

UNIVERSIDADE FEDERAL DE SANTA CATARINA
CENTRO TECNOLÓGICO
CURSO DE CIÊNCIAS DA COMPUTAÇÃO

Renan Pinho Assi

Trigate: Sistema Web para Palestras no Ambiente Virtual *OpenSim*

[Florianópolis]

[2021]

Renan Pinho Assi

Trigate: Sistema Web para Palestras no Ambiente Virtual *OpenSim*

Trabalho de Conclusão de Curso de Graduação em Curso de Ciências da Computação do Centro Tecnológico da Universidade Federal de Santa Catarina para a obtenção do título de Bacharel em Ciência da Computação.
Orientador: Prof. Rafael Luiz Cancian, Dr.

[Florianópolis]

[2021]

Renan Pinho Assi

Trigate: Sistema Web para Palestras no Ambiente Virtual *OpenSim*

Este Trabalho Conclusão de Curso foi julgado adequado para obtenção do Título de Bacharel em Ciência da Computação e aprovado em sua forma final pelo Curso de Ciências da Computação.

[Florianópolis], [24] de [09] de [2021].

Prof. Jean Everson Martina, Dr.
Coordenador(a) do Curso

Banca Examinadora:

Prof. Rafael Luiz Cancian, Dr.
Orientador
Instituição UFSC

Prof. Antonio Carlos Mariani, Me.
Avaliador
Instituição UFSC

Prof. Elder Rizzon Santos, Dr.
Avaliador
Instituição UFSC

RESUMO

Neste trabalho será explicado em detalhes todas as técnicas envolvidas para a construção de um sistema para *web* com funcionalidade de cadastro de usuários, gerenciamento de acessos, *streaming* de vídeo e sistema de *chat*. Além disso, será abordado a integração da mesma com o *OpenSim*, um ambiente de realidade virtual 3D.

Palavras-chave: Ambientes Virtuais. Educação a Distância. Open-Sim.

ABSTRACT

This thesis will explain in detail all the techniques involved to build an web application with user registration, access management and video streaming capabilities. Furthermore, the integration of this web application with OpenSim, a 3D virtual reality environment, will be addressed.

Keywords: *Virtual Environment. Distance Education. OpenSim.*

SUMÁRIO

1	INTRODUÇÃO	7
1.1	OBJETIVOS	8
2	FUNDAMENTAÇÃO TEÓRICA	9
2.1	ENSINO A DISTÂNCIA	9
2.2	FERRAMENTAS PARA ENSINO A DISTÂNCIA	10
2.2.1	Ambientes Virtuais de Aprendizagem	10
2.2.2	Plataformas de <i>Streaming</i> de Vídeo	11
2.2.3	Mundos Virtuais Tridimensionais	14
2.3	O PROJETO UFSC3D	16
3	SOLUÇÃO PROPOSTA	19
3.1	PROJETO	19
3.1.1	Requisitos	19
3.1.1.1	Requisitos Funcionais	19
3.1.1.2	Requisitos Não-Funcionais	20
3.1.2	Casos de Uso	20
3.1.2.1	Usuário Administrador	20
3.1.2.2	Usuário Palestrante	22
3.1.2.3	Usuário Aluno	23
3.1.3	Arquitetura do Sistema	24
3.1.4	Especificação	26
3.2	DESENVOLVIMENTO	31
3.2.1	Tecnologias e Infraestrutura	31
3.2.2	API REST	32
3.2.3	Aplicação de <i>Chat - Trigate Pigeon</i>	34
3.2.4	Aplicação de <i>Video Broadcasting</i>	34
3.2.5	Integração na UFSC3D	36
3.3	TESTES E VALIDAÇÃO	41
3.3.1	Testes simulando usuário	41

3.3.2	Testes de carga	42
3.4	RESULTADOS	44
4	CONCLUSÕES E RECOMENDAÇÕES PARA TRABALHOS FUTUROS	45
4.1	CONCLUSÕES	45
4.2	RECOMENDAÇÕES PARA TRABALHOS FUTUROS	46
	REFERÊNCIAS	47
5	APÊNDICES	52
5.1	APÊNDICE A - INSTALAÇÃO DO SISTEMA TRIGATE	52
5.1.1	Instalação do Portal do Sistema Trigate	53
5.1.2	Instalação dos Microsserviços de Chat e Vídeo <i>Broadcasting</i> do Sistema Trigate	53
5.2	APÊNDICE B - ARTIGO SOBRE O TCC	54

1 INTRODUÇÃO

Ao longo dos anos as formas de ensino vêm se modernizando e a modalidade de educação a distância (EaD) vem ganhando cada vez mais espaço no Brasil e no mundo. As restrições para conter o avanço da Covid-19 também fomentaram o aumento não somente da EaD como também de regimes de trabalho em casa (*Home Office*) e reuniões e palestras na *web* [18][12], o que, por sua vez, influenciou no crescimento do uso de ferramentas de comunicação *On-line* como o Zoom, Skype e outras mais. Também é preciso destacar a procura crescente de mundos virtuais para a adaptação ao contexto atual. Muitas entidades tem aderido a esses simuladores com o objetivo de trazer uma maior imersão entre as suas atividades com os seus participantes ou promover simulações de experimentos em ambientes controlados. Isso tem gerado uma nova onda de utilização massiva desses *softwares* que foram muito populares no começo da década de 2000 mas que tiveram um queda significativa no uso ao longo dos anos. O Second Life é um bom exemplo desse fenômeno. Durante a pandemia, registrou um aumento considerável na sua base de usuários, bem como um retorno significativo de usuários antigos [11].

Apesar do cenário dos últimos anos ter trazido possibilidades de inovação, ainda existe uma lacuna entre as ferramentas disponíveis pois cada uma tem seu próprio nicho específico de atuação. Este trabalho se propõem a apresentar o desenvolvimento de uma plataforma que une os pontos principais das ferramentas usadas para comunicação *On-line* e que ao mesmo tempo esteja implementada dentro de um mundo virtual, possibilitando desta forma que qualquer usuário possa fazer sua apresentação, aula ou palestra sem precisar estar vinculado no mesmo.

1.1 OBJETIVOS

O objetivo geral deste trabalho consiste no desenvolvimento de um sistema *web* que possibilite que aulas, cursos, palestras e outras formas de ensino, trabalho e interação a distância possam ocorrer de maneira mais facilitada por um mundo virtual tridimensional, aliando a facilidade de uso dos sistemas *web* com a interatividade e imersão realística dos mundos virtuais 3D.

Os objetivos específicos deste trabalho são os seguintes:

- Analisar os ambientes *web* utilizados para ensino a distância, cursos e outras formas de interação remota.
- Analisar os mundos virtuais 3D e suas aplicações no mesmo contexto.
- Desenvolver um sistema *web* para a realização de palestras que seja integrado a um mundo virtual e que inclua, ao menos: Comunicação em tempo real, apresentação de material do palestrante via tela do computador e apresentação de vídeos.
- Testar a solução proposta quanto a sua capacidade de absorção de chamadas HTTP e quanto à qualidade dos serviços disponibilizados.

2 FUNDAMENTAÇÃO TEÓRICA

2.1 ENSINO A DISTÂNCIA

A Educação a Distância (EaD) pode ser definida como "o processo de ensino-aprendizagem, mediado por tecnologias, onde professores e alunos estão separados espacial e/ou temporalmente"(MORAN, 1994). Existem registros de mais de 200 anos da prática de EaD. O marco inicial da EaD se deu no ano de 1728, onde anúncios de estudos por correspondência foram feitos em um jornal de Boston (ALVES, 2011). No Brasil, Segundo Mugnol em artigo publicado em 2009, "as primeiras iniciativas em educação a distância no Brasil se deram por meio de cursos por correspondência"[16] e essas iniciativas começaram antes de 1900 no Rio de Janeiro [20]. Com os adventos da tecnologia se consolidando cada vez mais no Brasil e no mundo, a EaD começou a ser aplicada em outros formatos. Atualmente, o Brasil que tem cerca de 134 milhões de internautas, segundo dados publicados pela Agencia Brasil em 2020, vem sendo muito receptivo ao formato de EaD via *internet*. Segundo dados do INEP apresentados pelo portal de notícias G1 [24], em 10 anos, o ensino a distância no Brasil cresceu 378,9% entre 2009 a 2019. Além disso, por conta da COVID-19, houve um aumento na procura por ensino na modalidade a distância. Segundo dados da *Udemy*, durante o período de Fevereiro a Abril de 2020, o número de inscrições em cursos na sua plataforma cresceu em 425% em escala global e 95% no Brasil [25].

Para atender a demanda crescente por EaD via internet os cursos ofertados precisam se apoiar em ferramentas que propiciem facilidades para os mais diversos objetivos como encontros entre educadores e educandos, disponibilização de conteúdos, simulação de experimentos, entre outros. Atualmente, grande parte dos cursos utilizam algum tipo de ambiente virtual de aprendizagem para gestão das disciplinas dos

cursos e dos seus usuários, além disso, estão sendo usadas plataformas de *streaming* de vídeo para aulas síncronas ou assíncronas. Por último, deve-se destacar a utilização de ambientes virtuais de tridimensionais como uma nova classe de ferramentas que vem sendo exploradas nos cursos de EaD.

2.2 FERRAMENTAS PARA ENSINO A DISTÂNCIA

Nesta seção, serão abordados 3 classes de ferramentas que são utilizados na EaD e que estão relacionadas diretamente ao sistema desenvolvido neste trabalho. Serão apresentados também, alguns exemplos de aplicações de cada classe.

2.2.1 Ambientes Virtuais de Aprendizagem

Um Ambiente Virtual de Aprendizagem (AVA) pode ser descrito como uma plataforma de gerenciamento de materiais e pessoas, tendo como papel principal centralizar toda a estrutura de um curso. Dentro das mais variadas opções disponíveis, é de consenso que as principais características são o controle de acesso dos usuários, as funcionalidades para gerenciamento de materiais do curso e o acompanhamento administrativo do progresso dos alunos (Milligan, 1999) [13]. Todavia, cada implementação de AVA pode ter características específicas. Abaixo, destaca-se alguns AVA disponíveis:

- Moodle: Lançado em 2002 e presente em mais de 200 países com mais de 250 milhões de usuários, o Moodle é um AVA de código aberto que tem como missão facilitar a criação de cursos, fornecendo muitas funcionalidades, como a criação de atividades, fórum, tarefas, controle dos alunos, importação de pacotes SCORM, etc.

- Blackboard: Com mais de 100 milhões de usuários cadastrados na plataforma, o Blackboard está presente em mais de 90 países [9]. Sua primeira versão foi liberada em janeiro de 1997 e oferece funcionalidades de criação e gestão de curso como o Moodle. O grande diferencial do Blackboard é que seu modelo de negócio principal é focado na distribuição paga, variando em funcionalidades com o tipo de licença adquirida.
- Canvas: lançado em 2011, o Canvas conta com mais de 30 milhões de usuários é o mais novo dos AVA citados até aqui. Seu código é aberto ao público e os usuários podem instalar o Canvas em sua própria estrutura sem custos, porém, também é oferecido soluções para hospedar completamente o curso na estrutura da própria empresa distribuidora do Canvas, mediante pagamento. Além disso, o Canvas fornece vários serviços como *Cloud* para armazenamento de arquivos, suporte e até serviços de integração.

2.2.2 Plataformas de *Streaming* de Vídeo

Streaming de vídeo é a técnica de enviar pequenos pacotes de vídeo para o usuário final em tempo real pela *web*. Ao mesmo tempo que o usuário assiste um vídeo, ele também está o carregando. Essa técnica evita que o vídeo tenha que ser carregado completamente para ser visualizado (Hartsell, T., & Yuen, S., 2006). Com base nisso é possível definir que plataformas de *streaming* de vídeos são sites, aplicações ou *softwares* que se utilizam dessa técnica para transmitir vídeos, sejam eles em tempo real (como *web* conferências) ou gravações. Apesar de não serem voltadas para um propósito específico, atualmente tem-se utilizado cada vez mais essas plataformas para educação e trabalho. Segundo a revista ISTOÉ, com as medidas de restrição impostas pela

COVID-19, a demanda por plataformas de *streaming* de vídeo aumentaram. A plataforma *Skype*, por exemplo, obteve um crescimento de 70% de usuários em apenas 1 mês [19]. O portal *TechRepublic* publicou dados de um monitoramento feito entre 17 de fevereiro a 6 de abril. Nesta pesquisa, das 6 plataformas monitoradas, o menor crescimento foi de 166% da plataforma *Skype*, ao passo que o maior crescimento foi da plataforma *Zoom*, com 552% [4]. Abaixo, divididas em duas sub-categorias, estão algumas das plataformas de *streaming* de vídeo importantes no cenário mundial e que são utilizadas na EaD:

Plataformas de compartilhamento de vídeos:

- **Vimeo:** Com cerca de 200 milhões de usuários, o Vimeo, fundado em 2004, é uma plataforma voltada para o compartilhamento de vídeos. Usuários podem tanto ver vídeos como fazer *uploads* dos seus próprios conteúdos na plataforma. Usuários no plano gratuito tem uma limitação de 500mb diário. Também é possível fazer transmissões ao vivo, porém, o usuário precisa ter assinado o plano "*Premium*" para acessar essa funcionalidade. Duas grandes vantagens do Vimeo em relação a outros serviços similares, é a não aplicação de propagandas dentro dos vídeos e quantidade de opções de customização, podendo inclusive não mostrar o logo do Vimeo, para *embed* dos vídeos.
- **YouTube:** Fundado em 2005 e adquirido pelo Google em 2006, o YouTube é atualmente a plataforma de compartilhamento de vídeos mais popular no mundo, tendo cerca de 2 Bilhões de usuários [3]. Assim como o Vimeo, os usuários podem enviar seus vídeos para a plataforma com o diferencial de não haver limitação de *uploads* por dia.

Plataformas de video conferência:

- **Google Meets:** Disponibilizado em 2017, o Google Meets é a continuação do Google Hangouts, plataforma de *chat* e vídeo lançada em 2013. Diferentemente do Google Hangouts que unia as funcionalidades de vídeo conferência e chat via e-mail, o Google Meets foca apenas em video conferência. Em pesquisa realizada pelo próprio Google e disponibilizada pela TechTudo [7], houve um crescimento de 275% de usuários no Brasil em 2021. Quanto a finalidade do uso da ferramenta, o Google ouviu 1551 usuários e mais de 70% responderam que utilizam o Google Meets para fins de educacionais. Além de vídeo chamada, com a possibilidade de compartilhamento de tela, entre 2 ou mais pessoas, o Google Meets pode ser gerenciado pela aplicação de agenda do Google (Google Agenda), o que permite marcar reuniões em datas determinadas. Para acessar uma reunião, não é necessário ter uma conta Google. A plataforma oferece também a funcionalidade de lousa interativa, onde todos os participantes podem escrever/desenhar. Além disso, existem funcionalidades disponíveis apenas para usuários *premium*, são elas: Dividir a chamada em grupos, enquete e gravação da chamada.
- **Skype:** Apesar de ser líder em utilização em escala global entre as plataformas de vídeo chamadas (TechRepublic, 2020), o Skype, que teve sua primeira versão em 2003 e atualmente pertence à Microsoft, está sendo substituído pelo Microsoft Teams. Além do serviço de chamadas por vídeo ou apenas por voz, a plataforma oferecia um chat integrado e a possibilidade de criação de grupos para conversas e chamadas. O Skype fornecia também funcionalidades de envio de arquivos, enquete e agendamento

de conversas integrados ao chat. Dentro da chamada de vídeo, a ferramenta possibilita a funcionalidade de "levantar a mão", o que destacava o usuário com a mão levantada.

