

UNIVERSIDADE ESTADUAL PAULISTA “JÚLIO DE MESQUITA FILHO”

FACULDADE DE CIÊNCIAS

BACHARELADO EM SISTEMAS DE INFORMAÇÃO

BRUNA DOS SANTOS PASSERANI

**SISTEMA DE GERENCIAMENTO DE PACIENTES E MANIPULAÇÃO DE
IMAGENS MÉDICAS (SGPMIM)**

BAURU - SP

2019

BRUNA DOS SANTOS PASSERANI

**SISTEMA DE GERENCIAMENTO DE PACIENTES E MANIPULAÇÃO DE
IMAGENS MÉDICAS (SGPMIM)**

Trabalho de Conclusão de Curso apresentado à
Universidade Estadual Paulista “Júlio de
Mesquita Filho” como exigência para
finalização do curso de Bacharelado em
Sistemas de Informação da Faculdade de
Ciências – UNESP Campus Bauru.

Orientador: Prof. Dr. José Remo Ferreira
Brega

BAURU - SP

2019

RESUMO

Imagens médicas vêm sendo utilizadas há vários anos para realização de diagnósticos e exames. Desde os primórdios, o ser humano possui curiosidade sobre a aparência de seu próprio corpo, realizando autópsias para obtenção das primeiras imagens anatômicas. Porém, com o advento da tecnologia, as imagens médicas tendem a ser minimamente invasivas, precisas e até mesmo interativas. Os sistemas para análise destas, entretanto, são geralmente complexos e requerem um conhecimento aprofundado sobre sua funcionalidade. Assim, esta monografia apresenta uma proposta de solução quanto à visualização e manipulação de neuroimagens com o formato NIfTI-1, bem como o gerenciamento de informações de pacientes, utilizando a técnica de Realidade Virtual para tornar a experiência do usuário mais intuitiva e imersiva.

Palavras-chave: NIfTI-1. Neuroimagem. Realidade Virtual. Ressonância Magnética. Unity.

ABSTRACT

Medical images have been used for several years to perform diagnoses and tests. From the earliest times, the human being has curiosity about the appearance of his own body, performing autopsies to obtain the first anatomical images. However, with the advent of technology, medical images tends to be minimally invasive, accurate, and even interactive. The systems which analyses that images, although, are often complex and require in-depth knowledge about their functionality. Thus, this monograph presents a solution proposal regarding the visualization and manipulation of neuroimaging with the NIfTI-1 format, as well as patient information management, using the Virtual Reality technique to make the user experience more intuitive and immersive.

Keywords: NIfTI-1. Neuroimaging. Virtual Reality. Magnetic Resonance. Unity.

LISTA DE FIGURAS

Figura 1 - Menu da Aplicação BrainVR	14
Figura 2 - Tela Inicial do Laudo	15
Figura 3 - Diagrama de Caso de Uso	18
Figura 4 - Diagrama de Classe	19
Figura 5 - Diagrama de Sequência: Efetuar Login	20
Figura 6 - Diagrama de Sequência: Cadastrar Usuário	20
Figura 7 - Diagrama de Sequência: Cadastrar Paciente	21
Figura 8 - Diagrama de Sequência: Cadastrar Exame	22
Figura 9 - Diagrama de Sequência: Pesquisar Paciente	22
Figura 10 - Diagrama de Sequência: Selecionar Neuroimagem Paciente	23
Figura 11 - Diagrama de Entidade e Relacionamento	24
Figura 12 - HMD VR Box	26
Figura 13 - Representação do Funcionamento do SGPMIM	29
Figura 14 - Método “selecionarDiretorio()” do SGPMIM	30
Figura 15 - Método “Cadastrar()” do <i>script</i> “PacienteMotor”	31
Figura 16 - <i>Script</i> “cadPac” do SGPMIM	32
Figura 17 - Divisão Geral do SGPMIM	33
Figura 18 - Tela “Login”	34
Figura 19 - Menu Principal do SGPMIM	35
Figura 20 - Tela “Cadastro de Usuário”	36
Figura 21 - Tela “Cadastro de Paciente”	36

Figura 22 - Menu de Seleção de Imagem NIfTI-1	37
Figura 23 - Interface de Manipulação de Neuroimagem	38
Figura 24 - Visualização de Diferentes Pontos da Neuroimagem	38
Figura 25 - Seleção da Pasta de Armazenamento de Prints	39
Figura 26 - Interface Padrão Trinus VR Para Conexão	40
Figura 27 - Neuroimagem Renderizada em Realidade Virtual	41
Figura 28 - Tela “Pesquisar Paciente”	42
Figura 29 - Tela “Cadastro de Exame”	43
Figura 30 - Interface de Cadastro de Imagem	43
Figura 31 - Tela “Histórico - Paciente”	44

LISTA DE SIGLAS

2D	Bidimensional
3D	Tridimensional
API	<i>Application Programming Interface</i> (Interface de Programação de Aplicativos)
BD	Banco de Dados
CPU	<i>Central Processing Unit</i> (Unidade Central de Processamento)
DAO	<i>Data Access Object</i> (Objeto de Acesso a Dados)
DFWG	<i>Data Format Working Group</i>
DICOM	<i>Digital Imaging and Communication in Medicine</i> (Imagem Digital e Comunicação em Medicina)
E/S	Entrada e Saída
fMRI	<i>Functional Magnetic Resonance Imaging</i> (Imagem de Ressonância Magnética Funcional)
GUI	<i>Graphical User Interface</i> (Interface Gráfica do Usuário)
HMD	<i>Head Mounted Display</i>
IDE	Integrated Development Environment (Ambiente de Desenvolvimento Integrado)
JPG	<i>Joint Photographics Experts Group</i>
MRI	<i>Magnetic Resonance Imaging</i> (Imagem por Ressonância Magnética)
NIfTI	<i>Neuroimaging Informatics Technology Initiative</i>
PC	<i>Personal Computer</i> (Computador Pessoal)
PNG	<i>Portable Network Graphics</i>
RV	Realidade Virtual

SGBD	Sistema de Gerenciamento de Banco de Dados
SGPMIM	Sistema de Gerenciamento de Pacientes e Manipulação de Imagens Médicas
SQL	<i>Structured Query Language</i> (Linguagem de Consulta Estruturada)
UML	<i>Unified Modeling Language</i> (Linguagem de Modelagem Unificada)

SUMÁRIO

1 INTRODUÇÃO	9
1.1 Objetivo	9
1.2 Estrutura do Projeto	10
2 DETALHAMENTO DO PROBLEMA	11
3 SOLUÇÕES EXISTENTES	13
3.1 BrainVR	13
3.2 Laudo Web	14
3.3 Análise	16
4 ESPECIFICAÇÃO DO SISTEMA	17
5 IMPLEMENTAÇÃO DO SISTEMA	25
5.1 Tecnologias Utilizadas	25
5.1.1 Unity	25
5.1.2 HMD (<i>Head Mounted Display</i>)	25
5.1.3 Visual Studio	26
5.1.4 MySQL	27
5.1.5 MySQL Workbench	27
5.1.6 XAMPP	28
5.2 Arquitetura do SGPMIM	28
6 USO DO SISTEMA	34
7 CONCLUSÕES	45
REFERÊNCIAS	46

1 INTRODUÇÃO

A utilização de imagens médicas para realização de diagnósticos é algo comum há vários anos. De acordo com Ribeiro (2014), desde os primórdios o ser humano possui curiosidade sobre seu próprio corpo e as primeiras imagens anatômicas foram obtidas através de autópsias. Porém, com o advento da tecnologia, as imagens médicas tendem a ser minimamente invasivas, precisas e até mesmo interativas.

A visualização de imagens médicas de forma simples e completa ainda é um desafio, como informado por Ribeiro (2014), pois é realizada por *softwares* de processamento e análise de dados que muitas vezes não são intuitivos e nem mesmo multiplataformas.

Federizzi e Souza (2013) informam que as neuroimagens, obtidas com a tecnologia MRI (*Magnetic Resonance Imaging* – Imagem por Ressonância Magnética) possuem por padrão o formato NIfTI (*Neuroimaging Informatics Technology Initiative*) e que os sistemas utilizados para o processamento de imagens com este formato de arquivo não são, de modo geral, intuitivos.

Através da Realidade Virtual (RV) e da robótica é possível expandir os sentidos humanos. Segundo Ribeiro (2014), existem três conceitos interligados e responsáveis por essa expansão. O primeiro é a imersão, que decorre do envolvimento do indivíduo com o ambiente. O segundo é a navegação, onde é possível que o usuário navegue no ambiente virtual. O terceiro conceito é a interação, em que o utilizador pode controlar o mundo virtual de maneira intuitiva e simplificada.

Sendo assim, esta monografia apresenta uma proposta de solução quanto à visualização e manipulação de neuroimagens com o formato NIFTI-1, bem como o gerenciamento de informações de pacientes, visando superar o desafio de desenvolver uma ferramenta intuitiva através da utilização da Realidade Virtual e de um ambiente 3D, facilitando e auxiliando estudos médicos.

1.1 Objetivo

Esta monografia tem por objetivo o desenvolvimento de um sistema que facilitará o gerenciamento de informações de pacientes e a análise de neuroimagens obtidas com exame de Ressonância Magnética, as quais possuem a extensão .nii (NIFTI-1 *Data Format File*) e

poderão ser visualizadas em um computador ou através de óculos de Realidade Virtual, sendo vinculadas ou não a um dos pacientes cadastrados na aplicação.

1.2 Estrutura do Projeto

Esta monografia está dividida em capítulos. No Capítulo 2 o problema abordado será detalhado e seus requisitos especificados. No Capítulo 3 serão apresentadas algumas soluções já existentes para o problema em questão e suas diferenças em relação à solução proposta no presente trabalho. O Capítulo 4 apresentará a especificação do sistema, não somente na forma textual como também através de diagramas. A implementação do sistema será apresentada no Capítulo 5, expondo as tecnologias utilizadas e a arquitetura do SGPMIM. O Capítulo 6 mostrará o modo de utilização do sistema. Por fim, o Capítulo 7 apresentará as conclusões do projeto, expondo se o objetivo principal foi atingido, os desafios enfrentados e as propostas de melhorias futuras.

