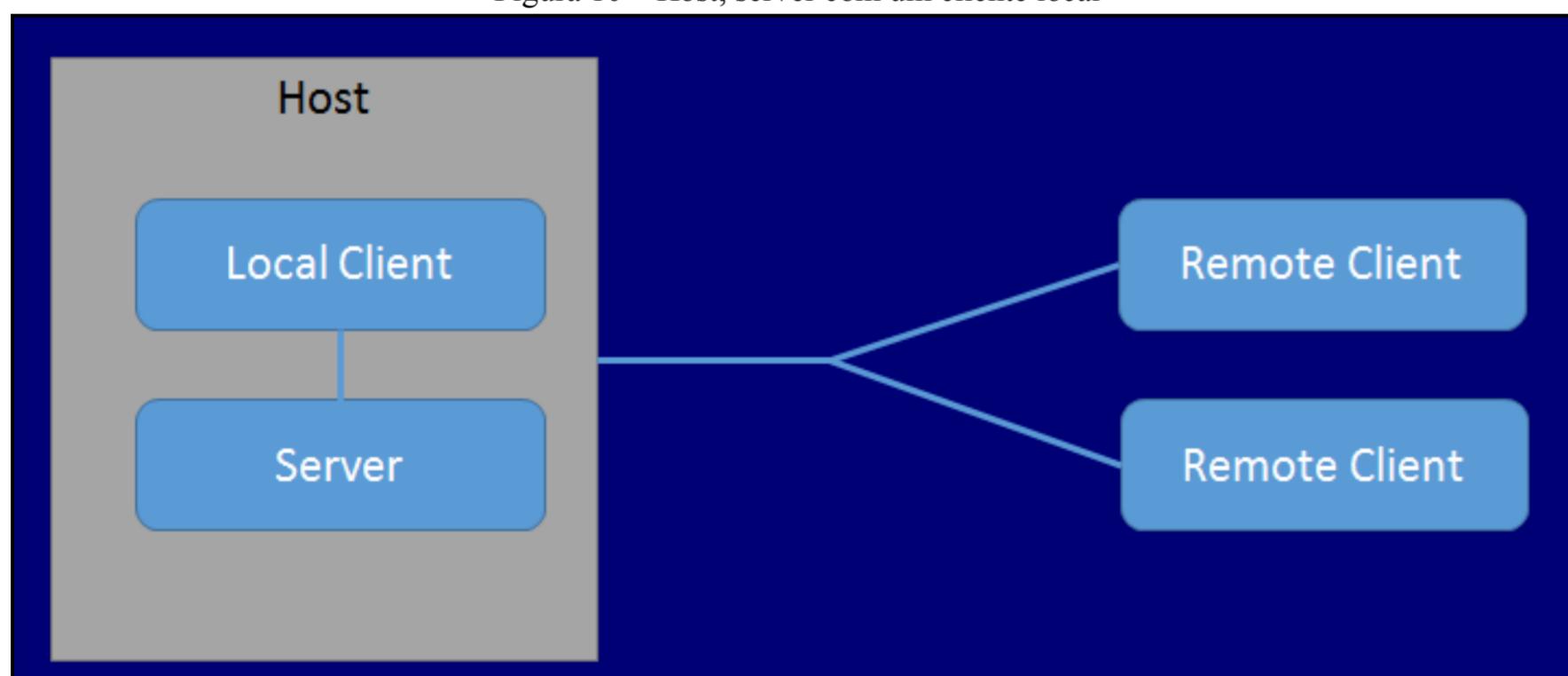


Fonte: Mirror. (2020).

Mirror (2020).

Com o mirror também é possível que o server seja um server com um cliente, chamado de host. O host é um server com um cliente local iniciado, juntamente com o server (Figura 10). Neste caso a aplicação não possui um servidor dedicado, um dos clientes faz o papel de server também. Neste projeto foi utilizada esta arquitetura, pois o aquário é um host, possui um server e um cliente local.

Figura 10 – Host, server com um cliente local



Fonte: Mirror. (2020).

No desenvolvimento do trabalho foi necessário realizar o compartilhamento da câmera no dispositivo host para os clientes, mas para isto foi encontrado duas limitações da biblioteca mirror.³⁴ Primeiro o mirror somente realiza a

Mirror

Mirror

sincronização da posição do objeto, não da sua textura, e de variáveis primitivas como por exemplo `int`. Para realizar esta sincronização foi necessário a cada *frame* converter a imagem da câmera em pixel, salvar em uma imagem separada e após isso converter esta imagem em um array de bytes. Como podemos observar no script do Quadro 5, são obtidos os pixels da câmera e salvos na variável `photo`. Após isso é reduzida a resolução da imagem em 25% do tamanho original, para poder ter um array de bytes pequeno o suficiente para ser sincronizado. Após isso é convertida a imagem para o array de bytes e este é salvo na variável `inventory` que é sincronizada entre os clientes do host.