- **Zoom:** Com um dos maiores crescimentos registrados em 2020 [4], o Zoom, que foi fundado em 2011 vem tomando espaço de outras plataformas ao longo do período de pandemia. Suas funcionalidades se assemelha muito ao Skype, muito embora o Zoom oferece a possibilidade de criar reuniões com tempos determinados e de gravação da reunião, que é disponibilizada no final do encontro.

2.2.3 Mundos Virtuais Tridimensionais

Um mundo virtual tridimensional (MVT) pode ser definido como um espaço modelado em 3 dimensões se utilizando de computação gráfica. Além disso, o mundo virtual tem como prerrogativa a dinamicidade, se adaptando/modificando as ações dos usuários (Schlemmera; Backesb, 2008). Em adição a essas características, muitos dos MVT disponibilizam linguagens de programação que são interpretadas dentro dos próprios ambientes, o que facilita a expansão e dinamicidade do mundo pelos usuários. As aplicações dos MVT são imensas, mas o interesse de aplicação desses *softwares* no EaD vem crescendo cada vez mais, visto que podem funcionar como ambientes sociais, lúdicos e até mesmo como simuladores. Para exemplificar, serão apresentados abaixo alguns MVT disponíveis.

- **Second Life:** Lançado oficial em junho de 2003, o Second Life é um dos MVT mais populares no mundo. Apesar de, desde seu lançamento, ter tido um declínio no número de usuários online, estima-se que a média de usuários fica um pouco acima de 15 mil jogadores (Tectudo, 2019) [21]. Uma das caracte-

rísticas mais marcantes do *Second Life* é a possibilidade de construção de conteúdo proprietário dentro do mundo virtual. Tudo pode ser customizado e novos objetos, com funcionalidades próprias, podem ser adicionados ao mundo virtual. Conteúdos construídos por usuários podem ser comercializados na loja do *Second Life*(<https://marketplace.secondlife.com/>) utilizando a moeda virtual do jogo (Lindens), que pode ser adquirida com dinheiro real.

- **OpenSimulator:** Baseado nos códigos liberados pela *Linden Lab*, o *OpenSimulator* é um projeto *open-source* de ambientes virtuais. Muito similar e compatível ao *Second Life*, oferece funcionalidades para customização do mundo partindo dos usuários. É possível customizar desde a aparência do seu avatar até objetos do terreno e até mesmo o próprio mapa ao qual se está interagindo. Apesar de compartilhar muitas coisas *Second Life*, o *Open Simulator* desenvolve suas próprias funcionalidades como por exemplo a capacidade de conectar um mundo virtual com outros. Diferentemente do *Second Life*, por se tratar de um projeto de código aberto, qualquer pessoa pode criar seu próprio mundo virtual.
- **Minecraft: Education Edition:** Lançado em 2016, o *Minecraft: Education Edition* é a versão para fins educacionais do jogo *sandbox* homônimo. Esta versão oferece um aplicativo que permite a visualização dos usuários/alunos por parte dos administradores/professores de forma remota ao *Minecraft*. Tutoriais direcionados aos professores estão inclusos, bem como facilidades para criação de salas de aula. Por ultimo, os usuários criar códigos por meio da funcionalidade *Code Builder* para interagir com o mundo virtual.

- **World of Warcraft:** O World of Warcraft, lançado em 2004, é um dos jogos mais jogados no mundo do gênero MMORPG, tendo cerca de 4 milhões de jogadores ativos diariamente e mais de 100 milhões cadastrados ao redor do mundo [17]. Apesar de não ter nenhuma funcionalidade voltada para fins educacionais, existem projetos que utilizam o World of Warcraft na Educação, um exemplo é o projeto WoWinSchool (<http://wowinschool.pbworks.com/w/page/5268731/FrontPage>) que busca fomentar habilidades como escrita, literatura, matemática e liderança.

Apesar de todas as ferramentas apresentadas trazerem boas funcionalidades, cada uma tem sua própria aplicabilidade. Integrar em um mundo virtual e treinar os usuários em cada uma das ferramentas exigiria um grau alto de esforço. Por esse motivo, uma solução efetiva seria a centralização dos melhores pontos dessas aplicações em uma plataforma.

2.3 O PROJETO UFSC3D

O projeto de extensão "Construção ambiente virtual 3D em multiverso para ensino, pesquisa e e extensão", simplificada e intitulada "UFSC3D" visa criar pelo menos um mundo virtual 3D que represente a UFSC e que permita que possam ser oferecidos pelo menos os serviços de divulgação da instituição, a realização de aulas, cursos e palestras, incluindo mecanismos de avaliação da aprendizagem, e de experimentação remota. Os mundos virtuais são ferramentas poderosas de simulação e possuem aplicação em diversas áreas. Embora seu uso tenha diminuído consideravelmente nos últimos anos, devido o advento das redes sociais, sua aplicabilidade e potencialidade não diminuíram. Muitas universidades e empresas privadas utilizam

Figura 1 – Algumas vistas da UFSC3D, o mundo virtual tridimensional da UFSC



(a) O prédio do INE



(b) A reitoria



(c) Vista parcial do CTC



(d) Uma sala de aula no CTC

mundos virtuais como plataforma de ensino, divulgação, comunicação e comércio eletrônico.

Um mundo virtual que represente a UFSC poderia ser usado para aulas virtuais à distância (em tempo real ou não), oferecimento de cursos, realização de feiras virtuais (como a SEPEX por exemplo), passeios virtuais em diferentes cenários (como os diferentes campi e as Fortalezas da Ilha por exemplo), possivelmente guiados por NPC (personagens virtuais interativos), comércio de serviços e produtos e várias outras opções, incluindo pesquisa em algumas áreas de conhecimento.

Este TCC está vinculado ao projeto UFSC3D e fornece uma plataforma *on-line* para comunicação em tempo real de palestrantes com usuários do OpenSim, denominada Trigate. Seu objetivo é viabilizar a interação de usuários leigos em relação aos mundos virtuais tridimensionais, como a UFSC3D, pois uma conexão direta no multiverso não seria necessária. A plataforma Trigate precisa ter serviços

para comunicação bidirecional por texto e um *streaming* de vídeo capaz de suportar uma sala de usuários, além de ser acessível de forma externa ao mundo virtual, fazendo com que o usuário da plataforma consiga se comunicar com os usuários do multiverso. Algumas visões da versão desse mundo virtual podem ser vistas na figura 1.

No próximo capítulo, serão discutidos pormenores da implementação da plataforma e de sua compatibilidade com o mundo virtual UFSC3D.

3 SOLUÇÃO PROPOSTA

3.1 PROJETO

3.1.1 Requisitos

Nesta seção serão apresentados os requisitos funcionais e não-funcionais do projeto. Como o sistema final é a junção de micros-serviços, será distinguido os requisitos em razão dos seus respectivos microsserviços.

3.1.1.1 Requisitos Funcionais

- Permitirá que os usuários possam acessar o sistema via autenticação por usuário e senha.
- Listará as salas virtuais cadastradas no banco de dados.
- O usuário deverá interagir com o *chat* via plataforma *Trigate*.
- O usuário poderá iniciar compartilhamento de tela via plataforma *Trigate*.
- A plataforma deve ser integrada ao *OpenSim*, permitindo que usuários da plataforma e usuários do *OpenSim* possam se comunicar via *chat* e vídeo.
- As conversas devem estar contidas no escopo da sala virtual.
- Usuários do *OpenSim* devem ser capazes de visualizar o vídeo transmitido em tempo real pelos usuários da plataforma.
- As transmissões de vídeo devem estar contidas no escopo da sala virtual.

3.1.1.2 Requisitos Não-Funcionais

- O servidor HTTP para a plataforma deve ser o Apache.
- O servidor HTTP para os serviços deve ser o Node.JS HTTP *server*.
- O microsserviços devem usar *WebSockets* para comunicação entre clientes.
- A interface do sistema deve ser responsiva, ou seja, se adaptar nos mais variados tamanhos de telas.
- O sistemas de suportar o funcionamento de mais de uma sala simultaneamente.
- O sistema deve suportar uma carga de pelo menos 10 usuários por sala.
- Os microsserviços de *chat* e vídeo devem ser compatíveis com o navegador do visualizador Firestorm.

3.1.2 Casos de Uso

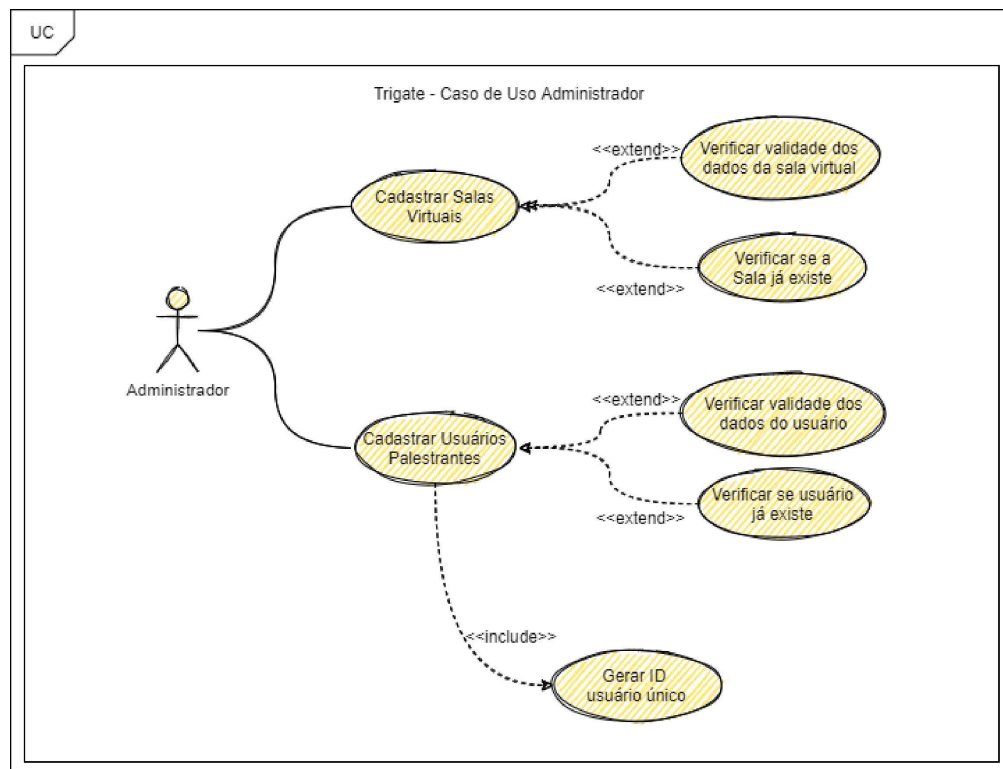
3.1.2.1 Usuário Administrador

Esse papel é desempenhado por qualquer pessoa que tenha acesso ao banco de dados, visto que essas tarefas devem ser feitas diretamente no SGBD do sistema. O administrador é responsável por cadastrar novos usuários e salas virtuais no sistema.

No cadastro de novas salas, o administrador deve preencher os dados básicos de identificação da sala virtual e criar um identificador único da mesma. O banco de dados se encarregará de verificar se todos os dados são válidos com base nas regras já estabelecidas.

Para cadastrar um novo usuário palestrante na plataforma, o administrador também deve preencher os dados descritivos bem como os dados de acesso ao sistema pelo usuário. O sistema verificará se os dados são válidos e se esse usuário não existe no sistema e caso tudo ocorra com sucesso, o sistema gera um identificador único desse usuário com base nos dados fornecidos.

Figura 2 – Caso de Uso Administrador.



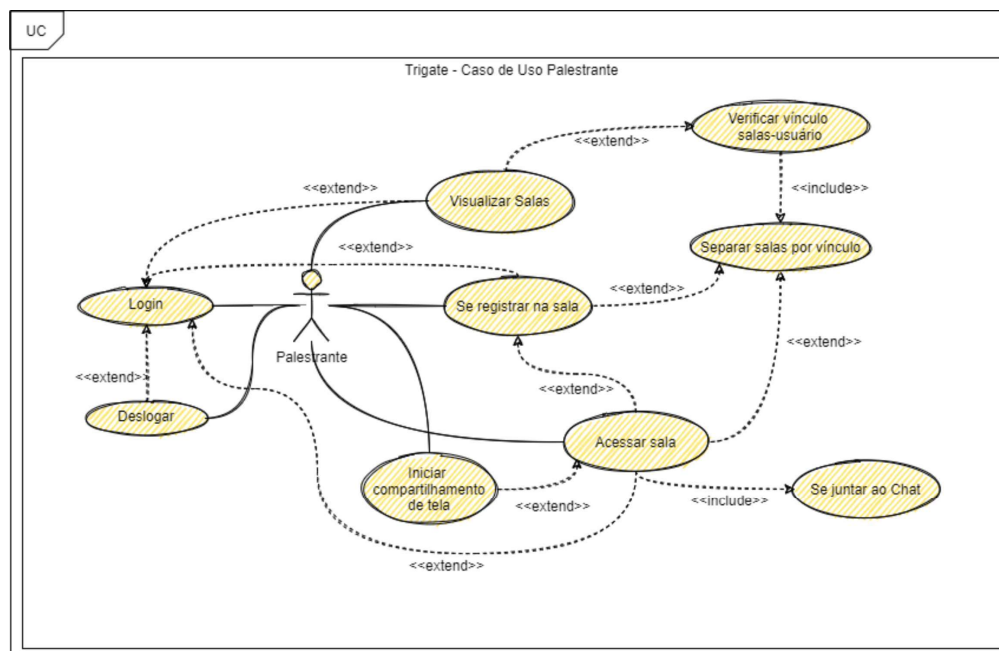
Fonte – Autor, 2021.

3.1.2.2 Usuário Palestrante

O usuário palestrante é o papel desempenhado por qualquer usuário cadastrado na plataforma com permissões de iniciar uma apresentação. Em suma, esses usuários são os professores, palestrantes e convidados. Seu acesso é exclusivamente pela plataforma Trigate, usando os seus dados de acesso para conectar-se ao sistema e então utilizar as funcionalidades de visualizar e cadastrar-se em salas virtuais e iniciar apresentações por compartilhamento de tela.

Para iniciar qualquer palestra, o usuário precisa escolher uma sala para acessar, isso pode ser feito consultando a lista de salas disponíveis no servidor. O usuário precisa estar registrado na sala, o que pode ser feito apenas com um *click* na opção "Registrar" da sala desejada. Para salas que o usuário já está vinculado, a opção "Acessar" se torna visível e com isso, o usuário acessa o ambiente privado da sala, contendo o painel de *chat* e o painel para iniciar um compartilhamento de tela.

Figura 3 – Caso de Uso Palestrante.