2 DETALHAMENTO DO PROBLEMA

Segundo Ribeiro (2014), a ressonância magnética é uma das tecnologias mais utilizadas para aquisição de imagens do corpo humano e a neurociência é a área onde esta técnica é especialmente importante. Vários *softwares* para processamento das neuroimagens foram desenvolvidos. Porém, para que os dados das imagens obtidas sejam visualizados é necessário um conhecimento aprofundado do funcionamento do *software*.

Tendo em vista esta situação, foi pensada na projeção de neuroimagens de modo a facilitar a visualização e a manipulação destas, criando um ambiente imersivo através da Realidade Virtual, tornando intuitiva a análise das imagens obtidas pela MRI e possibilitando visualizar os dados de pacientes vinculados a estas imagens.

De acordo com Federizzi e Souza (2013), a tecnologia MRI é uma das grandes descobertas que apoiam a medicina, em especial para:

[...] os diversos ramos da neurociência, que estudam tanto a estrutura fisiológica quanto as capacidades mentais mais complexas do ser humano como a linguagem e a autoconsciência, os exames de fMRI (Functional MRI) são uma importante ferramenta de apoio ao diagnóstico, permitindo aos especialistas um estudo aprofundado das patologias do sistema nervoso e do comportamento humano. Entre as vantagens desse exame, pode-se destacar a quantidade de informações que disponibiliza sobre o órgão em análise, a característica de ser um método não invasivo e o fato de não utilizar radiação para gerar as imagens. (FEDERIZZI E SOUZA 2013, p. 14).

Com o objetivo de padronizar o formato de arquivo utilizado por sistemas de processamento de fMRI (Imagem de Ressonância Magnética Funcional), foi desenvolvido pelo DFWG (*Data Format Working Group*) o formato de imagem NIFTI (*Neuroimaging Informatics Technology Initiative*), conforme informado por Federizzi e Souza (2013). Segundo Federizzi, esses sistemas necessitam de um estudo específico para serem utilizados. Assim sendo, o desenvolvimento de aplicações mais intuitivas pode auxiliar a comunidade médica tanto na formação de profissionais quanto na agilidade ao analisar neuroimagens.

Para que estes benefícios sejam atingidos, alguns requisitos devem ser respeitados, conforme citado por Netto, Machado e Oliveira (2002), tais como:

- A ferramenta desenvolvida deve possuir uma interface limpa e de fácil compreensão;
- Para imersão e aproveitamento da RV, o uso de um HMD (*Head Mounted Display*) será necessário; e
- Durante a visualização da MRI no ambiente virtual não deve haver travamentos, os quais podem prejudicar a imersão do utilizador.

3 SOLUÇÕES EXISTENTES

Neste capítulo serão apresentados dois projetos que possuem objetivos próximos ao trabalho proposto. O primeiro apresenta o desenvolvimento de uma interface de visualização de conectividade cerebral, nomeada de BrainVR, utilizando Realidade Virtual e controle por gesto (RIBEIRO, 2014) e suas principais características serão mostradas na seção 3.1. O segundo exibe o desenvolvimento do sistema Laudo Web (CARITÁ, 2002) cujos detalhes serão mostrados na seção 3.2.

3.1 BrainVR

Este trabalho foi desenvolvido por Ricardo Filipe Henriques Ribeiro em 2014 e tem por objetivo principal o desenvolvimento de uma interface que permite a visualização de dados de imagens obtidas através de ressonância magnética tanto em um monitor de computador quanto em um sistema de Realidade Virtual, sendo bastante próximo do objetivo do presente trabalho.

A aplicação foi desenvolvida utilizando a plataforma Unity e dividida em seis opções distintas, conforme pode ser visualizado na Figura 1.

Figura 1 – Menu da Aplicação BrainVR



Fonte: Ribeiro, 2014, p. 45

A primeira opção do menu, conforme descrito por Ribeiro (2014), exhibe ao usuário a interface de visualização de imagem estrutural. A segunda e terceira exibem uma interface de conectividade estrutural, com dados estatísticos. A quarta opção realiza o teste de conexão do sistema com o *Leap Motion*, tecnologia que possibilita a navegação por gestos no ambiente virtual através de uma aplicação chamada *Touchless*, a qual substitui o mouse de computador por um mouse virtual. A quinta alternativa apresenta informações do projeto e a sexta finaliza a execução do sistema.

3.2 Laudo Web

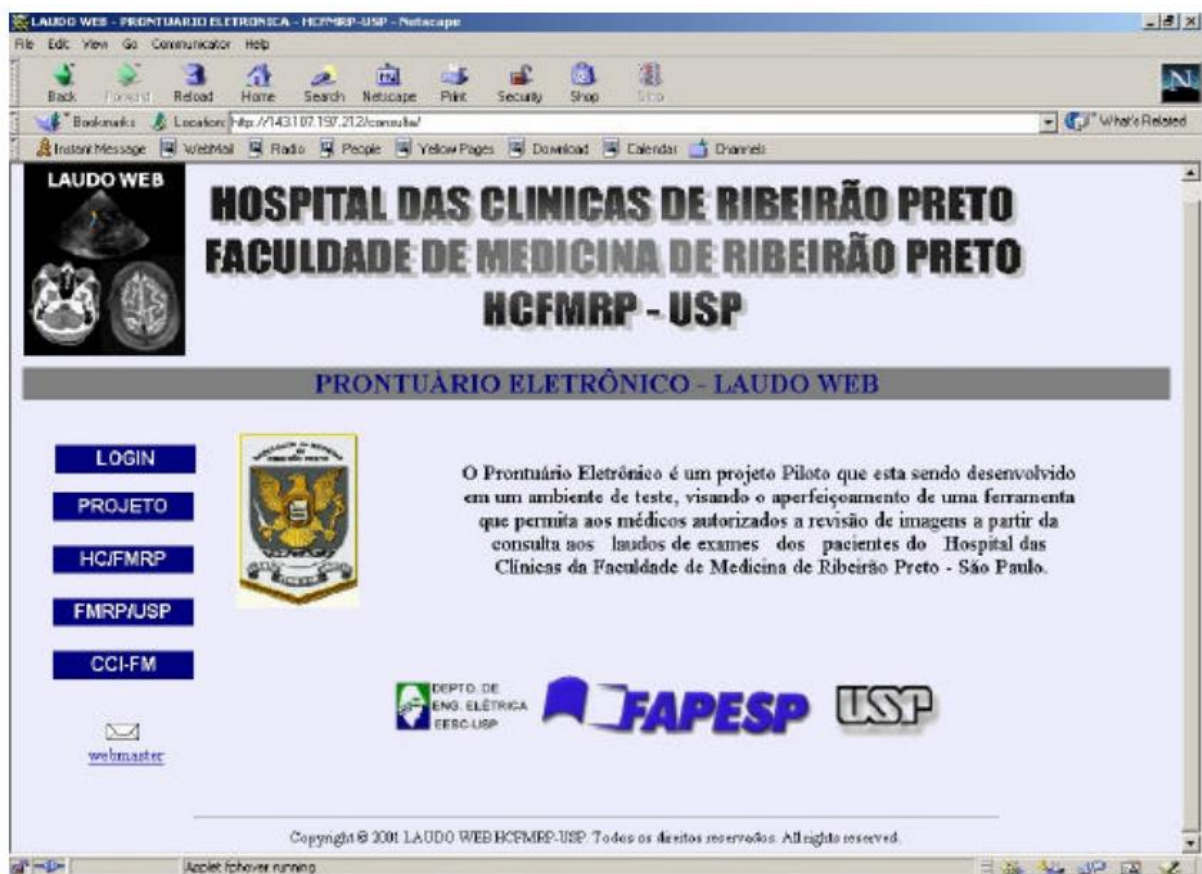
Esta aplicação foi desenvolvida por Edilson Carlos Caritá em 2002. O principal objetivo de seu projeto consiste em desenvolver um sistema para vinculação, recuperação e

disponibilização de imagens médicas obtidas por radiologia em computadores conectados à rede *ethernet*, conforme descrito pelo autor (CARITÁ, 2002).

O sistema desenvolvido, nomeado pelo autor de Laudo Web, possui uma base de dados visando o gerenciamento de imagens no padrão DICOM 3.0 (*Digital Imaging and Communication in Medicine*), o qual possibilita a transferência de imagens e informações associadas a estas para o banco de dados. Arquivos do padrão DICOM possuem informações como seu identificador único, nome do paciente, data de realização do exame, idade, entre outras.

A tela inicial do Laudo Web, sistema desenvolvido para ser executado em *browsers*, pode ser visualizada na Figura 2.

Figura 2 – Tela Inicial do Laudo Web



Fonte: Caritá, 2002, p. 74

É permitido ao usuário autenticado pelo sistema a consulta de dados, exames e imagens médicas de tomografia computadorizada e ressonância magnética dos pacientes, inseridos no banco de dados do programa.

3.3 Análise

Os dois sistemas apresentados se aproximam da solução proposta no presente projeto. Porém, a ferramenta desenvolvida por Ribeiro (2014) é a mais próxima do objetivo do SGPMIM, por se tratar de visualização de imagens médicas, em especial de ressonâncias magnéticas, em ambiente virtual, desenvolvido através da multiplataforma Unity.

Entre as diferenças do software a ser desenvolvido e o apresentado por Caritá (2002) está o fato de o sistema deste ser executado em um *browser*, fazendo o uso da rede *Ethernet*. Outra distinção é a apresentação das imagens médicas ser realizada por um computador sem utilização da Realidade Virtual. Pode-se citar ainda o formato utilizado pelas imagens, por se tratar do padrão DICOM, enquanto o padrão estudado neste projeto é o NIfTI-1.

Apesar da proximidade, há várias diferenças entre o software desenvolvido e o de Ribeiro. Entre elas, destaca-se o fato de o SGPMIM possibilitar o cadastro e gerenciamento de dados, exames e histórico de imagens de pacientes específicos. Além disso, na dissertação de Ribeiro há um estudo realizado sobre a conectividade cerebral, parte da neurociência que não será abordada neste trabalho.