Quadro 5 – Script para capturar a imagem da câmera e converter em array de bytes

```
Texture2D photo = new Texture2D (width, height, TextureFormat.RGB24, false);
photo.SetPixels (backCam.GetPixels ());
photo.Apply ();
TextureScale.Bilinear (photo, width / 4, height / 4);
byte[] photoByte = photo.EncodeToJPG ();
item.back = photoByte;
if (photoByte.LongCount () < 10000) {
    inventory.Clear ();
    inventory.Add (item);
}
```

Fonte: Elaborado pelo autor.

Nos clientes do host foi desenvolvido um script para converter o array de bytes novamente em imagem. Podendo assim aplicar esta imagem na textura do objeto de parede do aquário gerando o efeito de vidro, possibilitando ver fora do aquário. Conforme podemos observar o script no Quadro 6 é criada uma variável local chamada também de `photo` e uma variável local chamada `photoByte`. Na variável local `photoByte` é atribuído o array de bytes que foi sincronizado e utilizado este array, para converter novamente a imagem na variável `photo`. Após convertida a imagem é atribuída a textura do objeto, que está com este script atribuído.

Quadro 6 – Script para converter o array de bytes novamente em imagem.

```
Texture2D photo = new Texture2D (width / 4, height / 4, TextureFormat.RGB24, false);
byte[] photoByte = inventory[0].back;
photo.LoadImage (photoByte);
GetComponent<Renderer> ().material.mainTexture = photo;
```

Fonte: Elaborado pelo autor.

4 RESULTADOS

O projeto desenvolvido neste artigo teve os resultados esperados alcançados, sendo realizada a integração entre o multiplayer e a realidade virtual, possibilitando utilizar os dois recursos, juntamente com o projeto já existente do Losada (2019). Os testes do projeto foram realizados no decorrer do desenvolvimento do projeto. Com o Host sendo executado em um celular Samsung A30 e os clientes executados em um notebook Acer. Foi colocado 2 clientes juntamente com o host, totalizando 3 peixes em modo multiplayer no aquário. O projeto foi limitado para 4 clientes, devido a limitações de hardware. Na Figura 11, pode-se observar um print do celular que está executando o aquário como host. Neste pode-se observar os 3 peixes que estão sendo sincronizados entre os clientes.