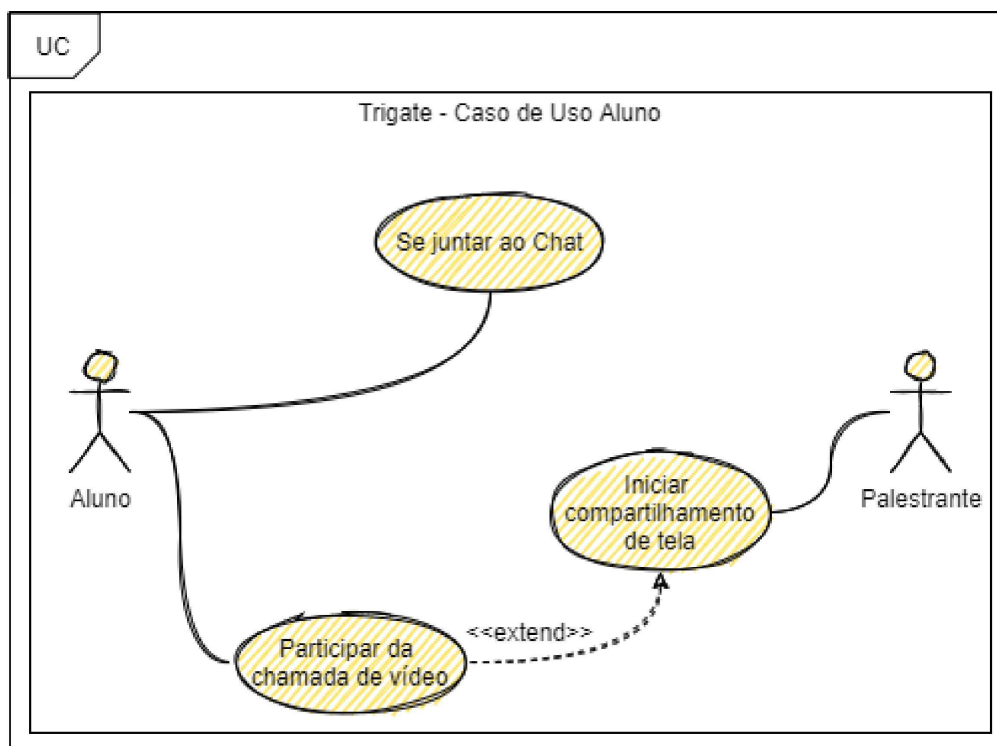
Fonte – Autor, 2021.

3.1.2.3 Usuário Aluno

O usuário aluno acessa o sistema por meio do OpenSim, mais especificamente no mundo da UFSC3D. Dentro do mundo virtual, assim como no mundo real, o aluno deverá procurar a sala a qual ele deve comparecer para acompanhar a apresentação desejada. Cada sala terá dois objetos modelos em formato de televisores contendo tanto o serviço de *Chat* como o serviço de *stream* de vídeo do apresentador (usuário palestrante). O aluno pode apertar nesses televisores para inicializar a visualização de ambos os serviços. O serviço de *chat* fica ativo inclusive nos momentos em que palestra alguma esteja ocorrendo, enquanto que o serviço de *stream* de vídeo só é efetivamente ligado

quando o usuário palestrante inicia uma apresentação.

Figura 4 – Caso de Uso Aluno.



Fonte – Autor, 2021.

3.1.3 Arquitetura do Sistema

O sistema *Trigate* é composto por diferentes componentes e portanto alguns aspectos da estrutura precisam ser destacados. Os componentes do sistema estão embasados em um servidor Ubuntu versão 18.04 da UFSC, ao todo são 2 microserviços (*chat* e *streaming* de vídeo), uma aplicação *web* (portal *Trigate*) e um banco de dados.

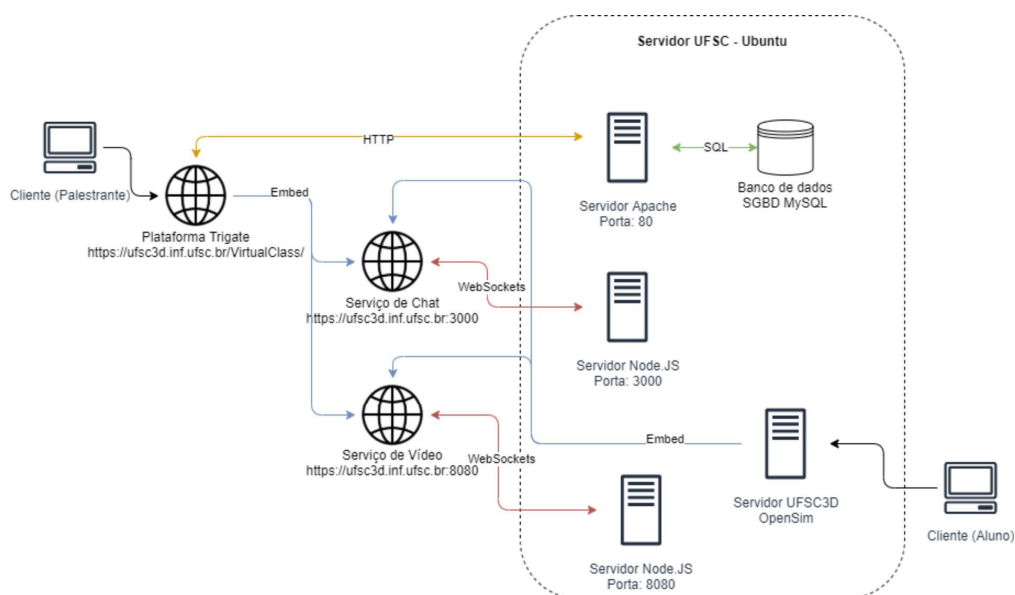
A aplicação *web* do portal está hospedada na porta 80, no subdiretório *VirtualClass*. Essa aplicação conta com o *Front-End* do

portal e da API Rest que contém métodos para a comunicação com o banco e outras funcionalidades. Os serviços de *chat* e vídeo são embutidos no portal por meio de *iframes*, o que confere uma maior independência entre todos os agentes do sistema.

Os microsserviços de *chat* e *vídeo* são hospedados por servidores *Node.js* nas portas 3000 e 8080 respectivamente. Esses serviços utilizam a biblioteca *socket.io* para viabilizar a comunicação em tempo real usando o protocolo *full-duplex WebSockets*.

Por fim, o Banco de dados, que foi construído com o SGBD *MySQL*, tem como função armazenar todos os dados pertinentes a plataforma Trigate, sendo mais especificamente os dados de cadastro dos usuários palestrantes e as salas que são cadastradas manualmente.

Figura 5 – Diagrama dos Componentes do Sistema.



Fonte – Autor, 2021.

3.1.4 Especificação

Figura 6 – Especificação do Caso de Uso "Cadastro"

Nome do UC	Realizar Cadastro	
Descrição	Este caso de uso é utilizado para cadastrar um usuário na plataforma Trigate.	
Pré-condições	Nenhuma.	
Pós-condições	Ter cadastro na plataforma.	
Ator Primário	Palestrante	
Fluxo de Eventos Principal	1. Cadastro	1.1 O sistema apresenta a tela de de cadastro com os campos "Nome da Conta", "Senha" vazios, "E-Mail", "Pseudônimo" vazios.
		1.2 O usuário preenche os campos com seus dados.
		1.3 O sistema envia um código por E-Mail e apresenta o campo de confirmação do código.
		1.4 O usuário digita o código no campo de confirmação.
		1.5 O sistema valida o código.
		1.6 O caso de uso é finalizado.
Fluxos de Exceção	E1. Usuário já existente	1. O sistema não valida os dados.
		2. A mensagem "Usuário já existe." é mostrada ao usuário.

Fonte – Autor, 2021.

Figura 7 – Especificação do Caso de Uso "Cadastro de Salas Virtuais"

Nome do UC	Cadastrar Salas Virtuais	
Descrição	Este caso de uso é utilizado para cadastrar salas virtuais na plataforma	
Pré-condições	Nenhuma.	
Pós-condições	Ter nova sala cadastrada no sistema.	
Ator Primário	Administrador	
Fluxo de Eventos Principal	1. Cadastro	1.1 O usuário acessa o SGBD da aplicação.
		1.2 O usuário preenche os dados da nova sala.
		1.3 O SGBD verifica se os dados são válidos e o ID especial da sala é único.
		1.4 O caso de uso é finalizado.
Fluxos de Exceção	E1. ID especial não é único	1. A sala não é cadastrada no banco de dados e o SGBD avisa o usuário.
	E2. Os dados não são válidos	1. A sala não é cadastrada no banco de dados e o SGBD avisa o usuário.

Fonte – Autor, 2021.

Figura 8 – Especificação do Caso de Uso "Acessar o Sistema"

Nome do UC	Realizar Login	
Descrição	Este caso de uso é utilizado para autenticar um usuário na plataforma Trigate.	
Pré-condições	Usuário ter cadastro na plataforma.	
Pós-condições	Acesso ao sistema, permitindo o uso das funcionalidades da plataforma.	
Ator Primário	Palestrante	
Fluxo de Eventos Principal	1. Login	1.1 O sistema apresenta a tela de de autenticação com os campos "Nome da Conta" e "Senha" vazios.
		1.2 O usuário preenche os campos com seus dados.
		1.3 O sistema valida os dados do usuário.
		1.4 O caso de uso é finalizado.
Fluxos de Exceção	E1. Senha e/ou Nome da Conta incorreto	1. O sistema não valida os dados.
		2. A mensagem "Usuário ou senha inválido." é mostrada ao usuário.

Fonte – Autor, 2021.

Figura 9 – Especificação do Caso de Uso "Visualizar Salas Cadastradas"

Nome do UC	Visualizar Salas Virtuais	
Descrição	Este caso de uso é utilizado para listar todas as salas virtuais cadastradas no sistema.	
Pré-condições	Estar Logado.	
Pós-condições	Lista de salas cadastradas.	
Ator Primário	Palestrante	
Fluxo de Eventos Principal	1. Iniciar <i>Streaming</i> de vídeo	1.1 O usuário acessa a área de escolha de salas.
		1.2 O sistema pesquisa as salas cadastradas e apresenta visualmente para o usuário.
		1.3 O caso de uso é finalizado.

Fonte – Autor, 2021.

Figura 10 – Especificação do Caso de Uso "Iniciar *Streaming* de vídeo"

Nome do UC	Iniciar <i>Streaming</i> de Vídeo	
Descrição	Este caso de uso é utilizado iniciar uma <i>stream</i> de vídeo.	
Pré-condições	Estar Logado; Ter escolhido uma sala.	
Pós-condições	Ter uma <i>stream</i> de vídeo.	
Ator Primário	Palestrante	
Fluxo de Eventos Principal	1. Iniciar <i>Streaming</i> de vídeo	1.1 O usuário acessa a área de <i>Streaming</i> de vídeo.
		1.2 O usuário clica no botão para iniciar a <i>stream</i> .
		1.3 A <i>stream</i> de vídeo inicializa e um preview da sua gravação é mostrada ao usuário.
		1.4 O caso de uso é finalizado.
Fluxos de Exceção	E1. Camera inexistente/bloqueada	1. O sistema não consegue iniciar a <i>stream</i> de vídeo.

Fonte – Autor, 2021.

Figura 11 – Especificação do Caso de Uso "Iniciar *Chat*"

Nome do UC	Iniciar Chat.	
Descrição	Este caso de uso é utilizado para iniciar o chat em salas virtuais.	
Pré-condições	Estar Logado; Ter acessado uma sala pela plataforma.	
Pós-condições	Estar conectado ao Chat.	
Ator Primário	Palestrante	
Fluxo de Eventos Principal	1. Iniciar Chat	1.1 O usuário acessa a sala via plataforma ou mundo virtual.
		1.2 O Chat inicializa automaticamente.
		1.3 O caso de uso é finalizado.

Fonte – Autor, 2021.

Figura 12 – Especificação do Caso de Uso "Registrar na Sala"

Nome do UC	Registrar em Sala Virtual.	
Descrição	Este caso de uso é utilizado para registrar em uma sala virtual existente.	
Pré-condições	Estar Logado; Ter acessado uma sala pela plataforma.	
Pós-condições	Estar registrado em uma sala virtual.	
Ator Primário	Palestrante	
Fluxo de Eventos Principal	1. Registro	1.1 O usuário acessa a área de salas virtuais.
		1.2 O usuário aperta no botão "Registrar"
		1.3 O sistema registra o relacionamento sala-usuário e em sequencia libera a opção "Entrar" (para acessar a sala) para o usuário.
		1.4 O caso de uso é finalizado.

Fonte – Autor, 2021.

Figura 13 – Especificação do Caso de Uso "Juntar-se ao Chat"

Nome do UC	Iniciar Chat no mundo virtual.	
Descrição	Este caso de uso é utilizado para iniciar o chat em salas virtuais na UFSC3D.	
Pré-condições	Ter acessado uma sala virtual na UFSC3D.	
Pós-condições	Estar conectado ao Chat.	
Ator Primário	Aluno	
Fluxo de Eventos Principal	1. Iniciar Chat	1.1 O usuário acessa a sala via mundo virtual.
		1.2 O usuário acessa o objeto contendo a visualização do chat clicando no mesmo para inicializar a visualização.
		1.3 O Chat inicializa automaticamente.
		1.4 O caso de uso é finalizado.

Fonte – Autor, 2021.

Figura 14 – Especificação do Caso de Uso "Juntar-se a uma videochamada"

Nome do UC	Participar da chamada de vídeo	
Descrição	Este caso de uso é utilizado para acessar uma chamada de vídeo.	
Pré-condições	Ter acessado uma sala virtual na UFSC3D; Haver uma chamada de vídeo iniciada pelo usuário palestrante.	
Pós-condições	Participar da chamada de vídeo.	
Ator Primário	Aluno	
Fluxo de Eventos Principal	1. Cadastro	1.1 O usuário acessa a sala via mundo virtual.
		1.2 O usuário acessa o objeto contendo a visualização da chamada clicando no mesmo para inicializar a visualização.
		1.3 O usuário aperta o botão inicializar do sistema de chamadas.
		1.4 O sistema gera a visualização em tempo real da chamada.
		1.5 O caso de uso é finalizado.
Fluxos de Exceção	E1. Chamada não iniciada	1. O sistema não inicializa nenhuma visualização ao usuário.

Fonte – Autor, 2021.

3.2 DESENVOLVIMENTO

3.2.1 Tecnologias e Infraestrutura

Para a construção da plataforma do sistema e de seus microsserviços foram necessários a utilização de várias tecnologias. Para uma maior integrabilidade, se optou por manter as mesmas linguagens de programação entre os microsserviços e a plataforma. Além disso, foram escolhidas linguagens consolidadas no mercado com o auxílio de poucas bibliotecas, visando assim, diminuir a quantidade de refatoração do código ao longo do tempo em decorrência de possíveis descontinuações das mesmas.

Para o *Front-End* da plataforma e dos microsserviços foram utilizados HTML e CSS para a estruturação visual e JavaScript como

linguagem de programação para a criação de funcionalidades dinâmicas. A biblioteca JQuery, bem como os *frameworks* Bootstrap e ELC - Easy Constructor também foram utilizadas no desenvolvimento *Front-End*. Essas tecnologias foram escolhidas em razão da sua popularidade e estabilidade ao longo dos anos.