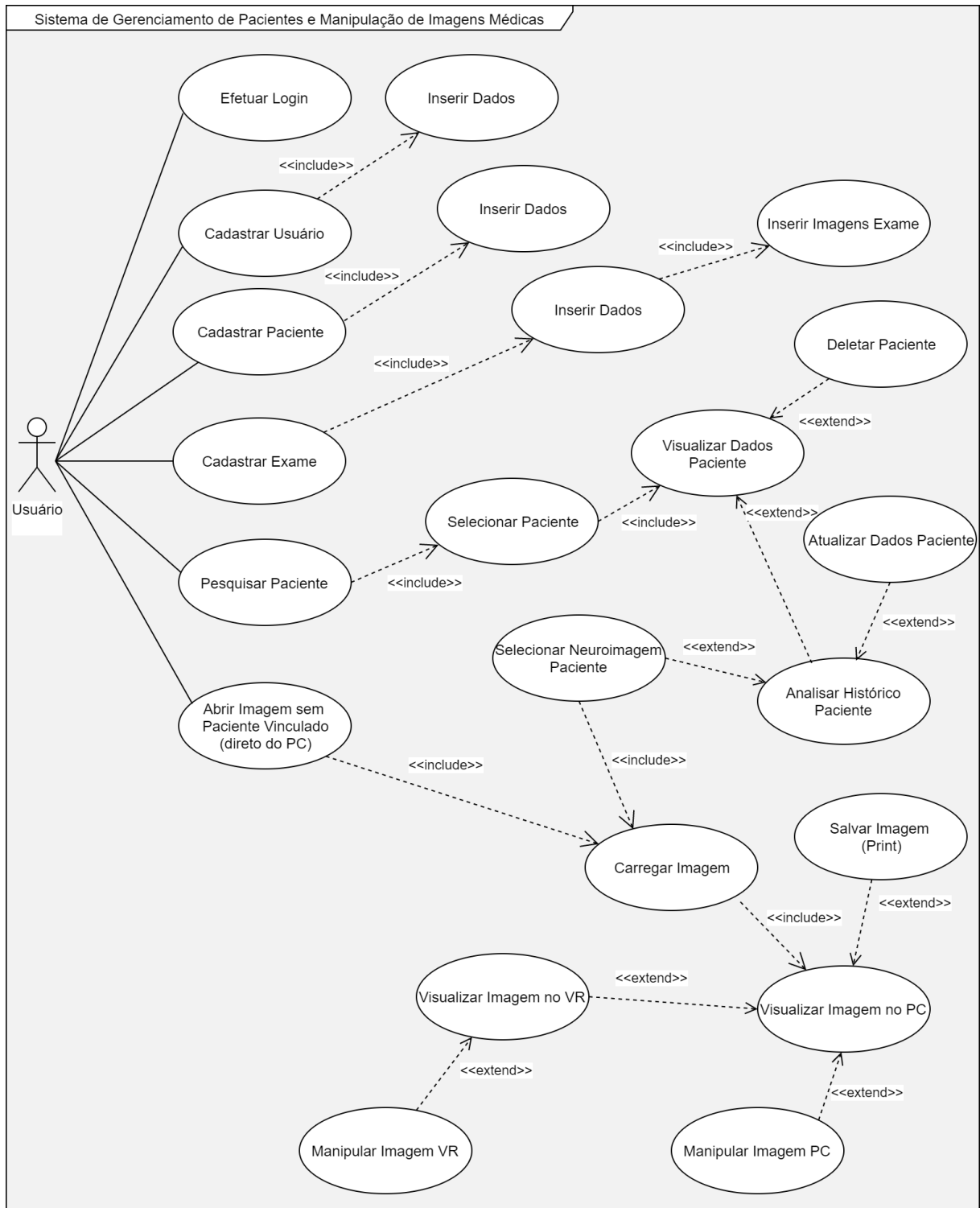
4 ESPECIFICAÇÃO DO SISTEMA

Com a finalidade de representar de maneira detalhada o funcionamento do sistema, foram criados diagramas UML e um Diagrama de Entidade e Relacionamento utilizando a ferramenta Draw.io (DRAW.IO, 2016).

Na Figura 3 é apresentado o diagrama de caso de uso, cujos principais casos são: “Efetuar Login”, “Cadastrar Usuário”, “Cadastrar Paciente”, “Cadastrar Exame”, “Pesquisar Paciente” e “Abrir Imagem Sem Paciente Vinculado”. Estes constituem o menu inicial que é exibido ao usuário. Especificamente, as funcionalidades de cada um destes casos são:

- “Efetuar Login”: o primeiro caso de uso possibilita a inserção do nome de usuário e senha para que seja realizado o *login* e liberado o acesso ao sistema pelo utilizador.
- “Cadastrar Usuário”: com o segundo caso de uso é possível a realização de cadastros de outros usuários para acesso à aplicação.
- “Cadastrar Paciente”: o terceiro caso de uso permite o cadastramento de dados e informações de pacientes no banco de dados do sistema.
- “Cadastrar Exame”: com o quarto caso de uso é possível a realização do cadastro de exames de pacientes e do caminho das imagens médicas destes para que estas sejam visualizadas e manipuladas no sistema.
- “Pesquisar Paciente”: a quinta opção do menu possibilita ao usuário a busca e seleção de determinado paciente. Em seguida, serão visualizadas suas informações cadastradas e será possível excluí-lo ou analisar seu histórico, onde pode-se comparar imagens médicas obtidas em diferentes exames, atualizar dados e ainda selecionar uma imagem específica do paciente para ser carregada no sistema. Caso esta seja uma neuroimagem e possua o formato NIFTI-1, poderá ser visualizada e manipulada no computador e através de óculos de realidade virtual. No PC, será possível salvá-la da maneira que está mostrada no monitor. Caso a imagem possua o formato PNG ou JPG, poderá ser visualizada de maneira estática no sistema.
- “Abrir Imagem sem Paciente Vinculado”: o sexto caso de uso permite ao usuário o carregamento no sistema de uma neuroimagem que não está vinculada a nenhum paciente. Esta também poderá ser visualizada e manipulada tanto no computador

Figura 3 – Diagrama de Caso de Uso



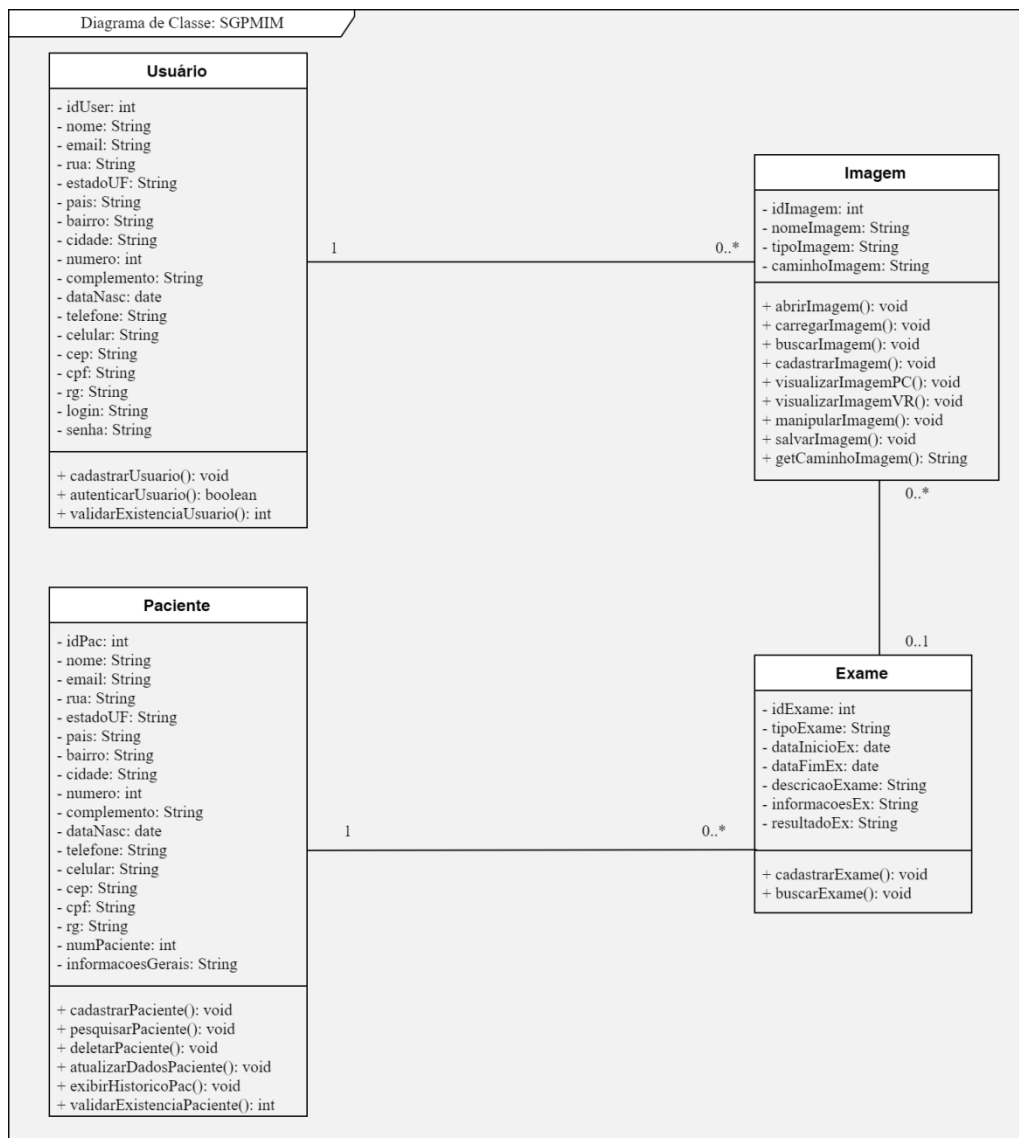
Fonte: Autoria Própria

quanto no ambiente virtual, para que o usuário possua uma experiência imersiva e possa manusear as imagens, como rotacioná-las, efetuar cortes (opção disponível somente no

computador) e ver pontos específicos, em um ambiente 3D. Além disso, é possível salvar a imagem da maneira que está sendo exibida no monitor em formato PNG no computador utilizado.

A Figura 4 exibe o Diagrama de Classe do sistema, onde é possível visualizar, de maneira geral, o funcionamento das principais classes dos *scripts* desenvolvidos no decorrer do projeto. Visando facilitar o desenvolvimento do sistema no ambiente Unity, alguns métodos destas classes foram criados em *scripts* separados.