1
Fonte courier.

2
câmera em uma matriz de pixels

3
Como se pode observar

4
Itálico.

5
Itálico.

6
Conforme se pode observar

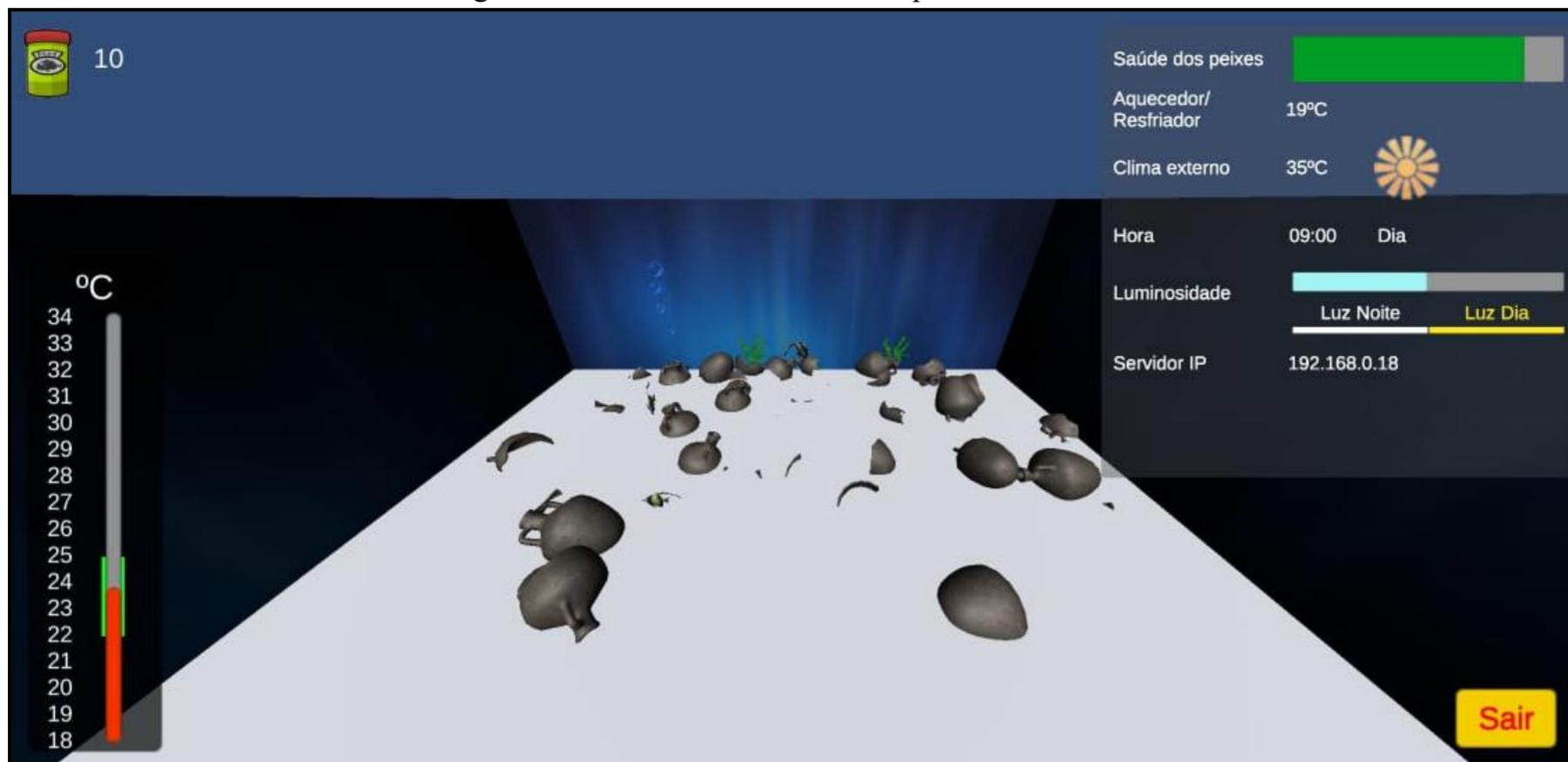
7
Itálico.

8
Remover ponto final.

9
Hum, acho que fica melhor se em todo o texto usares Host com letra maiúscula e na fonte courier.

Eu tinha colocado em itálico, mas não usar itálico.

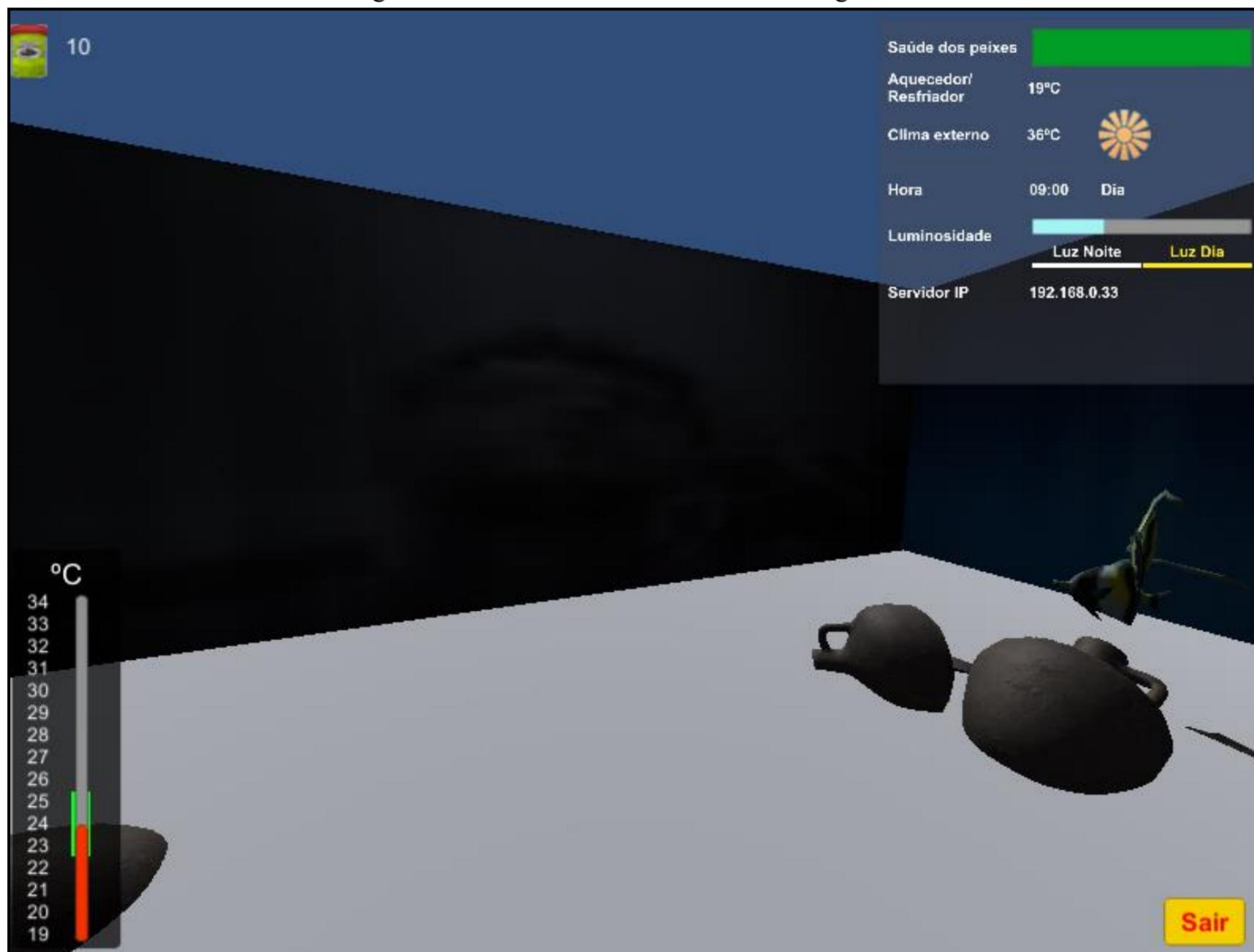
Figura 11 – Celular executando o aquário como host.