Para o desenvolvimento *Back-End*, foi utilizado PHP e JavaScript (Node.JS) como linguagens de programação. As bibliotecas Socket.io e PHP-JWT foram escolhidas para facilitar certas funcionalidades. A biblioteca Socket.io tem como objetivo facilitar a comunicação via protocolo WebSocket, enquanto a PHP-JWT facilita a manipulação de *tokens* JWT.

Para as aplicações de *chat* e *streaming* de vídeo, foi escolhido o servidor Node.JS, pois o mesmo suporta a comunicação via *WebSocket* pela implementação da biblioteca Socket.io. Para o portal da plataforma, foi utilizado o servidor Apache pela popularidade do mesmo. O SGBD utilizado é o MySQL e sua escolha se deu pela ampla utilização do mesmo, o que confere muita credibilidade e fácil acesso a conteúdos explanativos.

3.2.2 API REST

A *API* usada pelo sistema foi construída em *PHP* utilizando um paradigma misto de programação. Para se fazer a virtualização das tabelas do banco de dados foi utilizado o paradigma de orientação a objetos. O paradigma procedural foi utilizado para construir cada método da *API*. Todas as consultas feitas ao banco de dados utilizam a interface *PDO* (*PHP Data Objects*). Essa interface oferece vantagens como a preparação das consultas *SQL*, o que faz uma pré-verificação se a consulta funciona da maneira esperada, executando a consulta a qual ela foi designada evitando, deste modo, ataques do tipo *SQL injection*. Outra vantagem é que o tratamento de erros nas requisições

é facilitado, pois são oferecidos métodos para receber informações sobre o resultado de uma requisição.

Para a *API* do sistema, foi elaborado um padrão de *endpoints* para retratar com clareza a ação alvo. A estrutura do padrão é relativamente simples:

$\langle \text{Url base} \rangle / + \text{api} / + \langle \text{escopo da ação} \rangle + / \langle \text{ação} \rangle$

Onde:

- *Url base*: *Url* onde o sistema reside.
- Escopo da ação: Referente ao grupo que será atingido pela ação. Um exemplo seria o acesso a informações de usuários, logo o escopo da ação seria "*users*".
- Ação: Efetivamente a ação que será feita. No mesmo exemplo supracitado, a ação seria "*get_user_info*".

Apesar de preferencialmente começar com a palavra "*get*" no começo de todas as ações que farão requisições do tipo *GET*, não existe uma padronização para nomear as ações com relação ao tipo da requisição.

É importante ressaltar que o acesso à plataforma funciona com *tokens*. Cada usuário cadastrado, no ato de acessar o sistema, recebe um *token* temporário. Esse *token* será utilizado para determinar se o usuário tem acesso a ação que se pretende performar. Nesta *API*, alguns métodos não necessitam de *token* para serem utilizados (são métodos globais que podem ser utilizados por todos os tipos de usuários e até mesmo não usuários do sistema), porém, outros métodos exigem um *token* válido. Um *token* é válido se seu tempo de expiração não foi atingido e se suas informações internas são válidas (caso haja uma alteração nas informações internas, o *token* não será decifrado com sucesso). Além disso, outras informações internas do *token* po-

dem validar ou invalidar sua utilização, como por exemplo o nível de permissão do usuário.

3.2.3 Aplicação de Chat - Trigate Pigeon

Pigeon é a aplicação de *chat* em tempo real desenvolvida ao longo do projeto. Essa aplicação se utiliza da biblioteca *socket.io* para trocar mensagens via *websockets* com outros usuário da aplicação. Um serviço simples que exige apenas um tratamento nas mensagens para retirar caracteres especiais e evitar ataques do tipo *XSS injection*. Esse tratamento, para simplificar ainda mais a mecânica da aplicação, é feita no lado do cliente apenas, muito embora, para uma versão final o aconselhado seria migrar o tratamento para o lado do servidor, evitando assim que o usuário possa driblar a mecânica de segurança implementada. Além do mecanismo de filtragem de mensagens, foi preciso implementar um ouvinte no lado do servidor que recebe uma mensagem e envia ao demais usuários da aplicação. Um ouvinte no lado do cliente, que recebe uma mensagem e a coloca na área de texto da aplicação, também se faz necessário. Para otimizar a aplicação, as mensagens pertencentes a um usuário são colocadas diretamente na sua área de texto local, sem esperar a resposta do servidor.

3.2.4 Aplicação de Video Broadcasting

Trigate Video Broadcasting ou TVB é uma aplicação de vídeo chamada em tempo real na modalidade *broadcast*, o que significa que a mídia compartilha e o áudio flui apenas de um usuário para os demais usuários. A TVB utiliza *WebRTC* e *RTCMultiConnection*, um *framework* que permite comunicação em vídeo e/ou áudio em tempo real. O projeto é apoiado por grandes empresas como o *Apple*, *Google* e *Microsoft*. Esse *framework* implementou nos navegadores

várias funcionalidades para o propósito de comunicação em tempo real e é preciso destacar duas funcionalidades:

- ***getUserMedia***: Funcionalidade para pedir acesso a câmera ou tela do dispositivo e/ou microfone. Além de pedir acesso, também recebe as *streams* das mídias caso seja permitido.
- ***RTCPeerConnection***: Habilita a comunicação entre *Peers*. Também performa outras ações para viabilizar o envio das mídias entre os pontos conectados.

Na figura abaixo, é possível ver quais navegadores oferecem suporte ao *WebRTC*.

Figura 15 – Tabela de compatibilidade do WebRTC.

IE	Edge *	Firefox	Chrome	Safari	Opera
	12-14	2-21	4-22		10-17
	³ 15-18	22-43 ⁻	23-55 ⁻	3.1-10.1	18-42 ⁻
6-10	79-81	44-76	56-81	11-13	43-67
11	83	77	83	13.1	68
		78-79	84-86	14-TP	

Fonte – <https://bloggeek.me/webrtc-browser-support>, 2021.

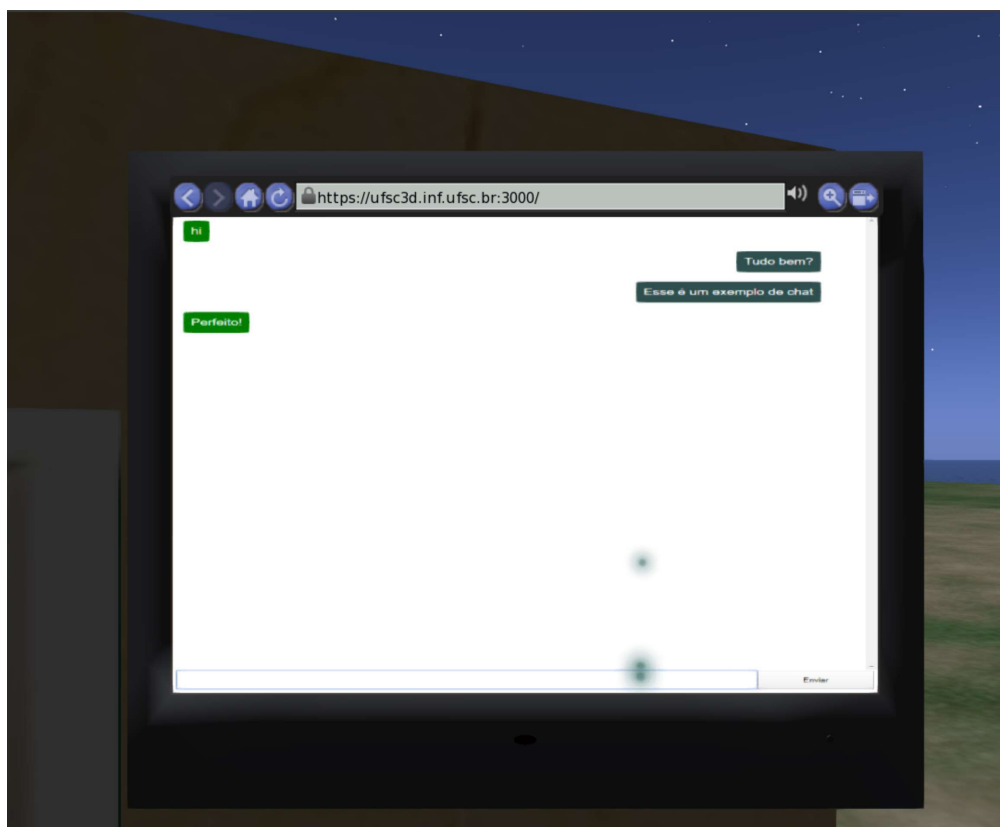
O *framework open-source RTC Multi Connection* também foi utilizado. Este *framework* oferece facilidades para conexões *peer-to-peer*, tornando o processo de conectar mais usuários em uma mesma sala mais fácil.

A TVB funciona com um sistema de salas, ao conectar no microsserviço, o usuário é colocado na sala a qual o parâmetro de *query* (parâmetro colocado após o *url*) *room-id* indica. Atualmente, a aplicação foi configurada para funcionar em um modo automático de entrar/criar sala. Caso não exista ninguém na sala, a aplicação presume que este usuário é um palestrante e então pede acesso a tela do computador e ao microfone. Se já existe algum usuário na sala, a aplicação recebe a mídia que está sendo compartilhada (som e áudio) e então aplica na tela do novo usuário conectado. O ponto principal dessa aplicação é a utilização de servidores de estabelecimento de conectividade interativa (*ICE Servers*). Seu papel é descobrir o *IP* público da máquina que está acessando a aplicação *web* e o tipo de *NAT* que esta máquina está. É preciso observar que esse passo viabiliza a conexão entre *peers* e após isso, as mídias serão compartilhadas diretamente entre os *peers*.

3.2.5 Integração na UFSC3D

As aplicações do sistema *Trigate* foram projetadas para serem utilizadas independente de qualquer outro *software* que não o navegador e por esse motivo elas podem, inclusive, serem utilizadas separadas do sistema *Trigate*. Essa característica confere uma facilidade em termos de integração com outros sistemas. O requisito mínimo para rodar as aplicações é ter um *browser* que suporte operações assíncronas e *websockets* e *WebGL*. As aplicações foram integradas em um objeto 3D, em forma de televisor, utilizando a funcionalidade *Shared Media* do *Open Simulator*.

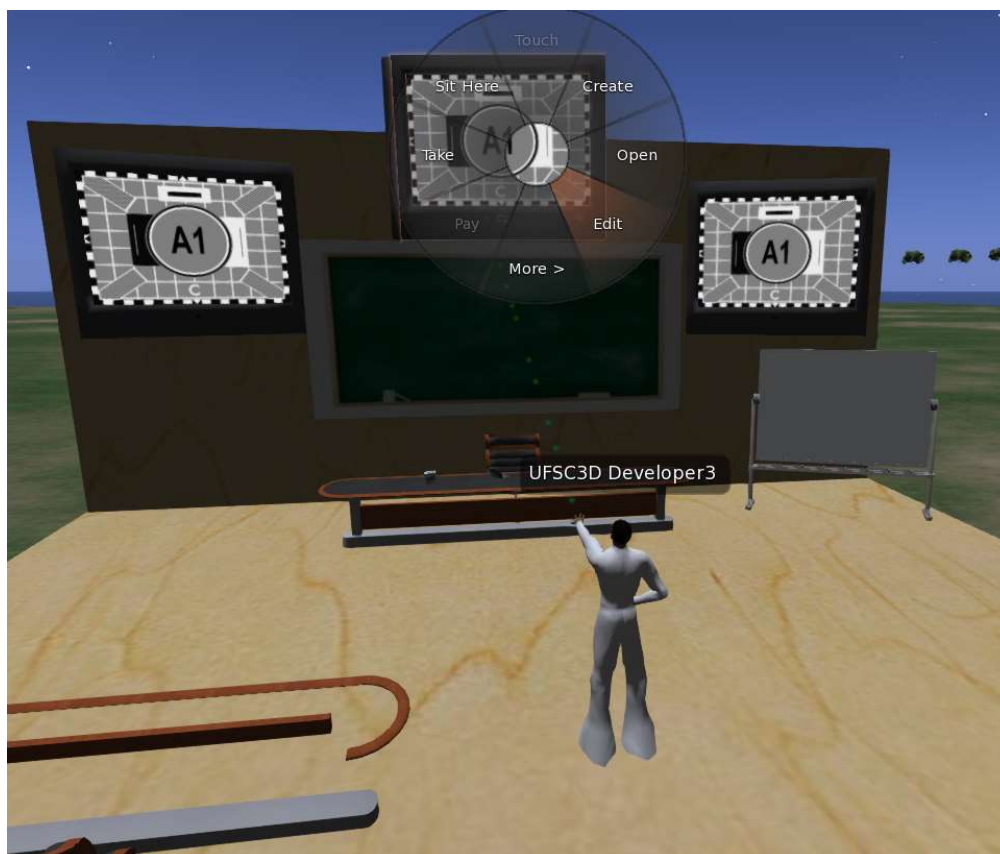
Figura 16 – Exemplo de microsserviço aplicado no OpenSim usando a função *Shared Media*.



Fonte – Autor, 2021.

Essa funcionalidade permite a visualização de um conteúdo na *web* dentro do mundo virtual. A aplicação se dá na textura de um objeto. Para isso, primeiramente é preciso selecionar o objeto e escolher a opção "*Edit*":

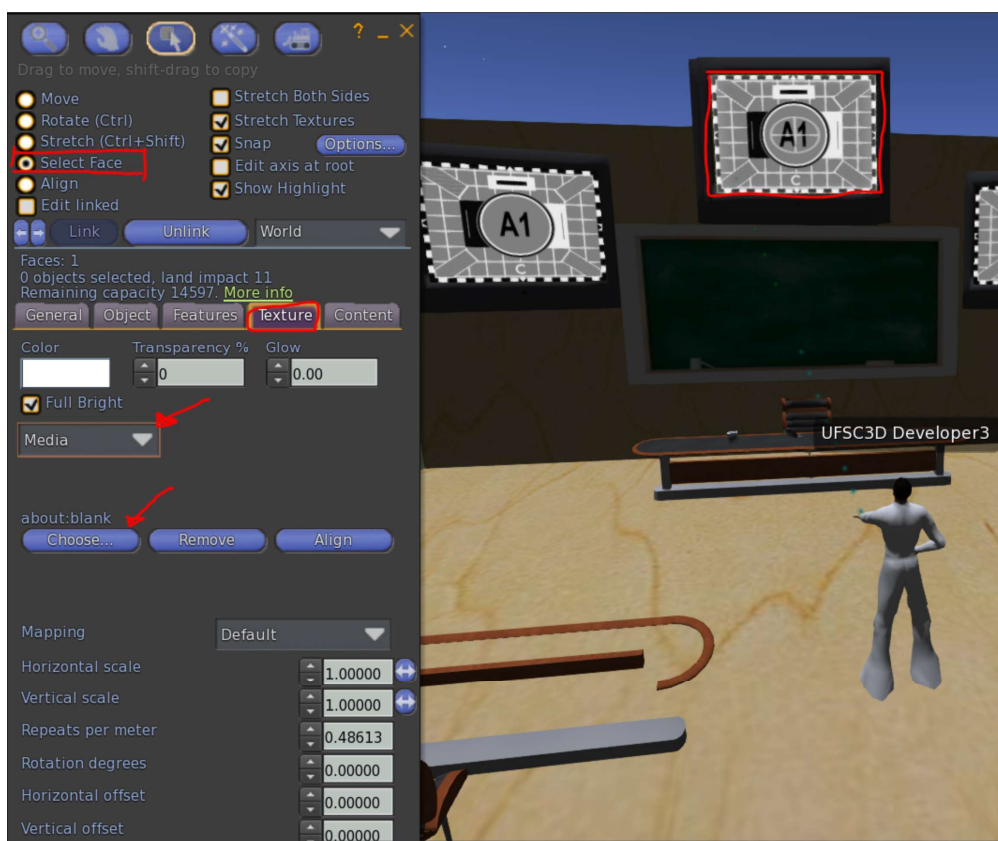
Figura 17 – Exemplo de aplicação de textura usando *Shared Media* - Passo 1.