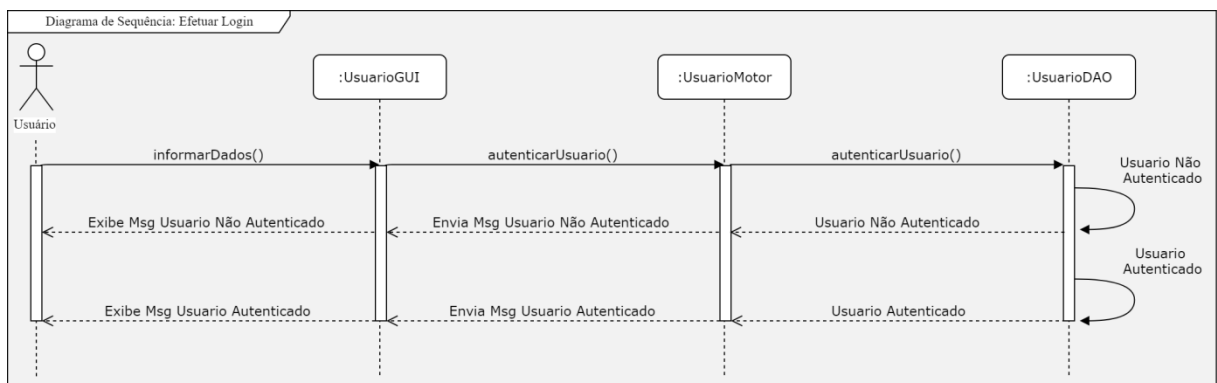
Figura 4 – Diagrama de Classe



Fonte: Autoria Própria

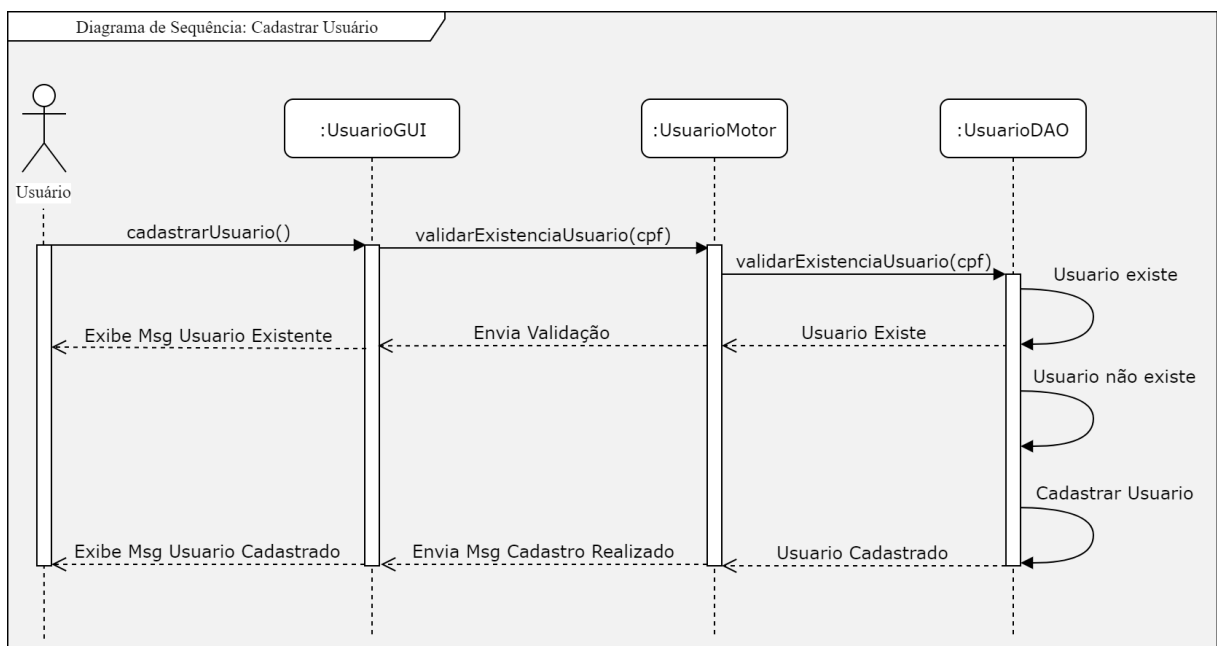
Nas Figuras 5 a 10 são mostrados os Diagramas de Sequência, com os quais se pode visualizar dinamicamente o funcionamento dos principais casos de uso e do caso “Selecionar Neuroimagem Paciente” (Figura 10), com o qual é possível visualizar a chamada dos métodos para a abertura de neuroimagens no computador.

Figura 5 – Diagrama de Sequência: Efetuar Login



Fonte: Autoria Própria

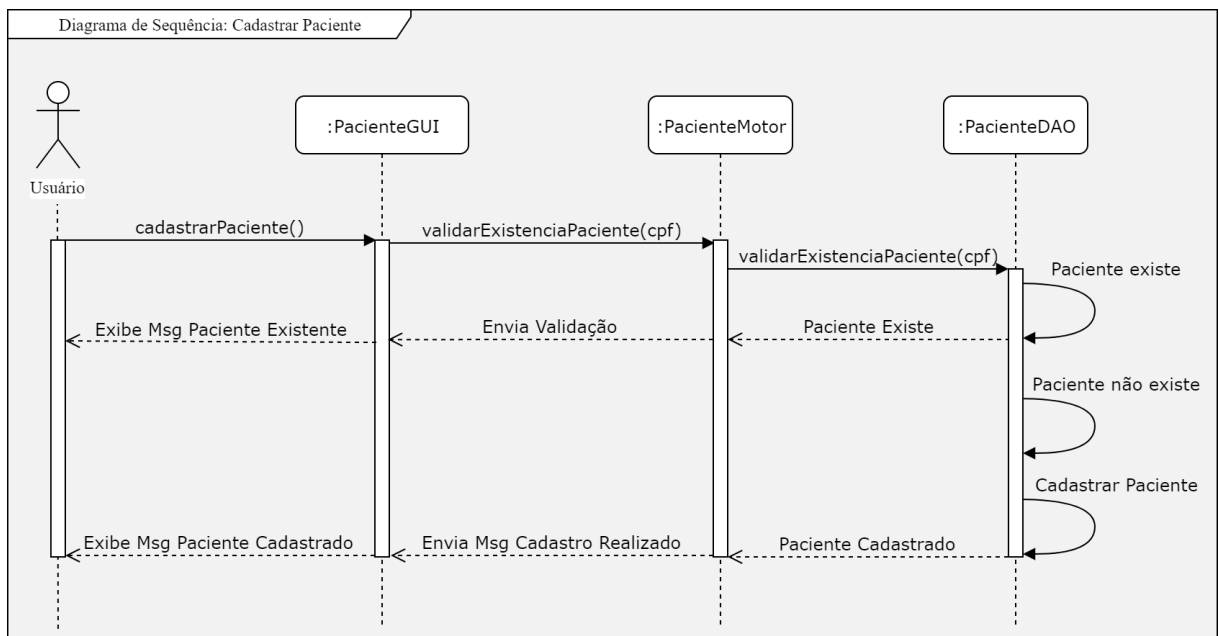
Figura 6 – Diagrama de Sequência: Cadastrar Usuário



Fonte: Autoria Própria

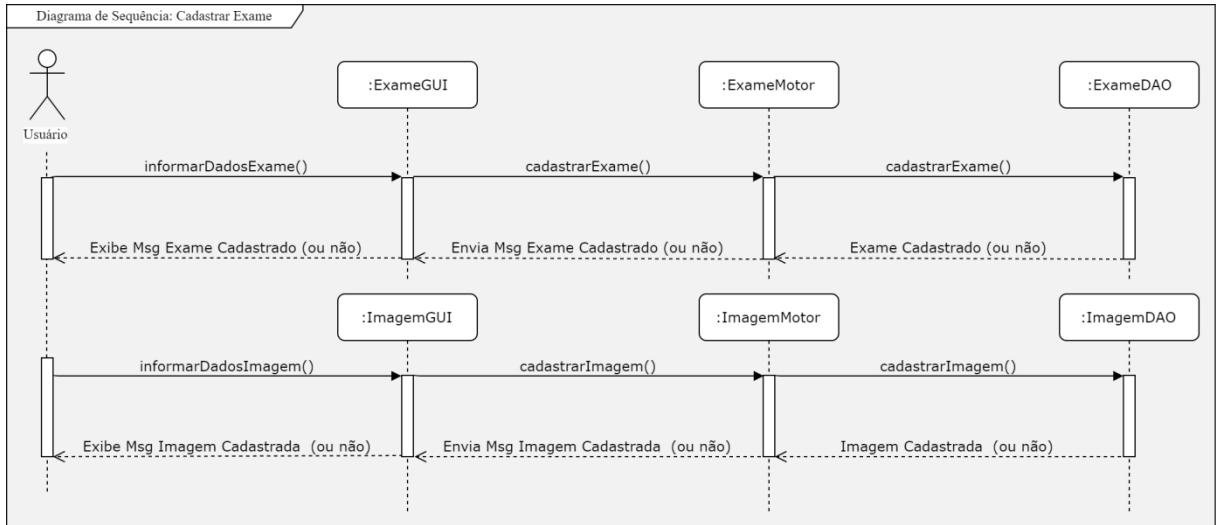
A Figura 7 demonstra a maneira como o cadastro de um paciente é realizado pelo sistema. Pode-se observar que há a interação de três componentes: “PacienteGUI”, “PacienteMotor” e “PacienteDAO”. O primeiro representa a interface gráfica mostrada ao usuário. O segundo contém os métodos necessários e interliga o “PacienteGUI” com o “PacienteDAO”, o qual é formado por *scripts* em PHP responsáveis pelo cadastro de informações do paciente no banco de dados do sistema. Este diagrama é semelhante ao exibido na Figura 6.