Fonte: Elaborado pelo autor.

Na Figura 12, tem-se um cliente que está conectado ao host da Figura 11, visualizando a imagem capturada pela câmera do celular host, que está sendo sincronizada. Nesta pode-se observar a captura da câmera e os outros 2 peixes clientes conectados no host.

Figura 12 – Cliente conectado no Host da Figura 11.



Com relação ao multiplayer, para realizar estes em uma rede não local, seria necessário liberar a porta 7777, configurada para a comunicação multiplayer, na biblioteca Mirror. Devido a ter um roteador bloqueado não foi possível

realizar esta liberação de porta, desta forma foi realizada a criação de uma máquina virtual na Amazon e liberada a porta 7777 e executado o host nesta máquina virtual e foi possível se conectar com os clientes, com isso podemos concluir que a funcionalidade multiplayer atende tanto redes locais e não locais, tendo em vista que para conexão com rede não local é necessário ter a porta 7777 liberado na rede do host.

Em relação a realidade virtual foram realizados diversos testes com um celular Samsung A30, que demonstraram a utilização da realidade virtual tranquila e fluída neste dispositivo, tanto executando o aquário em modo de realidade virtual com o multiplayer ativo ou não.

5 CONCLUSÕES

Os resultados esperados foram alcançados no desenvolvimento deste projeto. Houve o desenvolvimento do Aquário Virtual como um simulador de ecossistema com a funcionalidade multiplayer e da realidade virtual, demonstrando funcionar bem em conjunto.

A biblioteca Mirror se mostrou eficiente para o desenvolvimento de aplicações multiplayer e fácil utilização e aprendizado. Esta biblioteca se mostrou muito poderosa para o desenvolvimento de aplicações multiplayer, pois permite manter sincronizados vários clientes com o servidor, contanto que o servidor contenha desempenho para tal. A biblioteca permite sincronizar a animação e posição do player entre os vários clientes, de forma transparente para o desenvolvedor facilitando o desenvolvimento de aplicações multiplayer.

Mesmo que os objetivos do projeto tenham sido alcançados com sucesso, ainda há a possibilidade de extensão, conforme abaixo.

- alterar o sistema para permitir que o usuário consiga controlar o peixe com a orientação da visão da realidade virtual. Atualmente o projeto mantem a movimentação do peixe autônoma e o usuário apenas acompanha o peixe nadando pelo aquário;
- adicionar áudio e compartilhar com os clientes no multiplayer;
- adicionar novas funções ao ambiente de aquário, como objetivos que o usuário precisa cumprir para manter o peixe vivo;
- trabalhar o ecossistema incluindo novas variações, como poluição, por exemplo.

REFERÊNCIAS

ALMEIDA, Maria Elizabeth Bianconcini. **Tecnologia de Informação e Comunicação na Escola:** novos horizontes na produção escrita (2004). Disponível em: http://www.eadconsultoria.com.br/matapoio/biblioteca/textos_pdf/texto24.pdf. Acesso em: 12 de set. 2019.