Fonte – Autor, 2021.

Um menu com informações do objeto aparecerá, selecionaremos a seção de de escolher faces e a opção "*Select Face*". Isso possibilita a escolha da face do objeto que iremos aplicar o microserviço como textura. Com a face selecionada (clikando nela), selecionaremos a seção "*Texture*". O tipo de textura a ser usada é "*Media*". Por fim, apertaremos no botão "*Choose*":

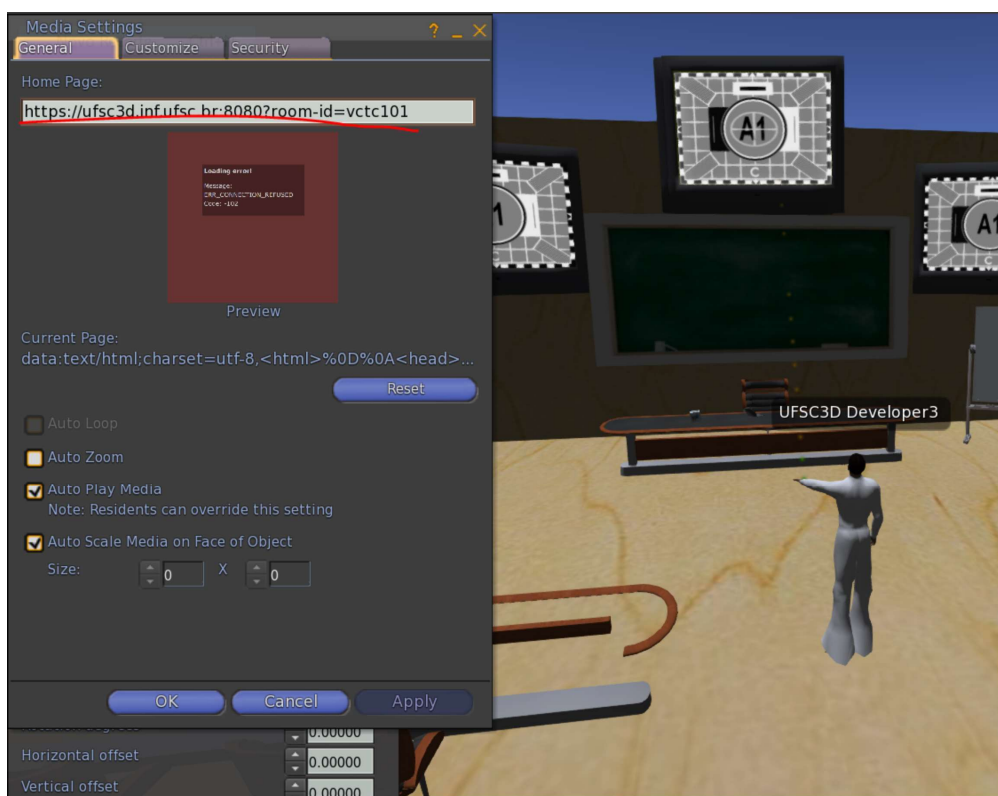
Figura 18 – Exemplo de aplicação de textura usando *Shared Media* - Passo 2.



Fonte – Autor, 2021.

Finalmente, podemos aplicar um *url* para ser a textura da face selecionada. Após inserir o *url*, devemos aperta no botão "*apply*":

Figura 19 – Exemplo de aplicação de textura usando *Shared Media* - Passo 3.



Fonte – Autor, 2021.

Após a aplicação, o usuário pode interagir com o site aplicado na textura do objeto como se estivesse usando a página diretamente no navegador.

Figura 20 – Texturas aplicadas no OpenSim.



Fonte – Autor, 2021.

3.3 TESTES E VALIDAÇÃO

Boa parte dos testes foram feitos simulando a utilização por parte de um usuário, porém, em alguns casos houve a necessidade de fazer testes mais elaborados. Neste seção serão discutidas algumas abordagens de testes utilizadas neste projeto.

3.3.1 Testes simulando usuário

Esse conjunto de testes teve como função validar funções-chaves do sistema. Primeiramente, foi testado o acesso a plataforma

utilizando dados incorretos para testar os mecanismos de bloqueio e dados corretos para testar o acesso efetivo. Também foram feitos em todas as funcionalidades menores da plataforma, como o fluxo de *logout* da plataforma, a retomada de páginas após um recarregamento do navegador e o acesso às salas virtuais. Esses testes foram aplicados tanto em um computador como em um celular, para testar a responsividade e *bugs* de *layout*. As ferramentas de *Chat* e de vídeo *broadcasting* também foram testadas nesse formato. Visualmente, foi avaliada a sua implementação dentro do portal. Quanto à ferramenta de *chat*, duas abas do navegador simularam dois diferentes usuários se comunicando via *chat* com o objetivo de avaliar a disposição das mensagens na tela, o tamanho das fontes e o tempo de resposta do servidor em relação a entrega das mensagens. O mesmo teste foi feito com o microsserviço de vídeo, mas com o objetivo de testar o funcionamento da funcionalidade *broadcast* para diferentes dispositivos. Por fim, ambos os serviços foram testados utilizando o usuário de testes da UFSC3D

Apesar de simples, a aplicação desses testes ajudaram na resolução de muitos *bugs* e também no aperfeiçoamento da usabilidade do sistema.

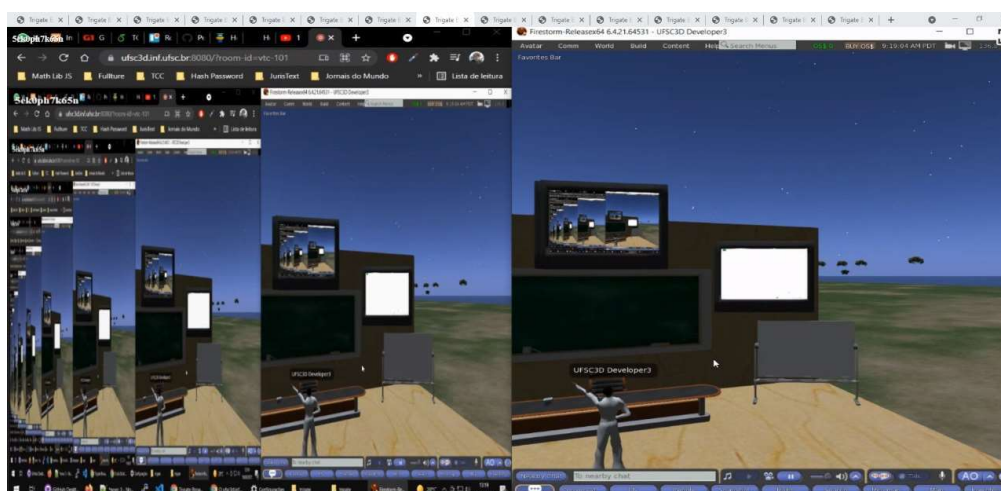
3.3.2 Testes de carga

Foram aplicados testes de carga para avaliar as condições dos componentes em relação ao servidor. Para API REST da plataforma, foi utilizado a ferramenta Postman, que permite a criação de cenários de testes com requisições HTTP. Para esse teste foram feitas 2000 requisições à API, tendo como resultado uma média de 13 ms por chamada.

Para testar a capacidade de carga do serviço de vídeo e *chat* foi elaborado um teste para simular condições de uso em situações reais.

Foram utilizados 4 dispositivos diferentes para este teste. 1 dispositivo tomou o papel do palestrante, sendo usado para fazer a captura da tela em tempo real e também para acompanhar a gravação dentro da UFSC3D e 3 dispositivos foram destinados a simular os participantes da palestra, onde foram abertos 15 abas dos serviços de *chat* e vídeo para assistir o vídeo e interagir com o *chat*.

Figura 21 – Teste de carga em um dos dispositivos ouvintes.



Fonte – Autor, 2021.

Este teste teve duração de 30 minutos e durante este tempo foi capturado um vídeo em contagem regressiva para testar o *delay* da transmissão do vídeo. Algumas mudanças de telas e outros exercícios foram feitos com o mesmo propósito. Com o serviço de *chat*, foram enviadas mensagens de várias abas, escolhidos de forma aleatória, com o objetivo de também testar o *delay* entre o envio e recebimento das mensagens.

3.4 RESULTADOS

Com o final dos testes, pode se observar que o sistema pode ser acessado de forma satisfatória tanto em computadores como em dispositivos móveis, isso pela propriedade de responsividade da plataforma, ou seja, a capacidade de se adequar aos mais diferentes formatos de telas. Apesar dos testes terem revelado que nos dispositivos móveis não é possível compartilhar a tela (por limitações do próprio navegador) e conseqüentemente apresentar uma palestra, foram feitos testes usando a câmera dos dispositivos e nessa modalidade é possível abrir uma *stream* de vídeo e com isso permitir que a palestra seja feita utilizando celulares e *tablets*.

Com os testes de carga foi possível verificar que o servidor consegue lidar com uma carga significativa de usuários. Os testes com a API mostraram que se vários usuários acessarem a ferramenta em um mesmo tempo, o servidor consegue dar boa vasão as requisições e manter o funcionamento do sistema com um tempo normal de resposta. Também foi mostrado que para cenários similares aos reais, onde se estima um número considerável de usuários, o sistema consegue manter as funções de *chat* e vídeo com tempos de *delay* irrisórios, mantendo assim um bom funcionamento no geral. Os testes também demonstraram boa conectividade entre os participantes em relação ao *streaming* de áudio, que se mostrou muito satisfatório até quando testado em conexões mais lentas. Apesar de tudo, uma acurácia melhor seria possível se fossem aplicados testes em condições reais de funcionamento, ou seja, com uma quantidade similar ao de uma turma de um curso da UFSC.

Por fim, a independência das aplicações mostrou uma facilidade muito grande quanto a integração das mesmas com outras aplicações, permitindo que a ferramenta possa atender não só o mundo virtual da UFSC3D, mas também outros mundos virtuais ao mesmo tempo.

4 CONCLUSÕES E RECOMENDAÇÕES PARA TRABALHOS FUTUROS

4.1 CONCLUSÕES

Neste trabalho, buscou-se implementar uma ferramenta para palestras com integração ao mundo virtual UFSC3D hospedado no *OpenSim*. Como objetivo, o sistema deveria entregar funcionalidades de *streaming* de vídeo e *chat* em tempo real e o palestrante deveria poder se comunicar, apenas utilizando a plataforma por meio desses serviços, com usuários do *OpenSim*. Além disso, a plataforma deveria contar com um sistema de autenticação para os palestrantes, permitindo que o usuário pudesse se cadastrar na plataforma, via confirmação de *e-mail*, para efetivamente usá-la.

Conforme relatado na seção de resultados, o objetivos que tangem a capacidade do sistema ser utilizado em tempo real em situações reais foram atingidos. Além disso, o sistema é utilizado de forma totalmente independente ao *OpenSim*, que era um dos objetivos pretendidos. Todavia, em função do tempo para a conclusão do trabalho e da migração do servidor externo ao da UFSC, que foi utilizado para desenvolver o projeto, para o servidor final da UFSC, não foi possível reestabelecer um servidor de *e-mail*, o que tornou o sistema de cadastro inativo.

Apesar do sistema estar em fase inicial, com algumas implementações ainda não amadurecidas totalmente, a capacidade de expansão, conforme foi abordado nesta seção, e com os resultados dos testes aplicados, conclui-se que a implementação de um sistema para palestras em mundos virtuais é viável.

4.2 RECOMENDAÇÕES PARA TRABALHOS FUTUROS

Para a expansão deste trabalho, recomenda-se a implementação de um provedor de *e-mail* no servidor da UFSC para que o sistema de cadastro possa ser restabelecido ou a integração do mesmo ao sistema da UFSC. Como novas funcionalidades, se recomenda a implementação de papéis diferenciados para usuários cadastrados na plataforma e a construção de uma plataforma administrativa para gerenciar o sistema. A inserção de informações na tela do palestrantes, como a quantidade de usuários presentes na palestra e por quais plataformas estão acessando, também pode ser uma possibilidade de expansão do sistema. Outra adição valiosa seria a construção de uma nova aplicação que transmitisse o áudio dos usuários no mundo virtual para a plataforma, pois proveria maior interatividade entre mundos, visto que o formato atual é de transmissão unilateral do palestrante para os ouvintes.

Ainda se tratando de novas funcionalidades, mas com um viés mais ambicioso, a integração do mundo virtual com óculos de realidade virtual se mostra muito atrativa, bem como a construção de um visualizador do mundo virtual próprio para *web* para ser utilizado dentro da plataforma. Por fim e não menos ambicioso, se sugere a criação de um mundo virtual 3D próprio para substituir ou ser usando em conjunto com o atual *OpenSim* da UFSC3D utilizado neste trabalho.