Figura 7 – Diagrama de Sequência: Cadastrar Paciente



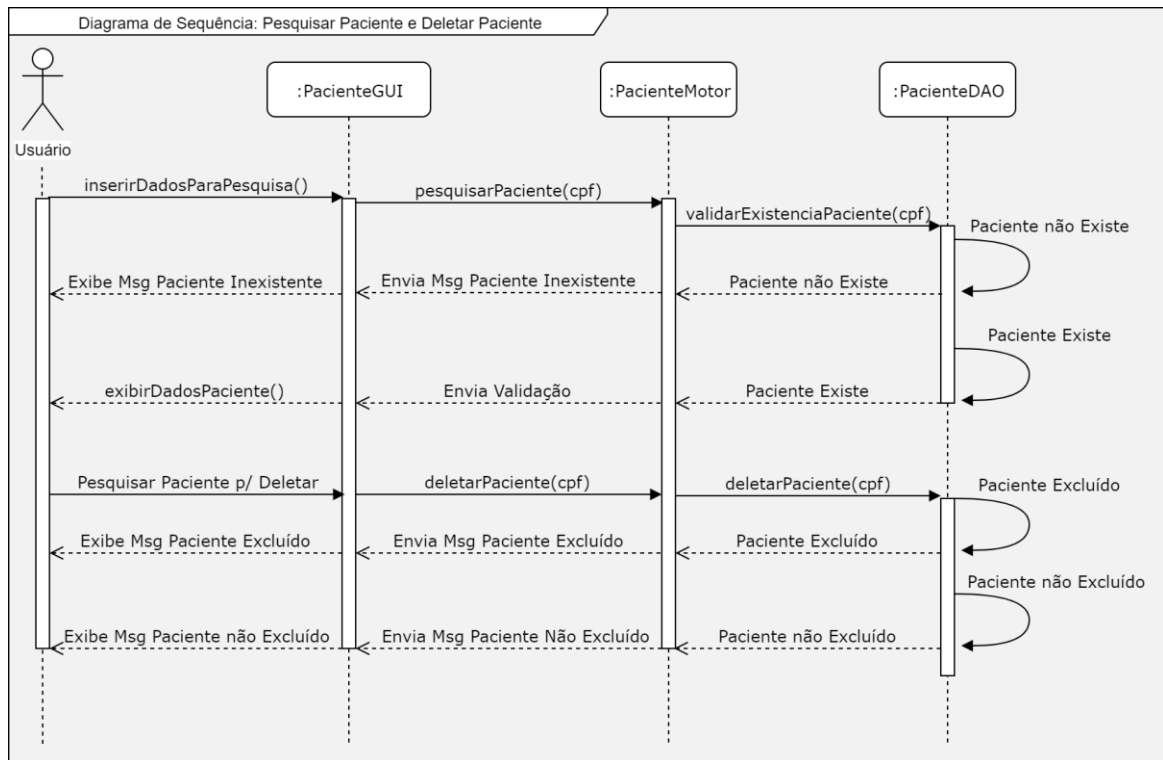
Fonte: Autoria Própria

Figura 8 – Diagrama de Sequência: Cadastrar Exame



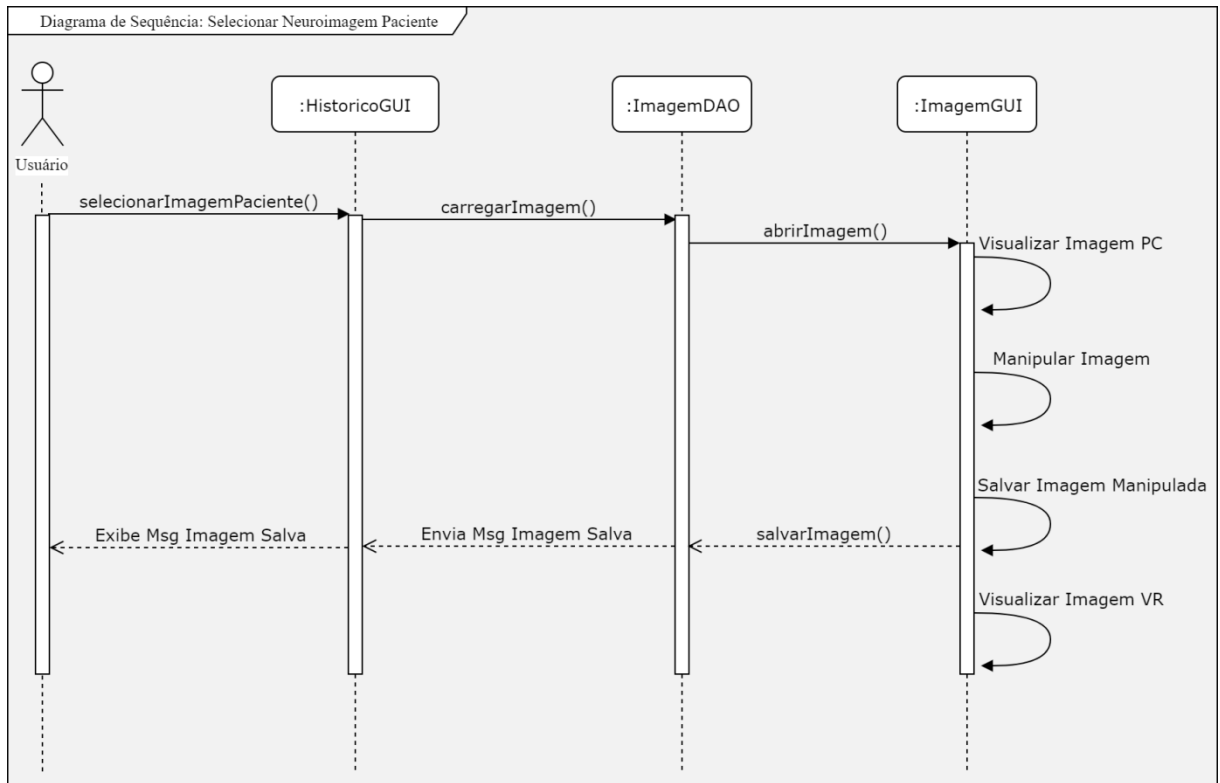
Fonte: Autoria Própria

Figura 9 – Diagrama de Sequência: Pesquisar Paciente



Fonte: Autoria Própria

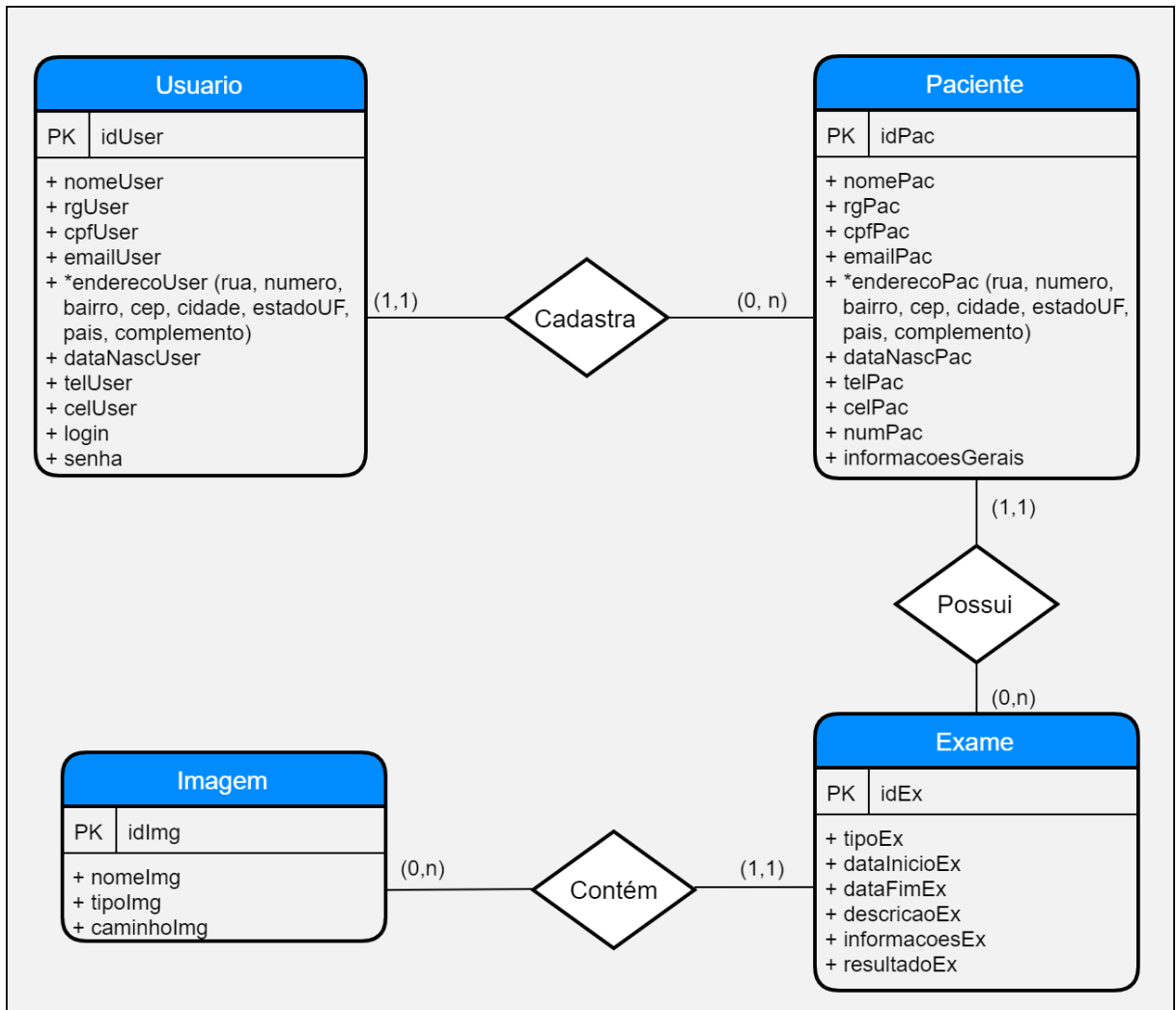
Figura 10 – Diagrama de Sequência: Selecionar Neuroimagem Paciente



Fonte: Autoria Própria

As entidades, que representam as tabelas que foram criadas no banco de dados do SGPMIM, bem como seus atributos e relacionamentos, podem ser visualizadas na Figura 11.