BAIERLE, Ivan Luis Feix, Lopes et al.; GLUZ, João. Watt: Imersão 3D Compartilhada e Acessível na Realidade Virtual do Surgimento da Revolução Industrial, In: BRAZILIAN SYMPOSIUM ON COMPUTERS IN EDUCATION (SIMPÓSIO BRASILEIRO DE INFORMÁTICA NA EDUCAÇÃO - SBIE), 10, 2017. Rio Grande do Sul. **Anais...** SBIE 2017. 585. Disponível em: <http://www.br-ie.org/pub/index.php/sbie/article/view/7587/5383>. Acesso em: 12 set. 2019.

CARDBOARD Google. 2019. Disponível em: https://arvr.google.com/intl/pt-BR_pt/cardboard/. Acesso em: 18 out. 2019.

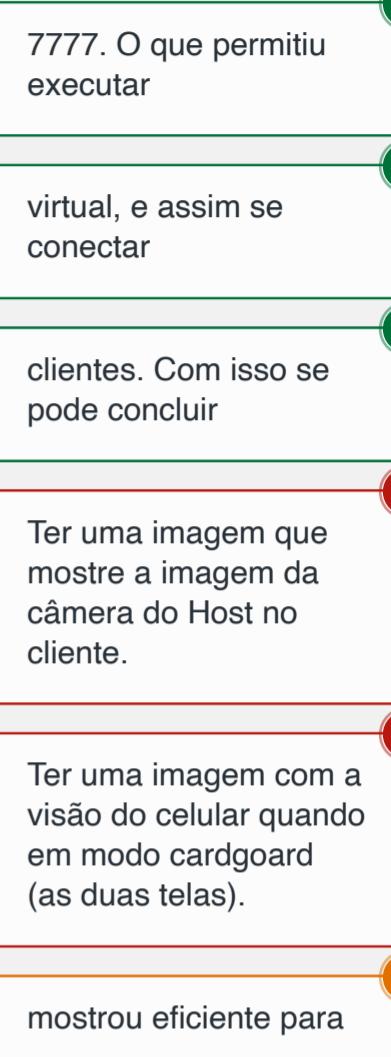
LOSADA, Flávio Omar. **Aquário Virtual:** Simulador De Ecossistema. 2019. Disponível em: http://dsc.inf.furb.br/arquivos/tccs/monografias/2019_1_flavio-omar-losada_monografia.pdf. Acesso em: 12 de set. 2019.

LUIS, Edwyn et al. Desenvolvimento e avaliação de um jogo multiplayer voltado à prática de atividades em sala de aula, In: BRAZILIAN SYMPOSIUM ON COMPUTERS IN EDUCATION (SIMPÓSIO BRASILEIRO DE INFORMÁTICA NA EDUCAÇÃO - SBIE), 10, 2016. Rio Grande do Norte. **Anais...** SBIE 2016. 647. Disponível em: <https://www.br-ie.org/pub/index.php/sbie/article/view/6746/4633>. Acesso em: 12 set. 2019.

MIRROR, Mirror. 2020. Disponível em: <https://mirror-networking.com/docs/General/index.html>. Acesso em: 20 jan. 2020.

QUEIROZ, Anna Carolina; TORI, Romero; NASCIMENTO, Alexandre. Realidade Virtual na Educação: Panorama das Pesquisas no Brasil, In: BRAZILIAN SYMPOSIUM ON COMPUTERS IN EDUCATION (SIMPÓSIO BRASILEIRO DE INFORMÁTICA NA EDUCAÇÃO - SBIE), 10, 2017. São Paulo. **Anais...** SBIE 2017. 203. Disponível em: <https://www.br-ie.org/pub/index.php/sbie/article/view/7549/5345>. Acesso em: 12 set. 2019.

SANCHES, Pablo; FAÊDA, Leonardo; MACHADO, Alex. VR Circuit: Realidade virtual aplicada ao ensino de circuitos elétricos, In: BRAZILIAN SYMPOSIUM ON COMPUTERS IN EDUCATION (SIMPÓSIO BRASILEIRO DE INFORMÁTICA NA EDUCAÇÃO - SBIE), 10. 2017. Minas Gerais. **Anais...** SBIE 2017. 887. Disponível em: <https://www.br-ie.org/pub/index.php/sbie/article/view/7617/5413>. Acesso em: 18 out. 2019.



Não encontrei está referência bibliográfica citada no texto.