REFERÊNCIAS

- [1] Valdenise Schmitt e Maria Regina Álvares C. Dias Alice Theresinha Cybis Pereira. “Ambientes Virtuais de Aprendizagem”. Em: (). DOI: https://www.researchgate.net/publication/324573397_Ambiente_Virtual_de_Aprendizagem_em_Arquitetura_e_Design.
- [2] Lucineia Alves. “Second Educação a distância: conceitos e história no Brasil e no mundo”. Em: *Revista Brasileira de Aprendizagem Aberta e a Distância* 10.7 (2011), pp. 84–92. DOI: <https://doi.org/10.17143/rbaad.v10i0.235>.
- [3] Salman Aslam. *YouTube by the Numbers: Stats, Demographics & Fun Facts*. Acessado: 26 de agosto de 2021. Omnicore. 2021. Disp. em: <https://www.omnicoreagency.com/youtube-statistics/>.
- [4] Macy Bayern. *Zoom grew by 574% in less than two months, but Skype for Business reigns supreme*. Acessado: 26 de agosto de 2021. TechRepublic. 2020. Disp. em: <https://www.techrepublic.com/article/zoom-grew-by-574-in-less-than-two-months-but-skype-for-business-reigns-supreme/>.
- [5] BlogGeek.Me. *WebRTC browser support on desktop and mobile*. Acessado: 26 de agosto de 2021. BlogGeek.Me. 2020. Disp. em: <https://bloggeek.me/webrtc-browser-support/>.
- [6] Luciana Backes Eliane Schlemmer. “METAVERSOS: novos espaços para construção do conhecimento”. Em: *Revista Diálogo Edu-*

- cacional* 8.24 (2008), pp. 519–532. DOI: <https://www.redalyc.org/comocitar.oa?id=189116834014>.
- [7] Rodrigo Fernandes. *Google Meet completa um ano grátis e cresce no Brasil*. Acessado: 26 de agosto de 2021. TechTudo. 2020. Disp. em: <https://www.techtudo.com.br/noticias/2021/04/google-meet-completa-um-ano-gratis-e-cresce-no-brasil.ghtml>.
- [8] Blackboard Inc. *About Us*. Acessado: 26 de agosto de 2021. Blackboard Inc. 2021. Disp. em: <https://www.blackboard.com/en-eu/about-us>.
- [9] Blackboard Inc. *Blackboard Delivers Worldwide Growth*. Acessado: 26 de agosto de 2021. PR Newswire. 2017. Disp. em: <https://www.prnewswire.com/news-releases/blackboard-delivers-worldwide-growth-300398129.html#:~:text=Blackboard%20is%20the%20leading%20provider,countries%20reaching%20100%20million%20users>.
- [10] Instructure. *Our Company Story*. Acessado: 26 de agosto de 2021. Instructure. 2021. Disp. em: <https://www.instructure.com/about/our-story>.
- [11] David Kariuki. *Pandemic spurs Second Life usage, book club, lower non-profit prices*. Acessado: 26 de agosto de 2021. Hypergrid Business. 2020. Disp. em: <https://www.hypergridbusiness.com/2020/04/second-life-sees-increase-in-users-during-coronavirus-pandemic/>.

- [12] Daniel Mello. *Home office foi adotado por 46% das empresas durante a pandemia*. Acessado: 26 de agosto de 2021. Agência Brasil. 2021. Disp. em: <https://agenciabrasil.ebc.com.br/economia/noticia/2020-07/home-office-foi-adotado-por-46-das-empresas-durante-pandemia>.
- [13] Colin Milligan. *The Role of Virtual Learning Environments in the Online Delivery of Staff Development*. Acessado: 26 de agosto de 2021. Heriot-Watt University. 1999. Disp. em: <http://www.icbl.hw.ac.uk/jtap-573/573r2-3.html>.
- [14] Moodle. *About Us*. Acessado: 26 de agosto de 2021. Moodle. 2021. Disp. em: <https://moodle.com/about/>.
- [15] José Moran. *O que é educação a distância*. Acessado: 26 de agosto de 2021. 1994. Disp. em: <http://www2.eca.usp.br/moran/wp-content/uploads/2013/12/dist.pdf>.
- [16] Marcio Mugnol. “A Educação a Distância no Brasil: Conceitos e Fundamentos”. Em: *Diálogo Educacional* 9.27 (2009), pp. 335–349. DOI: <http://dx.doi.org/10.7213/rde.v9i27.3589>.
- [17] MMO Populations. *World of Warcraft: Game population, subscribers, daily players, trends, about the game, and more on MMO Populations*. Acessado: 26 de agosto de 2021. MMO Populations. 2021. Disp. em: <https://mmo-population.com/r/wow>.
- [18] Larissa Quintino. *Ofertas de vagas em regime home office crescem mais de 300% em 2020*. Acessado: 26 de agosto de 2021. Agência Brasil. 2021. Disp. em: <https://veja.abril.com>.

br/economia/ofertas-de-vagas-em-regime-home-office-crescem-309-em-2020/.

- [19] Redação. *Número de usuários do Skype cresceu 70% durante a quarentena*. Acessado: 26 de agosto de 2021. Istoé. 2020. Disp. em: <https://www.istoedinheiro.com.br/numero-de-usuarios-do-skype-cresceu-70-durante-a-quarentena/>.
- [20] Camila Rodrigues. *Cursos por correspondência, hoje em desuso, recebiam mais de mil cartas por dia*. Acessado: 26 de agosto de 2021. UOL. 2012. Disp. em: <https://educacao.uol.com.br/noticias/2012/05/16/cursos-por-correspondencia-hoje-em-desuso-recebiam-mais-de-mil-cartas-por-dia.htm>.
- [21] Thomas Schulze. *Second Life: o que mudou nos 16 anos do popular game online*. Acessado: 26 de agosto de 2021. Liden Lab. 2019. Disp. em: <https://www.techtudo.com.br/noticias/2019/11/second-life-o-que-mudou-nos-16-anos-do-popular-game-online.ghtml>.
- [22] *Statistics*. Acessado: 26 de agosto de 2021. Moodle. 2021. Disp. em: <https://stats.moodle.org/>.
- [23] Steve Chi-Yin Yuen Taralynn Hartsell. “Video Streaming in Online Learning”. Em: *AACE Review* 14.1 (2006), pp. 31–43. DOI: <http://www.learntechlib.org/p/6152/>.
- [24] Luiza Tenente. *Em 10 anos, aumenta quase 5 vezes número de alunos que entram em cursos à distância do ensino superior, diz Inep*. Acessado: 26 de agosto de 2021. G1. 2012. Disp. em:

<https://g1.globo.com/educacao/noticia/2020/10/23/em-10-anos-quase-quadruplica-numero-de-alunos-que-entram-no-ensino-superior-e-optam-pela-educacao-a-distancia-diz-inep.ghtml>.

- [25] Udemy. *Online Education Steps Up: What the World is Learning (from Home)*. Acessado: 26 de agosto de 2021. Udemy. 2021. Disp. em: https://research.udemy.com/research_report/online-education-steps-up-what-the-world-is-learning-from-home/.
- [26] Vimeo. *About Us*. Acessado: 26 de agosto de 2021. Vimeo. 2021. Disp. em: <https://vimeo.com/>.
- [27] *Who we are*. Acessado: 26 de agosto de 2021. Liden Lab. Disp. em: <https://www.lindenlab.com/about>.

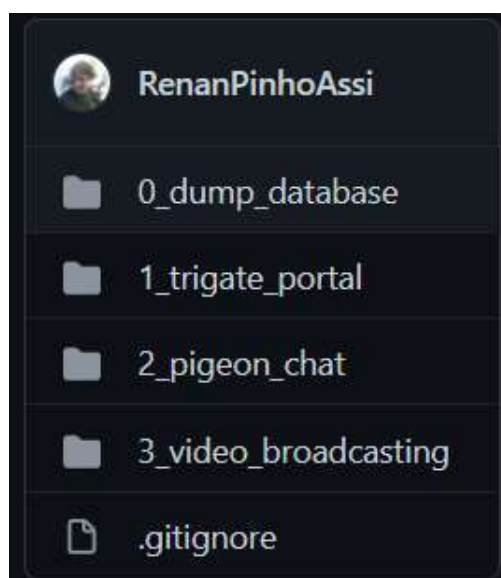
5 APÊNDICES

5.1 APÊNDICE A - INSTALAÇÃO DO SISTEMA TRIGATE

Para começar a trabalhar com o sistema é necessário baixar ou clonar o repositório público do Trigate, que fica disponível neste link: <https://github.com/RenanPinhoAssi/trigate>.

A estrutura do repositório é dividida em 4 partes conforme a figura:

Figura 22 – Estrutura do repositório do sistema Trigate.



Fonte – Autor, 2021.

É sugerido o acesso dos microsserviços e do portal Trigate via HTTPS.

5.1.1 Instalação do Portal do Sistema Trigate

Na pasta "1_trigate_portal" estão disponíveis todos os códigos relativos ao portal do sistema Trigate. Esses códigos, juntamente com a pasta "api" precisam ser hospedadas em um servidor a gosto do desenvolvedor. Além disso, é necessário a criação de um base de dados e um usuário com permissões totais de acesso a mesma. O arquivo "dump_trigate.sql", disponível na pasta "0_dump_database", pode ser usado para a criação da base de dados com as tabelas corretas e uma quantidade mínima de dados preenchidos.

Para a configuração correta do *front-end* do sistema é preciso editar a variável "base_url" no arquivo "scripts/user/global.js". Essa variável deve ter o caminho base de onde os arquivos do "front-end" estarão disponíveis.

Para a configuração da api, o arquivo "api/config/database.php" precisa ser preenchido com as informações pertinentes a base de dados criada.

5.1.2 Instalação dos Microsserviços de Chat e Vídeo *Broadcasting* do Sistema Trigate

Os microsserviços de chat e vídeo estão disponíveis, respectivamente, nas pastas "2_pigeon_chat" e "3_video_broadcasting". A instalação e inicialização de ambos os microsserviços é idêntica e por isso, os passos podem ser reproduzidos tanto em uma das pastas como na outra. Para instalar um dos microsserviços é preciso, primeiramente, instalar os pacotes demandados e suas dependências executado o comando "npm install". Com os pacotes instalados, edite os caminhos do certificado e chave privada do microsserviço. Para isso, abra arquivo "server.js" e edite as variáveis "key" e "cert". Por fim, execute o comando "node server.js" para inicializar o serviço.

5.2 APÊNDICE B - ARTIGO SOBRE O TCC

Trigate

Sistema Web para Palestras no Ambiente Virtual *OpenSim*

Renan Pinho Assi¹, Rafael Luiz Cancian¹

¹Instituto de Informatica e Estatística – Universidade Federal de Santa Catarina (UFSC)
Santa Catarina – SC – Brazil

Resumo. Neste trabalho será explicado em detalhes todas as técnicas envolvidas para a construção de um sistema para web com funcionalidade de cadastro de usuários, gerenciamento de acessos, streaming de vídeo e sistema de chat. Além disso, será abordado a integração da mesma com o *OpenSim*, um ambiente de realidade virtual 3D.

Abstract. This thesis will explain in detail all the techniques involved to build an web application with user registration, access management and video streaming capabilities. Furthermore, the integration of this web application with *OpenSim*, a 3D virtual reality environment, will be addressed.

1. Introdução

Ao longo dos anos as formas de ensino vêm se modernizando e a modalidade de educação a distância (EaD) vem ganhando cada vez mais espaço no Brasil e no mundo. As restrições para conter o avanço da Covid-19 também fomentaram o aumento não somente da EaD como também de regimes de trabalho em casa (*Home Office*) e reuniões e palestras na web [Quintino 2021][Mello 2021], o que, por sua vez, influenciou no crescimento do uso de ferramentas de comunicação *On-line* como o Zoom, Skype e outras mais. Também é preciso destacar a procura crescente de mundos virtuais para a adaptação ao contexto atual. Muitas entidades tem aderido a esses simuladores com o objetivo de trazer uma maior imersão entre as suas atividades com os seus participantes ou promover simulações de experimentos em ambientes controlados. Isso tem gerado uma nova onda de utilização massiva desses *softwares* que foram muito populares no começo da década de 2000 mas que tiveram um queda significativa no uso ao longo dos anos. O *Second Life* é um bom exemplo desse fenômeno. Durante a pandemia, registrou um aumento considerável na sua base de usuários, bem como um retorno significativo de usuários antigos [Kariuki 2020].

Apesar do cenário dos últimos anos ter trazido possibilidades de inovação, ainda existe uma lacuna entre as ferramentas disponíveis pois cada uma tem seu próprio nicho específico de atuação. Este trabalho se propõem a apresentar o desenvolvimento de uma plataforma que une os pontos principais das ferramentas usadas para comunicação *On-line* e que ao mesmo tempo esteja implementada dentro de um mundo virtual, possibilitando desta forma que qualquer usuário possa fazer sua apresentação, aula ou palestra sem precisar estar vinculado no mesmo.

2. Fundamentação Teórica

2.1. Ensino a Distância

A Educação a Distância (EaD) pode ser definida como "o processo de ensino-aprendizagem, mediado por tecnologias, onde professores e alunos estão separados espacial e/ou temporalmente" (MORAN, 1994). Existem registros de mais de 200 anos da prática de EaD.

O marco inicial da EaD se deu no ano de 1728, onde anúncios de estudos por correspondência foram feitos em um jornal de Boston (ALVES, 2011). No Brasil, Segundo Mugnol em artigo publicado em 2009, "as primeiras iniciativas em educação a distância no Brasil se deram por meio de cursos por correspondência" [Mugnol 2009] e essas iniciativas começaram antes de 1900 no Rio de Janeiro [Rodrigues 2012]. Com os adventos da tecnologia se consolidando cada vez mais no Brasil e no mundo, a EaD começou a ser aplicada em outros formatos. Atualmente, o Brasil que tem cerca de 134 milhões de internautas, segundo dados publicados pela Agência Brasil em 2020, vem sendo muito receptivo ao formato de EaD via *internet*. Segundo dados do INEP apresentados pelo portal de notícias G1 [Tenente 2012], em 10 anos, o ensino a distância no Brasil cresceu 378,9% entre 2009 a 2019. Além disso, por conta da COVID-19, houve um aumento na procura por ensino na modalidade a distância. Segundo dados da *Udemy*, durante o período de Fevereiro a Abril de 2020, o número de inscrições em cursos na sua plataforma cresceu em 425% em escala global e 95% no Brasil [Udemy 2021].

Para atender a demanda crescente por EaD via internet os cursos ofertados precisam se apoiar em ferramentas que propiciem facilidades para os mais diversos objetivos como encontros entre educadores e educandos, disponibilização de conteúdos, simulação de experimentos, entre outros. Atualmente, grande parte dos cursos utilizam algum tipo de ambiente virtual de aprendizagem para gestão das disciplinas dos cursos e dos seus usuários, além disso, estão sendo usadas plataformas de *streaming* de vídeo para aulas síncronas ou assíncronas. Por último, deve-se destacar a utilização de ambientes virtuais de tridimensionais como uma nova classe de ferramentas que vem sendo exploradas nos cursos de EaD.

2.2. Ferramentas para Ensino a Distância

Neste trabalho buscou-se incluir características de 3 tipos de classes de ferramentas que são utilizados na EaD. Esta seção irá apresentar brevemente estas classes.

2.2.1. Ambientes Virtuais de Aprendizagem

Um Ambiente Virtual de Aprendizagem (AVA) pode ser descrito como uma plataforma de gerenciamento de materiais e pessoas, tendo como papel principal centralizar toda a estrutura de um curso. Dentro das mais variadas opções disponíveis, é de consenso que as principais características são o controle de acesso dos usuários, as funcionalidades para gerenciamento de materiais do curso e o acompanhamento administrativo do progresso dos alunos (Milligan, 1999) [Milligan 1999]. Todavia, cada implementação de AVA pode ter características específicas.