Figura 11 – Diagrama de Entidade e Relacionamento



Fonte: Autoria Própria

A quantidade de diagramas apresentada é vasta. Porém, estes facilitam o entendimento quanto ao funcionamento interno do sistema e também foram essenciais durante seu desenvolvimento.

5 IMPLEMENTAÇÃO DO SISTEMA

Neste capítulo serão apresentadas as tecnologias utilizadas e a arquitetura do sistema SGPMIM.

5.1 Tecnologias Utilizadas

As principais tecnologias utilizadas durante o desenvolvimento do SGPMIM serão detalhadas nos itens 5.1.1 a 5.1.6.

5.1.1 Unity

Desenvolvido em 2005 pela organização *Unity Technologies*, o Unity é uma multiplataforma cujo principal foco é o desenvolvimento de jogos. Este software pode ser executado nos sistemas operacionais Windows e Mac OS (RIBEIRO, 2014).

Esta aplicação, segundo sua organização (TECHNOLOGIES, 2019a), possui uma *Engine* de alto desempenho e pode ser utilizado com a IDE Visual Studio (TECHNOLOGIES, 2019b). Ela também permite o desenvolvimento de jogos e ferramentas em 2D e 3D para múltiplas plataformas, como *desktop* e dispositivos móveis, por exemplo. No total, o Unity suporta mais de 25 plataformas (TECHNOLOGIES, 2019a).

Nesta multiplataforma podem ser programados *scripts* em linguagens de programação de alto nível, como o C# e o JavaScript, por exemplo (RIBEIRO, 2014). Para o desenvolvimento deste projeto foi utilizada a linguagem C# e a versão 5.3.5 do Unity.

5.1.2 HMD (*Head Mounted Display*)

O HMD pode ser descrito como um dispositivo em formato de capacete que possibilita a visualização de imagens através de uma tela posicionada na frente dos olhos do usuário (PEREIRA, 2018).

Conforme informado por Pereira (2018), existem, atualmente, dispositivos com valores, capacidades e funcionalidades diferentes. Entre os de alto custo, pode-se citar o Vive

VR e o Oculus Rift. Já dentre os de baixo custo, estão o Google Cardboard e o VR Box, os quais precisam da incorporação de um *joystick* e de um *smartphone* para a interação e apresentação de imagens em um ambiente 3D (PEREIRA, 2018).

No presente projeto foi utilizado o VR Box, que pode ser visualizado na Figura 12. Porém, por questões de incompatibilidade durante o desenvolvimento, não foi incorporado um *joystick* ao sistema.

Figura 12 – HMD VR Box



Fonte: Pereira, 2018, p. 7

5.1.3 Visual Studio

O Visual Studio é uma IDE (*Integrated Development Environment* ou Ambiente de Desenvolvimento Integrado) da Microsoft, que, conforme mencionado em sua documentação, pode ser definido como um painel de inicialização criativo que pode ser utilizado para editar, depurar, compilar códigos e publicar aplicativos (MICROSOFT, 2019a).

Dentre os recursos disponíveis que auxiliam na produtividade, podem ser citados, segundo a Microsoft (2019a):

- Rabiscos: sublinhados ondulados que avisam sobre erros ou potenciais problemas no código. São mostrados durante a digitação e, caso sejam focalizados, exibem informações adicionais sobre o possível erro;
- Limpeza de Código: recurso atualmente disponível apenas para C# que possibilita a formatação e correção do código de acordo com configurações pré-definidas;

- Refatoração: possibilita a renomeação inteligente de variáveis, extração de linhas de códigos em um novo método e alteração da ordem de parâmetros presentes em métodos; e
- *IntelliSense*: conjunto de recursos que mostram informações diretamente no editor sobre o código, podendo até mesmo escrever pequenos trechos de códigos.

Esta IDE pode ser utilizada nos sistemas operacionais Windows e Mac e possui as edições *Community*, *Professional* e *Enterprise*. A edição utilizada neste projeto será a *Community*, cujo download pode ser realizado gratuitamente (MICROSOFT, 2019b).

5.1.4 MySQL

O MySQL é um Sistema de Gerenciamento de Banco de Dados (SGBD) de código aberto que, atualmente, é o mais conhecido no mundo, segundo a Oracle (2019b), empresa que o mantém, desenvolve e distribui. Ele utiliza o modelo relacional, onde os dados são armazenados em tabelas separadas ao invés de em apenas uma tabela. (ORACLE, 2019b)

As letras "SQL" do nome "MySQL" significa *Structured Query Language* (Linguagem de Consulta Estruturada), a linguagem padronizada utilizada para acessar bancos de dados. É possível, dependendo do ambiente de programação usado, inserir o SQL diretamente, para gerar relatórios, por exemplo, incorporar instruções SQL no código escrito em outro idioma ou utilizar uma API específica do idioma que oculte a sintaxe SQL (ORACLE, 2019b).

O servidor de banco de dados MySQL é escalável e pode ser executado em um *desktop* ou notebook juntamente com outros aplicativos ou servidores *web*, mas também pode ser configurado para executar em uma máquina totalmente dedicada a ele, aproveitando toda a memória, CPU e capacidade de E/S disponíveis. O *MySQL Server* funciona tanto em sistemas embarcados quanto em cliente/servidor (ORACLE, 2019b).

5.1.5 MySQL Workbench

O MySQL Workbench é uma ferramenta gráfica para bancos de dados e servidores MySQL. Ele suporta as versões 5.6 e posteriores do *MySQL Server* e também é compatível

com as versões 5.x, exceto em alguns casos, por conta de alterações em tabelas do sistema. Não oferece suporte para as versões 4.x (ORACLE, 2019a).

De acordo com a Oracle (2019a), dentre as funcionalidades do MySQL Workbench estão:

- A criação e gerenciamento de conexões com servidores de bancos de dados e execução de consultas SQL nas conexões utilizando o Editor SQL integrado;
- A criação de modelos de esquema do banco de dados de forma gráfica e edição de todos os aspectos do banco utilizando o Editor de Tabelas; e
- A administração de instâncias do servidor MySQL que permite o gerenciamento de usuários, execução de *backup* e recuperação, inspeção de dados de auditoria, visualização da integridade do banco de dados e monitoramento do desempenho do servidor MySQL.

5.1.6 XAMPP

O XAMPP é uma distribuição do servidor *web* Apache que contém também os seguintes softwares livres: MariaDB, PHP e Perl (FRIENDS, 2019b). Seu objetivo é construir uma distribuição de fácil instalação para que os desenvolvedores ingressem no mundo Apache (FRIENDS, 2019a).

Esta distribuição é mantida pelo Apache Friends, um projeto sem fins lucrativos que visa promover o servidor *web* Apache (FRIENDS, 2019b). Segundo o site oficial do Apache Friends (2019b), o XAMPP existe há mais de 10 anos.

5.2 Arquitetura do SGPMIM

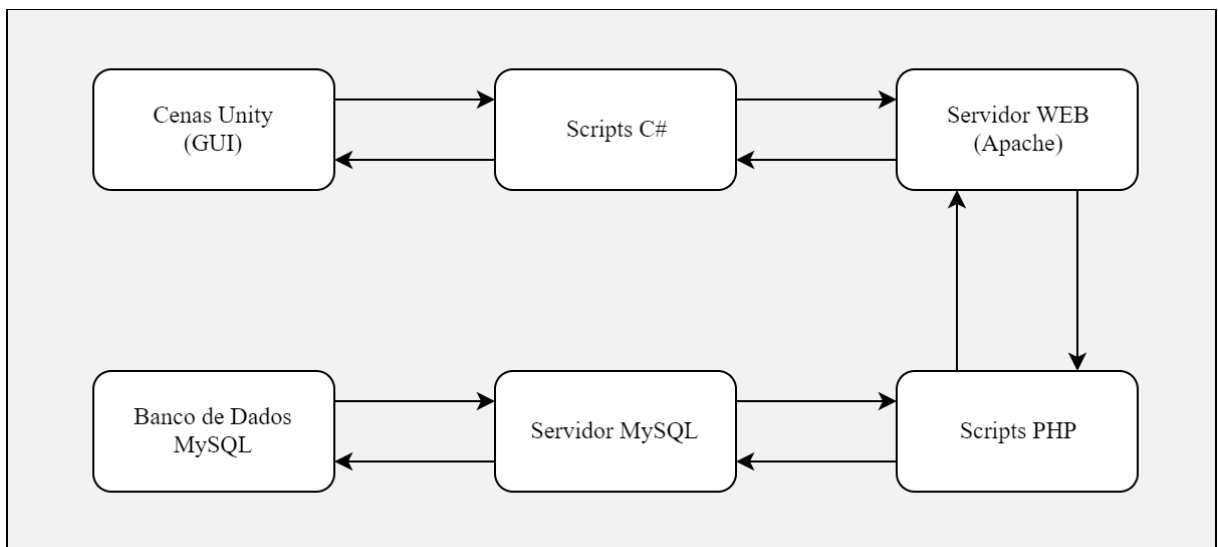
Os principais componentes responsáveis pelo funcionamento do SGPMIM são:

- Cenas criadas na Unity, que correspondem à interface gráfica apresentada ao usuário durante a utilização do sistema;
- *Scripts* na linguagem C#, desenvolvidos no Visual Studio e responsáveis pelo funcionamento da parte gráfica do sistema;

- Servidor Apache, iniciado através do painel de controle do XAMPP e responsável pela comunicação dos *scripts* em PHP com o BD;
- *Scripts* desenvolvidos na linguagem de *script* PHP (GROUP, 2019), responsáveis pelo acesso ao BD;
- Servidor MySQL; e
- Banco de Dados MySQL, manipulado através do MySQL Workbench.

Na Figura 13 pode-se visualizar o funcionamento do SGPMIM a partir da ligação entre estes seis componentes.

Figura 13 – Representação do Funcionamento do SGPMIM



Fonte: Autoria Própria

O primeiro componente é composto por dezesseis cenas no total, também chamadas de telas. Toda interação do usuário com o sistema acontecerá através delas. As principais telas estão apresentadas no Capítulo 6.

Os *scripts* em C# compõem o segundo componente. Eles são os responsáveis pelo funcionamento da parte gráfica e a comunicação do sistema com os *scripts* em PHP.

Foram utilizados três *assets* em conjunto com as cenas e os *scripts* desenvolvidos pela autora do presente projeto. O principal destes é o Volume Viewer Pro (LISCINTEC, 2017), o

qual é o responsável por renderizar as imagens com formato NIfTI-1 e também disponibilizar as opções de manipulação de imagens.

O Volume Viewer Pro não possibilita que seja informado pelo usuário ou por um registro do banco de dados o caminho de uma imagem a ser renderizada. Sendo assim, após leitura e entendimento dos *scripts* em C# e da documentação disponibilizados pelo *asset*, foi criado um *script* chamado “Dados” com uma variável estática denominada “path” que atribui o caminho da imagem desejada à variável “_dataPath” do *script* “VolumeFileLoader” (fornecido pelo *asset*) no método “Start”. Assim, é possível carregar no sistema uma neuroimagem a partir do caminho informado por determinado registro no banco de dados ou por um usuário.

O segundo *asset* utilizado chama-se Standalone File Browser (GKNGKC, 2018) e permite que o usuário navegue pelo seu computador e selecione diretórios ou arquivos de determinados formatos. Na Figura 14 pode-se visualizar o método “selecionarDiretorio()”, que utiliza a função “OpenFolderPanel()” disponibilizada pelo Standalone File Browser.

Figura 14 – Método “selecionarDiretorio()” do SGPMIM

```
public void selecionarDiretorio()
{
    caminhoPrint = StandaloneFileBrowser.OpenFolderPanel("Select Folder", "", false);

    Dados._path = String.Join("",
        new List<string>(caminhoPrint)
            .ConvertAll(i => i.ToString())
            .ToArray());
    //Debug.Log("Caminho Print é: " + Dados._path);
}
```

Fonte: Autoria Própria

O terceiro é o Trinus VR (SHEEP, 2019), o qual possibilita visualizar imagens em Realidade Virtual pelo celular, sendo necessário somente a instalação do Trinus CBVR ou Trinus CBRV Lite no telefone móvel e um HMD de baixo custo para complementar a experiência do usuário, como o VRBox ou Google Cardboard, por exemplo.

Para acoplar o Trinus VR ao SGPMIM foi utilizado o *prefab* “SampleIntegratedUIManager” que, conforme descrito na documentação do *asset*, possibilita a integração das funções disponíveis com o sistema desenvolvido, como por exemplo a função “openIntro()”, que inicia a interface padrão do Trinus VR de conexão do sistema com o celular.

As funções disponibilizadas por estes *assets* foram acopladas ao SGPMIM após leitura da documentação de cada um e entendimento de sua lógica.

Dentre os principais *scripts* em C#, desenvolvidos no decorrer do projeto, estão: “PacienteMotor”, “UsuarioMotor”, “ImagemMotor” e “ExameMotor”. Na Figura 15 observa-se uma parte do método “Cadastrar()” do *script* “PacienteMotor”, responsável por inserir no banco os dados de determinado paciente informados pelo usuário. Nele, é utilizada a classe WWWForm que, conforme a documentação do Unity (TECHNOLOGIES, 2019c), é uma classe auxiliar que possibilita a geração de formulários de dados e o envio destes à servidores *web*.

Figura 15 – Método “Cadastrar()” do *script* “PacienteMotor”

```
IEnumerator Cadastrar()
{
    WWWForm form = new WWWForm();

    form.AddField("nomePac", nomePac.text);
    form.AddField("rgPac", rgPac.text);
    form.AddField("cpfPac", cpfPac.text);
    form.AddField("emailPac", emailPac.text);
    form.AddField("rua", ruaPac.text);
    form.AddField("numero", numMoradiaPac.text);
    form.AddField("bairro", bairroPac.text);
    form.AddField("cep", cepPac.text);
    form.AddField("cidade", cidadePac.text);
    form.AddField("estadoUF", UFPac.text);
    form.AddField("pais", paisPac.text);
    form.AddField("complemento", complementoPac.text);
    form.AddField("dataNascPac", nascimentoPac.text);
    form.AddField("telPac", telPac.text);
    form.AddField("celPac", celPac.text);
    form.AddField("numPac", numPac.text);
    form.AddField("informacoesGerais", infoPac.text);
    form.AddField("idUser", idUser.text);

    WWW www = new WWW(url, form);
    yield return www;
}
```

Fonte: Autoria Própria

O “Servidor Web (Apache)” constitui o terceiro componente do sistema, presente na Figura 13, o qual é o responsável por possibilitar a comunicação dos *scripts* em C# com os *scripts* em PHP. Estes últimos compõem o quarto componente do sistema e possuem comandos escritos na linguagem SQL, os quais correspondem às solicitações realizadas pelos usuários do SGPMIM.

Na Figura 16 pode-se visualizar uma parte do *script* em PHP denominado “cadPac”, responsável por cadastrar novos pacientes no banco de dados.

Figura 16 – Script “cadPac” do SGPMIM

```
$sql="INSERT INTO paciente(nomePac, rgPac, cpfPac, emailPac, rua, numero, bairro, cep, cidade, estadoUF, pais, complemento, dataNascPac, telPac, celPac, numPac, informacoesGerais,idUser) VALUES ('$nomePac','$rgPac','$cpfPac', '$emailPac','$rua','$numero','$bairro', '$cep','$cidade','$estadoUF','$pais','$complemento','$dataNascPac', '$telPac','$celPac','$numPac','$informacoesGerais','$idUser');"

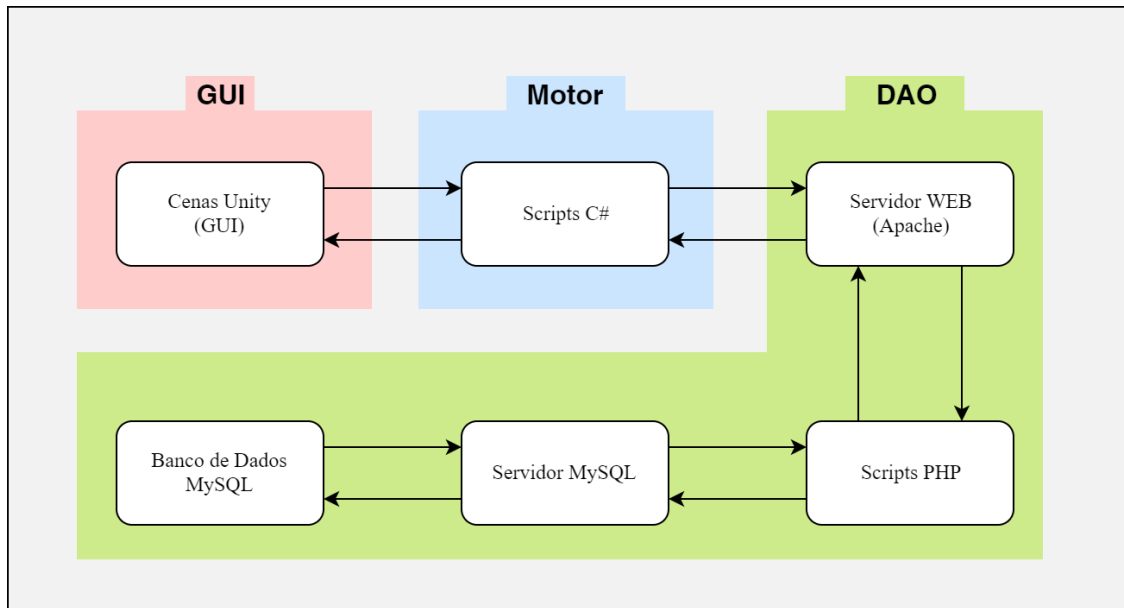
mysqli_query($con, $sql) or die ("Error: Insert Paciente query failed.");
```

Fonte: Autoria Própria

Há ainda o quinto componente, o servidor MySQL, que é o responsável por permitir a comunicação dos *scripts* em PHP com o banco de dados MySQL, o qual constitui o sexto e último componente do SGPMIM.

Visualizando de maneira mais geral, o SGPMIM pode ser dividido em três partes (Figura 17):

- GUI (*Graphical User Interface* ou Interface Gráfica do Usuário): nesta categoria encontram-se as telas criadas no Unity.
- Motor (funcionamento interno): localizam-se, nesta parte, os *scripts* na linguagem de programação C#.
- DAO (*Data Access Object* ou Objeto de Acesso a Dados): neste módulo encontram-se o banco de dados e servidor MySQL, servidor *web* Apache e os *scripts* em PHP, que possuem comandos SQL responsáveis por inserir, pesquisar, excluir e alterar informações do BD.

Figura 17 – Divisão Geral do SGPMIM

Fonte: Autoria Própria

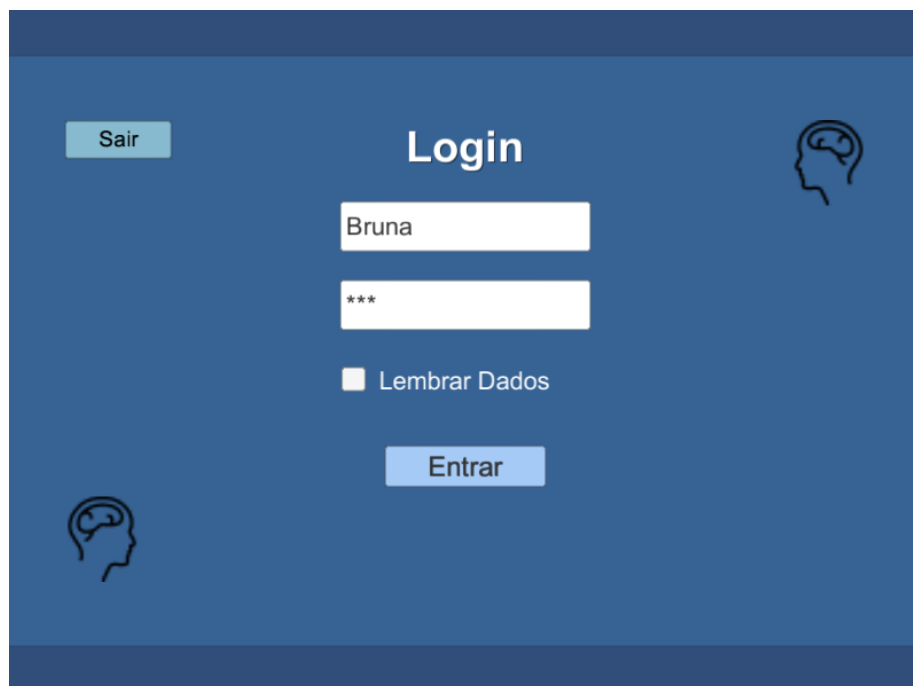
Esta divisão foi utilizada no decorrer do projeto visando facilitar o desenvolvimento e a comunicação entre os componentes, já que cada parte é responsável por determinada função e possui conceitos específicos.