2.2.2. Plataformas de *Streaming* de Vídeo

Streaming de vídeo é a técnica de enviar pequenos pacotes de vídeo para o usuário final em tempo real pela *web*. Ao mesmo tempo que o usuário assiste um vídeo, ele também está o carregando. Essa técnica evita que o vídeo tenha que ser carregado completamente para ser visualizado (Hartsell, T., & Yuen, S., 2006). Com base nisso é possível definir que plataformas de *streaming* de vídeos são sites, aplicações ou *softwares* que

se utilizam dessa técnica para transmitir vídeos, sejam eles em tempo real (como *web* conferências) ou gravações. Apesar de não serem voltadas para um propósito específico, atualmente tem-se utilizado cada vez mais essas plataformas para educação e trabalho. Segundo a revista ISTOÉ, com as medidas de restrição impostas pela COVID-19, a demanda por plataformas de *streaming* de vídeo aumentaram. A plataforma *Skype*, por exemplo, obteve um crescimento de 70% de usuários em apenas 1 mês [Redação 2020]. O portal *TechRepublic* publicou dados de um monitoramento feito entre 17 de fevereiro a 6 de abril. Nesta pesquisa, das 6 plataformas monitoradas, o menor crescimento foi de 166% da plataforma *Skype*, ao passo que o maior crescimento foi da plataforma *Zoom*, com 552% [Bayern 2020].

2.2.3. Mundos Virtuais Tridimensionais

Um mundo virtual tridimensional (MVT) pode ser definido como um espaço modelado em 3 dimensões se utilizando de computação gráfica. Além disso, o mundo virtual tem como prerrogativa a dinamicidade, se adaptando/modificando as ações dos usuários (Schlemmera; Backesb, 2008). Em adição a essas características, muitos dos MVT disponibilizam linguagens de programação que são interpretadas dentro dos próprios ambientes, o que facilita a expansão e dinamicidade do mundo pelos usuários. As aplicações dos MVT são imensas, mas o interesse de aplicação desses *softwares* no EaD vem crescendo cada vez mais, visto que podem funcionar como ambientes sociais, lúdicos e até mesmo como simuladores.

3. O projeto UFSC3D

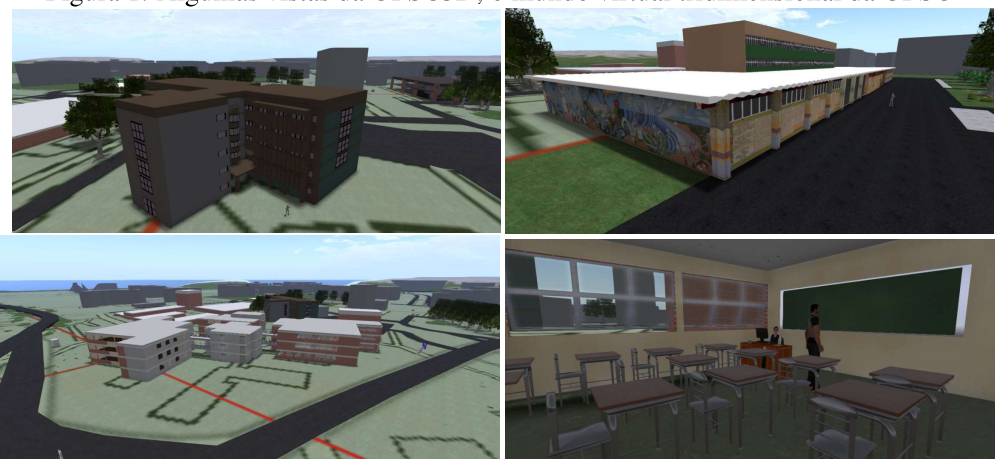
O projeto de extensão "Construção ambiente virtual 3D em multiverso para ensino, pesquisa e extensão", simplificada e intitulada "UFSC3D" visa criar pelo menos um mundo virtual 3D que represente a UFSC e que permita que possam ser oferecidos pelo menos os serviços de divulgação da instituição, a realização de aulas, cursos e palestras, incluindo mecanismos de avaliação da aprendizagem, e de experimentação remota. Os mundos virtuais são ferramentas poderosas de simulação e possuem aplicação em diversas áreas. Embora seu uso tenha diminuído consideravelmente nos últimos anos, devido o advento das redes sociais, sua aplicabilidade e potencialidade não diminuíram. Muitas universidades e empresas privadas utilizam mundos virtuais como plataforma de ensino, divulgação, comunicação e comércio eletrônico.

Um mundo virtual que represente a UFSC poderia ser usado para aulas virtuais à distância (em tempo real ou não), oferecimento de cursos, realização de feiras virtuais (como a SEPEX por exemplo), passeios virtuais em diferentes cenários (como os diferentes campi e as Fortalezas da Ilha por exemplo), possivelmente guiados por NPC (personagens virtuais interativos), comércio de serviços e produtos e várias outras opções, incluindo pesquisa em algumas áreas de conhecimento.

Este trabalho está vinculado ao projeto UFSC3D e fornece uma plataforma *on-line* para comunicação em tempo real de palestrantes com usuários do OpenSim, denominada Trigate. Seu objetivo é viabilizar a interação de usuários leigos em relação aos mundos virtuais tridimensionais, como a UFSC3D, pois uma conexão direta no multiverso não seria necessária. A plataforma Trigate precisa ter serviços para comunicação bidirecional

por texto e um *streaming* de vídeo capaz de suportar uma sala de usuários, além de ser acessível de forma externa ao mundo virtual, fazendo com que o usuário da plataforma consiga se comunicar com os usuários do multiverso. Algumas visões da versão desse mundo virtual podem ser vistas na figura 1.

Figura 1. Algumas vistas da UFSC3D, o mundo virtual tridimensional da UFSC



4. Solução Proposta

4.1. Arquitetura do Sistema

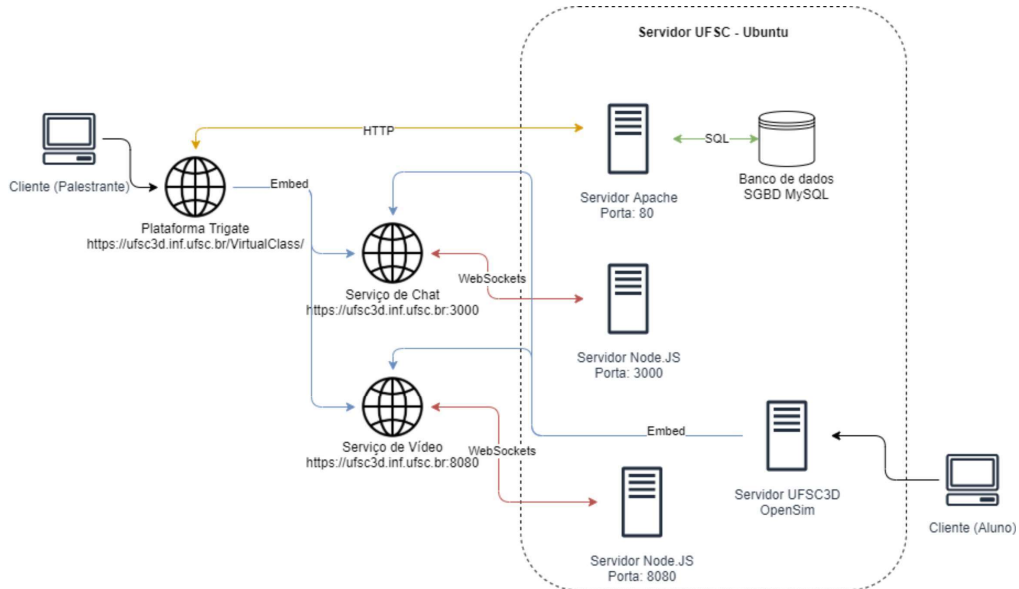
O sistema *Trigate* é composto por diferentes componentes e portanto alguns aspectos da estrutura precisam ser destacados. Os componentes do sistema estão embasados em um servidor Ubuntu versão 18.04 da UFSC, ao todo são 2 microsserviços (*chat* e *streaming* de vídeo), uma aplicação *web* (portal *Trigate*) e um banco de dados.

A aplicação *web* do portal está hospedada na porta 80, no subdiretório *Virtual-Class*. Essa aplicação conta com o *Front-End* do portal e da API Rest que contém métodos para a comunicação com o banco e outras funcionalidades. Os serviços de *chat* e vídeo são embutidos no portal por meio de *iframes*, o que confere uma maior independência entre todos os agentes do sistema.

Os microsserviços de *chat* e vídeo são hospedados por servidores *Node.js* nas portas 3000 e 8080 respectivamente. Esses serviços utilizam a biblioteca *socket.io* para viabilizar a comunicação em tempo real usando o protocolo *full-duplex WebSockets*.

Por fim, o Banco de dados, que foi construído com o SGBD *MySQL*, tem como função armazenar todos os dados pertinentes a plataforma *Trigate*, sendo mais especificamente os dados de cadastro dos usuários palestrantes e as salas que são cadastradas manualmente.

Figura 2. Diagrama dos Componentes do Sistema.



Fonte: elaborado pelo autor, 2021.

4.2. Tecnologias e Infraestrutura

Para a construção da plataforma do sistema e de seus microsserviços foram necessários a utilização de várias tecnologias. Para uma maior integrabilidade, se optou por manter as mesmas linguagens de programação entre os microsserviços e a plataforma. Além disso, foram escolhidas linguagens consolidadas no mercado com o auxílio de poucas bibliotecas, visando assim, diminuir a quantidade de refatoração do código ao longo do tempo em decorrência de possíveis descontinuações das mesmas.

Para o *Front-End* da plataforma e dos microsserviços foram utilizados HTML e CSS para a estruturação visual e JavaScript como linguagem de programação para a criação de funcionalidades dinâmicas. A biblioteca JQuery, bem como os *frameworks* Bootstrap e ELC - Easy Constructor também foram utilizadas no desenvolvimento *Front-End*. Essas tecnologias foram escolhidas em razão da sua popularidade e estabilidade ao longo dos anos.

Para o desenvolvimento *Back-End*, foi utilizado PHP e JavaScript (Node.JS) como linguagens de programação. As bibliotecas Socket.io e PHP-JWT foram escolhidas para facilitar certas funcionalidades. A biblioteca Socket.io tem como objetivo facilitar a comunicação via protocolo WebSocket, enquanto a PHP-JWT facilita a manipulação de *tokens* JWT.

Para as aplicações de *chat* e *streaming* de vídeo, foi escolhido o servidor Node.JS, pois o mesmo suporta a comunicação via *WebSocket* pela implementação da biblioteca Socket.io. Para o portal da plataforma, foi utilizado o servidor Apache pela popularidade do mesmo. O SGBD utilizado é o MySQL e sua escolha se deu pela ampla utilização do mesmo, o que confere muita credibilidade e fácil acesso a conteúdos explanativos.

4.3. API REST

A *API* usada pelo sistema foi construída em *PHP* utilizando um paradigma misto de programação. Para se fazer a virtualização das tabelas do banco de dados foi utilizado o paradigma de orientação a objetos. O paradigma procedural foi utilizado para construir cada método da *API*. Todas as consultas feitas ao banco de dados utilizam a interface *PDO (PHP Data Objects)*. Essa interface oferece vantagens como a preparação das consultas *SQL*, o que faz uma pré-verificação se a consulta funciona da maneira esperada, executando a consulta a qual ela foi designada evitando, deste modo, ataques do tipo *SQL injection*. Outra vantagem é que o tratamento de erros nas requisições é facilitado, pois são oferecidos métodos para receber informações sobre o resultado de uma requisição.

Para a *API* do sistema, foi elaborado um padrão de *endpoints* para retratar com clareza a ação alvo. A estrutura do padrão é relativamente simples:

`<Url base>/ + api/ + <escopo da ação> + /<ação>`

Onde:

- *Url base*: *Url* onde o sistema reside.
- Escopo da ação: Referente ao grupo que será atingido pela ação. Um exemplo seria o acesso a informações de usuários, logo o escopo da ação seria `"users"`.
- Ação: Efetivamente a ação que será feita. No mesmo exemplo supracitado, a ação seria `"get_user_info"`.

Apesar de preferencialmente começar com a palavra `"get"` no começo de todas as ações que farão requisições do tipo *GET*, não existe uma padronização para nomear as ações com relação ao tipo da requisição.

É importante ressaltar que o acesso à plataforma funciona com *tokens*. Cada usuário cadastrado, no ato de acessar o sistema, recebe um *token* temporário. Esse *token* será utilizado para determinar se o usuário tem acesso a ação que se pretende performar. Nesta *API*, alguns métodos não necessitam de *token* para serem utilizados (são métodos globais que podem ser utilizados por todos os tipos de usuários e até mesmo não usuários do sistema), porém, outros métodos exigem um *token* válido. Um *token* é válido se seu tempo de expiração não foi atingido e se suas informações internas são válidas (caso haja uma alteração nas informações internas, o *token* não será decifrado com sucesso). Além disso, outras informações internas do *token* podem validar ou invalidar sua utilização, como por exemplo o nível de permissão do usuário.

4.4. Aplicação de Chat - Trigate Pigeon

Pigeon é a aplicação de *chat* em tempo real desenvolvida ao longo do projeto. Essa aplicação se utiliza da biblioteca *socket.io* para trocar mensagens via *websockets* com outros usuário da aplicação. Um serviço simples que exige apenas um tratamento nas mensagens para retirar caracteres especiais e evitar ataques do tipo *XSS injection*. Esse tratamento, para simplificar ainda mais a mecânica da aplicação, é feita no lado do cliente apenas, muito embora, para uma versão final o aconselhado seria migrar o tratamento para o lado do servidor, evitando assim que o usuário possa driblar a mecânica de segurança implementada. Além do mecanismo de filtragem de mensagens, foi preciso implementar um ouvinte no lado do servidor que recebe uma mensagem e envia ao demais usuários da

aplicação. Um ouvinte no lado do cliente, que recebe uma mensagem e a coloca na área de texto da aplicação, também se faz necessário. Para otimizar a aplicação, as mensagens pertencentes a um usuário são colocadas diretamente na sua área de texto local, sem esperar a resposta do servidor.

4.5. Aplicação de Video *Broadcasting*

Trigate Video Broadcasting ou TVB é uma aplicação de vídeo chamada em tempo real na modalidade *broadcast*, o que significa que a mídia compartilha e o áudio flui apenas de um usuário para os demais usuários. A TVB utiliza *WebRTC* e *RTCMultiConnection*, um *framework* que permite comunicação em vídeo e/ou áudio em tempo real. O projeto é apoiado por grandes empresas como o *Apple*, *Google* e *Microsoft*. Esse *framework* implementou nos navegadores várias funcionalidades para o propósito de comunicação em tempo real e é preciso destacar duas funcionalidades:

- ***getUserMedia***: Funcionalidade para pedir acesso a câmera ou tela do dispositivo e/ou microfone. Além de pedir acesso, também recebe as *streams* das mídias caso seja permitido.
- ***RTCPeerConnection***: Habilita a comunicação entre *Peers*. Também performa outras ações para viabilizar o envio das mídias entre os pontos conectados.

Na figura abaixo, é possível ver quais navegadores oferecem suporte ao *WebRTC*.

Figura 3. Tabela de compatibilidade do WebRTC.

IE	Edge *	Firefox	Chrome	Safari	Opera
	12-14	2-21	4-22		10-17
	15-18	22-43	23-55	3.1-10.1	18-42
6-10	79-81	44-76	56-81	11-13	43-67
11	83	77	83	13.1	68
		78-79	84-86	14-TP	

Fonte: <https://bloggeek.me/webrtc-browser-support>, 2021.

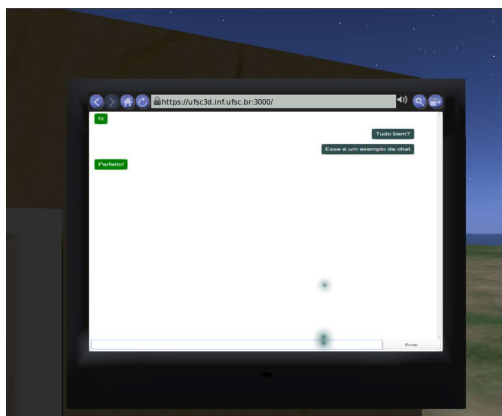
O *framework open-source RTC Multi Connection* também foi utilizado. Este *framework* oferece facilidades para conexões *peer-to-peer*, tornando o processo de conectar mais usuários em uma mesma sala mais fácil.

A TVB funciona com um sistema de salas, ao conectar no microsserviço, o usuário é colocado na sala a qual o parâmetro de *query* (parâmetro colocado após o *url*) *room-id* indica. Atualmente, a aplicação foi configurada para funcionar em um modo automático de entrar/criar sala. Caso não exista ninguém na sala, a aplicação presume que este usuário é um palestrante e então pede acesso a tela do computador e ao microfone. Se já existe algum usuário na sala, a aplicação recebe a mídia que está sendo compartilhada (som e áudio) e então aplica na tela do novo usuário conectado. O ponto principal dessa aplicação é a utilização de servidores de estabelecimento de conectividade interativa (*ICE Servers*). Seu papel é descobrir o *IP* público da máquina que está acessando a aplicação *web* e o tipo de *NAT* que esta máquina está. É preciso observar que esse passo viabiliza a conexão entre *peers* e após isso, as mídias serão compartilhadas diretamente entre os *peers*.

4.6. Integração na UFSC3D

As aplicações do sistema *Trigate* foram projetadas para serem utilizadas independente de qualquer outro *software* que não o navegador e por esse motivo elas podem, inclusive, serem utilizadas separadas do sistema *Trigate*. Essa característica confere uma facilidade em termos de integração com outros sistemas. O requisito mínimo para rodar as aplicações é ter um *browser* que suporte operações assíncronas e *websockets* e *WebGL*. As aplicações foram integradas em um objeto 3D, em forma de televisor, utilizando a funcionalidade *Shared Media* do *Open Simulator*.

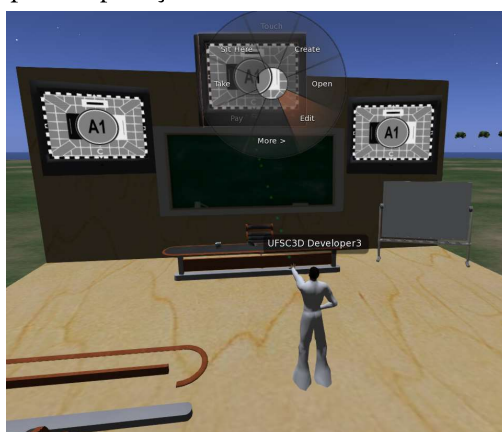
Figura 4. Exemplo de microsserviço aplicado no OpenSim usando a função *Shared Media*.



Fonte: elaborado pelo autor, 2021.

Essa funcionalidade permite a visualização de um conteúdo na *web* dentro do mundo virtual. A aplicação se dá na textura de um objeto. Para isso, primeiramente é preciso selecionar o objeto e escolher a opção *"Edit"*:

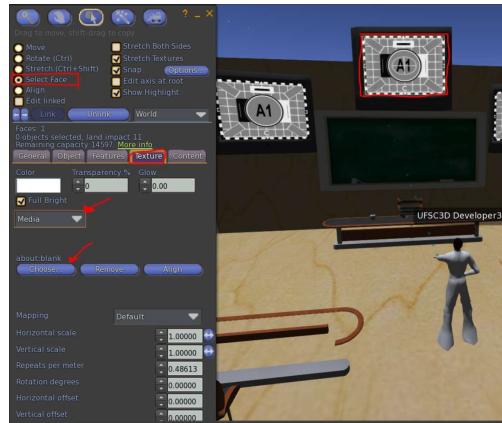
Figura 5. Exemplo de aplicação de textura usando *Shared Media* - Passo 1.



Fonte: elaborado pelo autor, 2021.

Um menu com informações do objeto aparecerá, selecionaremos a seção de escolher faces e a opção *"Select Face"*. Isso possibilita a escolha da face do objeto que iremos aplicar o microsserviço como textura. Com a face selecionada (clicando nela), selecionaremos a seção *"Texture"*. O tipo de textura a ser usada é *"Media"*. Por fim, apertaremos no botão *"Choose"*:

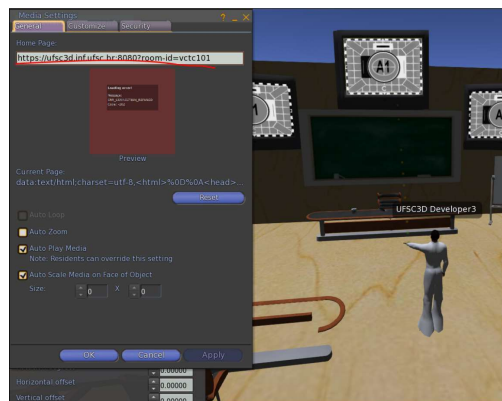
Figura 6. Exemplo de aplicação de textura usando *Shared Media* - Passo 2.



Fonte: elaborado pelo autor, 2021.

Finalmente, podemos aplicar um *url* para ser a textura da face selecionada. Após inserir o *url*, devemos aperta no botão "apply":

Figura 7. Exemplo de aplicação de textura usando *Shared Media* - Passo 3.



Fonte: elaborado pelo autor, 2021.

Após a aplicação, o usuário pode interagir com o site aplicado na textura do objeto como se estivesse usando a página diretamente no navegador.

Figura 8. Texturas aplicadas no OpenSim.



Fonte: elaborado pelo autor, 2021.

5. Testes e Validação

Boa parte dos testes foram feitos simulando a utilização por parte de um usuário, porém, em alguns casos houve a necessidade de fazer testes mais elaborados. Neste seção serão discutidas algumas abordagens de testes utilizadas neste projeto.

5.1. Testes simulando usuário

Esse conjunto de testes teve como função validar funções chaves do sistema. Primeiramente, foi testado o acesso a plataforma utilizando dados incorretos para testar os mecanismos de bloqueio e dados corretos para testar o acesso efetivo. Também foram feitos em todas as funcionalidades menores da plataforma, como o fluxo de *logout* da plataforma, a retomada de páginas após um recarregamento do navegador e o acesso às salas virtuais. Esses testes foram aplicados tanto em um computador como em um celular, para testar a responsividade e *bugs* de *layout*. As ferramentas de *Chat* e de vídeo *broadcasting* também foram testadas nesse formato. Visualmente, foi avaliada a sua implementação dentro do portal. Quanto à ferramenta de *chat*, duas abas do navegador simularam dois diferentes usuários se comunicando via *chat* com o objetivo de avaliar a disposição das mensagens na tela, o tamanho das fontes e o tempo de resposta do servidor em relação a entrega das mensagens. O mesmo teste foi feito com o microserviço de vídeo, mas com o objetivo de testar o funcionamento da funcionalidade *broadcast* para diferentes dispositivos. Por fim, ambos os serviços foram testados utilizando o usuário de testes da UFSC3D

Apesar de simples, a aplicação desses testes ajudaram na resolução de muitos *bugs* e também no aperfeiçoamento da usabilidade do sistema.

5.2. Testes de carga

Foram aplicados testes de carga para avaliar as condições dos componentes em relação ao servidor. Para API REST da plataforma, foi utilizado a ferramenta Postman, que permite a criação de cenários de testes com requisições HTTP. Para esse teste foram feitas 2000 requisições à API, tendo como resultado uma média de 13 ms por chamada.

Para testar a capacidade de carga do serviço de vídeo e *chat* foi elaborado um teste para simular condições de uso em situações reais. Foram utilizados 4 dispositivos

diferentes para este teste. 1 dispositivo tomou o papel do palestrante, sendo usado para fazer a captura da tela em tempo real e também para acompanhar a gravação dentro da UFSC3D e 3 dispositivos foram destinados a simular os participantes da palestra, onde foram abertos 15 abas dos serviços de *chat* e vídeo para assistir o vídeo e interagir com o *chat*.

Este teste teve duração de 30 minutos e durante este tempo foi capturado um vídeo em contagem regressiva para testar o *delay* da transmissão do vídeo. Algumas mudanças de telas e outros exercícios foram feitos com o mesmo propósito. Com o serviço de *chat*, foram enviadas mensagens de várias abas, escolhidos de forma aleatória, com o objetivo de também testar o *delay* entre o envio e recebimento das mensagens.

6. Resultados

Com o final dos testes, pode se observar que o sistema pode ser acessado de forma satisfatória tanto em computadores como em dispositivos móveis, isso pela propriedade de responsividade da plataforma, ou seja, a capacidade de se adequar aos mais diferentes formatos de telas. Apesar dos testes terem revelado que nos dispositivos móveis não é possível compartilhar a tela (por limitações do próprio navegador) e conseqüentemente apresentar uma palestra, foram feitos testes usando a câmera dos dispositivos e nessa modalidade é possível abrir uma *stream* de vídeo e com isso permitir que a palestra seja feita utilizando celulares e *tablets*.

Com os testes de carga foi possível verificar que o servidor consegue lidar com uma carga significativa de usuários. Os testes com a API mostraram que se vários usuários acessarem a ferramenta em um mesmo tempo, o servidor consegue dar boa vasão as requisições e manter o funcionamento do sistema com um tempo normal de resposta. Também foi mostrado que para cenários similares aos reais, onde se estima um número considerável de usuários, o sistema consegue manter as funções de *chat* e vídeo com tempos de *delay* irrisórios, mantendo assim um bom funcionamento no geral. Os testes também demonstraram boa conectividade entre os participantes em relação ao *streaming* de áudio, que se mostrou muito satisfatório até quando testado em conexões mais lentas. Apesar de tudo, uma acurácia melhor seria possível se fossem aplicados testes em condições reais de funcionamento, ou seja, com uma quantidade similar ao de uma turma de um curso da UFSC.

Por fim, a independência das aplicações mostrou uma facilidade muito grande quanto a integração das mesmas com outras aplicações, permitindo que a ferramenta possa atender não só o mundo virtual da UFSC3D, mas também outros mundos virtuais ao mesmo tempo.

7. Conclusões

Neste trabalho, buscou-se implementar uma ferramenta para palestras com integração ao mundo virtual UFSC3D hospedado no *OpenSim*. Como objetivo, o sistema deveria entregar funcionalidades de *streaming* de vídeo e *chat* em tempo real e o palestrante deveria poder se comunicar, apenas utilizando a plataforma por meio desses serviços, com usuários do *OpenSim*. Além disso, a plataforma deveria contar com um sistema de autenticação para os palestrantes, permitindo que o usuário pudesse se cadastrar na plataforma, via confirmação de *e-mail*, para efetivamente usá-la.

Conforme relatado na seção de resultados, o objetivos que tangem a capacidade do sistema ser utilizado em tempo real em situações reais foram atingidos. Além disso, o sistema é utilizado de forma totalmente independente ao *OpenSim*, que era um dos objetivos pretendidos. Todavia, em função do tempo para a conclusão do trabalho e da migração do servidor externo ao da UFSC, que foi utilizado para desenvolver o projeto, para o servidor final da UFSC, não foi possível reestabelecer um servidor de *e-mail*, o que tornou o sistema de cadastro inativo.

Apesar do sistema estar em fase inicial, com algumas implementações ainda não amadurecidas totalmente, a capacidade de expansão, conforme foi abordado nesta seção, e com os resultados dos testes aplicados, conclui-se que a implementação de um sistema para palestras em mundos virtuais é viável.

8. Recomendações para Trabalhos Futuros

Para a expansão deste trabalho, recomenda-se a implementação de um provedor de *e-mail* no servidor da UFSC para que o sistema de cadastro possa ser restabelecido ou a integração do mesmo ao sistema da UFSC. Como novas funcionalidades, se recomenda a implementação de papéis diferenciados para usuários cadastrados na plataforma e a construção de uma plataforma administrativa para gerenciar o sistema. A inserção de informações na tela do palestrantes, como a quantidade de usuários presentes na palestra e por quais plataformas estão acessando, também pode ser uma possibilidade de expansão do sistema. Outra adição valiosa seria a construção de uma nova aplicação que transmitisse o áudio dos usuários no mundo virtual para a plataforma, pois proveria maior interatividade entre mundos, visto que o formato atual é de transmissão unilateral do palestrante para os ouvintes.

Ainda se tratando de novas funcionalidades, mas com um viés mais ambicioso, a integração do mundo virtual com óculos de realidade virtual se mostra muito atrativa, bem como a construção de um visualizador do mundo virtual próprio para *web* para ser utilizado dentro da plataforma. Por fim e não menos ambicioso, se sugere a criação de um mundo virtual 3D próprio para substituir ou ser usando em conjunto com o atual *OpenSim* da UFSC3D utilizado neste trabalho.

9. Referências

- Alves, L. (2011). Second educação a distância: conceitos e história no brasil e no mundo. *Revista Brasileira de Aprendizagem Aberta e a Distância*, 10(7):84–92.
- Bayern, M. (2020). Zoom grew by 574% in less than two months, but skype for business reigns supreme. Acessado: 26 de agosto de 2021.
- BlogGeek.Me (2020). WebRTC browser support on desktop and mobile. Acessado: 26 de agosto de 2021.
- Eliane Schlemmer, L. B. (2008). Metaversos: novos espaços para construção do conhecimento. *Revista Diálogo Educacional*, 8(24):519–532.
- Kariuki, D. (2020). Pandemic spurs second life usage, book club, lower non-profit prices. Acessado: 26 de agosto de 2021.
- Mello, D. (2021). Home office foi adotado por 46% das empresas durante a pandemia. Acessado: 26 de agosto de 2021.

- Milligan, C. (1999). The role of virtual learning environments in the online delivery of staff development. Acessado: 26 de agosto de 2021.
- Moran, J. (1994). O que é educação a distância. Acessado: 26 de agosto de 2021.
- Mugnol, M. (2009). A educação a distância no brasil: Conceitos e fundamentos. *Diálogo Educacional*, 9(27):335–349.
- Quintino, L. (2021). Ofertas de vagas em regime home office crescem mais de 300% em 2020. Acessado: 26 de agosto de 2021.
- Redação (2020). Número de usuários do skype cresceu 70% durante a quarentena. Acessado: 26 de agosto de 2021.
- Rodrigues, C. (2012). Cursos por correspondência, hoje em desuso, recebiam mais de mil cartas por dia. Acessado: 26 de agosto de 2021.
- Taralynn Hartsell, S. C.-Y. Y. (2006). Video streaming in online learning. *AACE Review*, 14(1):31–43.
- Tenente, L. (2012). Em 10 anos, aumenta quase 5 vezes número de alunos que entram em cursos à distância do ensino superior, diz inep. Acessado: 26 de agosto de 2021.
- Udemy (2021). Online education steps up: What the world is learning (from home). Acessado: 26 de agosto de 2021.