6 USO DO SISTEMA

Antes de utilizar o sistema, é essencial, para desfrutar do módulo de Realidade Virtual, que o usuário realize a instalação do Trinus CBVR ou Trinus CBVR Lite em seu celular, possua internet *wi-fi* disponível conectado no aparelho e um HMD de baixo custo, como o VRBox, por exemplo.

Ao executar o sistema, o utilizador se depara, primeiramente, com a interface de “Login” (Figura 18), onde precisa inserir seu nome de usuário e senha cadastrados previamente no banco de dados.

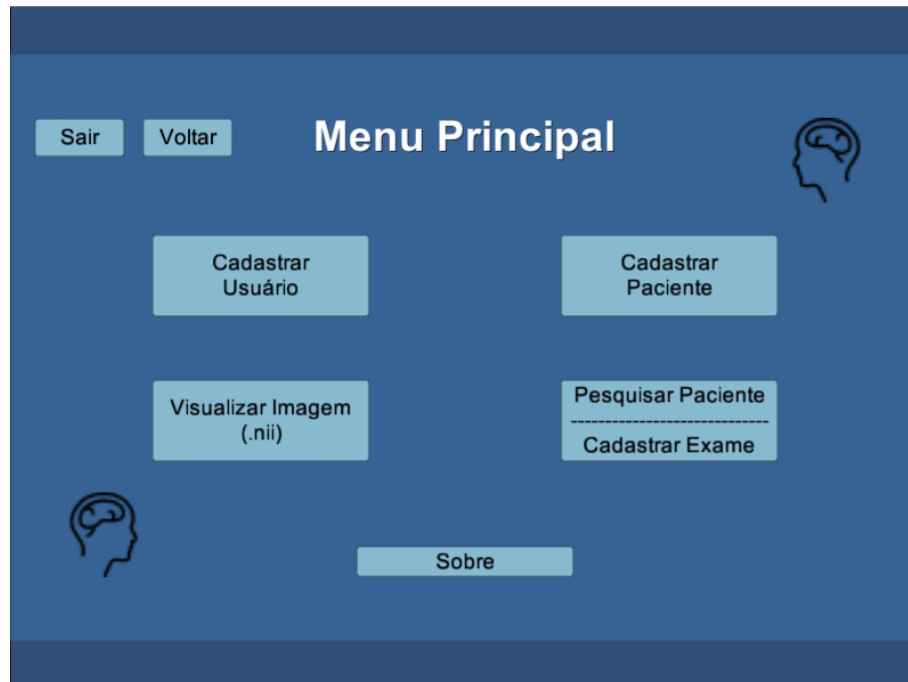
Figura 18 – Tela “Login”



A imagem mostra a interface de login do sistema. O fundo é azul escuro. No topo esquerdo, há um botão "Sair" em um retângulo azul claro. No topo direito, há um ícone de uma cabeça humana com um cérebro. No centro, o título "Login" está em branco. Abaixo dele, há dois campos de entrada de texto brancos: o primeiro contém o nome "Bruna" e o segundo contém "***". Abaixo dos campos, há uma opção "Lembrar Dados" com uma caixa de seleção vazia. No centro inferior, há um botão "Entrar" em um retângulo azul claro. No canto inferior esquerdo, há um ícone de uma cabeça humana com um cérebro.

Fonte: Autoria Própria

A Figura 19 apresenta a interface do menu principal do SGPMIM, exibida ao usuário após o *login* ser realizado. Esta possui os botões “Sair”, “Voltar”, “Sobre”, “Cadastrar Usuário”, “Cadastrar Paciente”, “Pesquisar Paciente/Cadastrar Exame” e “Visualizar Imagem (.nii)”.

Figura 19 – Menu Principal do SGPMIM

Fonte: Autoria Própria

Os botões “Voltar” e “Sair” possibilitam voltar à tela “Login” e sair do sistema, respectivamente. Já o botão “Sobre” apresenta informações referentes à desenvolvedora do projeto e seu orientador.

Ao selecionar o botão “Cadastrar Usuário”, o utilizador é redirecionado à tela exibida na Figura 20, onde pode inserir novos usuários e suas informações no banco de dados.

O botão “Cadastrar Paciente” exibe uma tela semelhante à de cadastro de usuários, mas com alguns campos diferenciados, conforme pode ser visualizado na Figura 21.

Figura 20 – Tela “Cadastro de Usuário”

Cadastro de Usuário

[Voltar](#)

*Nome: *Login:

*RG: *Senha:

*CPF:

E-mail:

Rua: Nº: Bairro:

CEP: Cidade: Complemento:

UF: País: Nascimento:

Tel.: Cel.:

[Cadastrar](#)

Fonte: Autoria Própria

Figura 21 – Tela “Cadastro de Paciente”

Cadastro de Paciente

[Voltar](#)

*Nome: Nº Paciente:

*RG: Info. Gerais:

*CPF:

E-mail:

Rua: Nº: Bairro:

CEP: Cidade: Complemento:

UF: País: Nascimento:

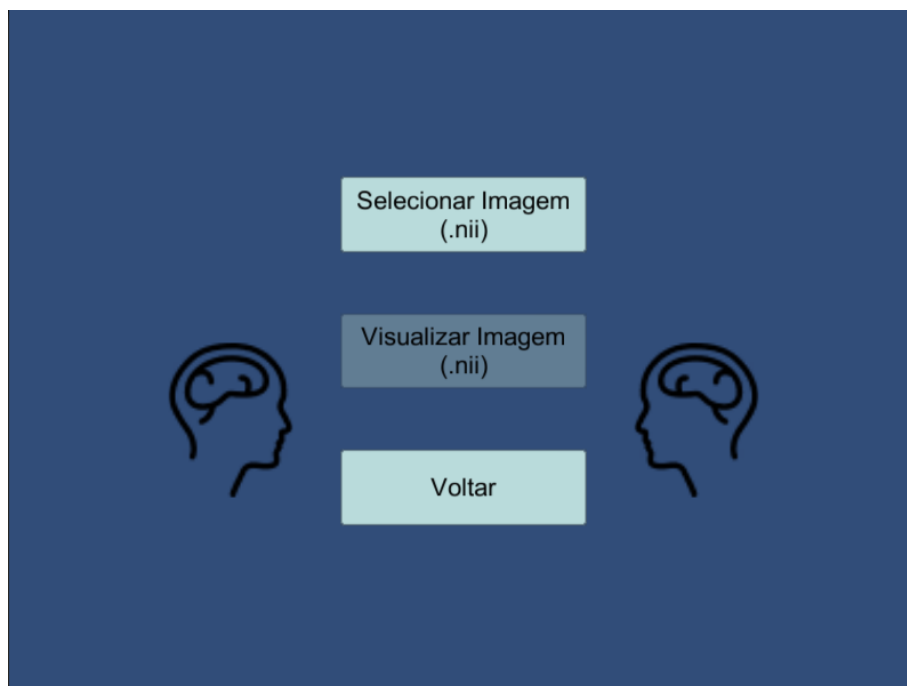
Tel.: Cel.: *ID usuário: 2

[Cadastrar](#)

Fonte: Autoria Própria

A Figura 22 mostra o menu de seleção de imagem NIfTI da aplicação, exibido após a seleção do botão “Visualizar Imagem (.nii)”. Nesta tela, são exibidos os botões “Selecionar Imagem (.nii)”, “Visualizar Imagem (.nii)” e “Voltar”. O segundo botão apenas é habilitado quando uma imagem com o formato NIfTI-1 é selecionada no primeiro botão, a qual deve estar armazenada no computador do usuário. O terceiro botão possibilita voltar ao menu principal.

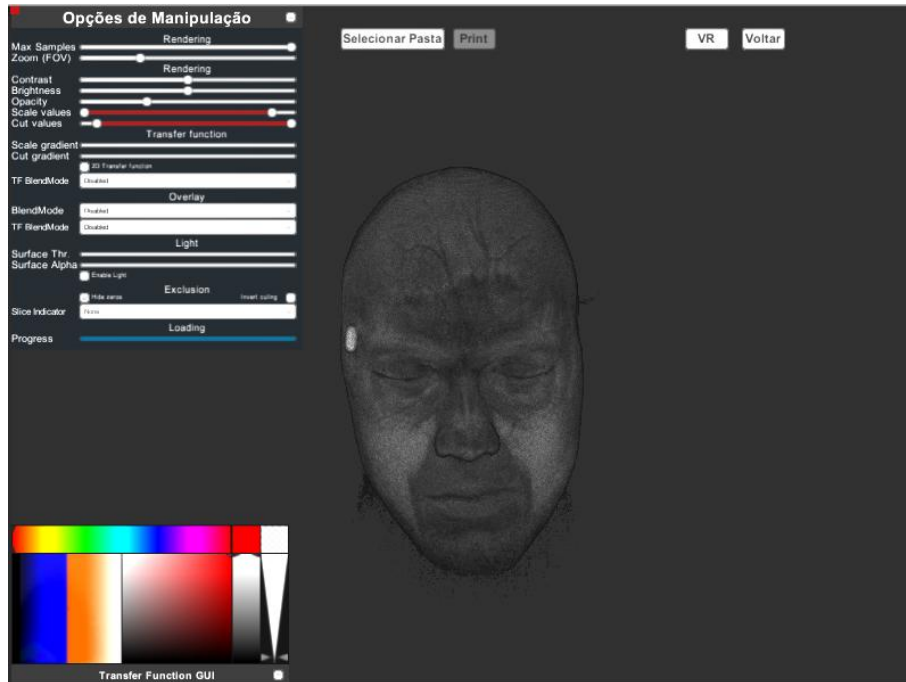
Figura 22 – Menu de Seleção de Imagem NIfTI-1



Fonte: Autoria Própria

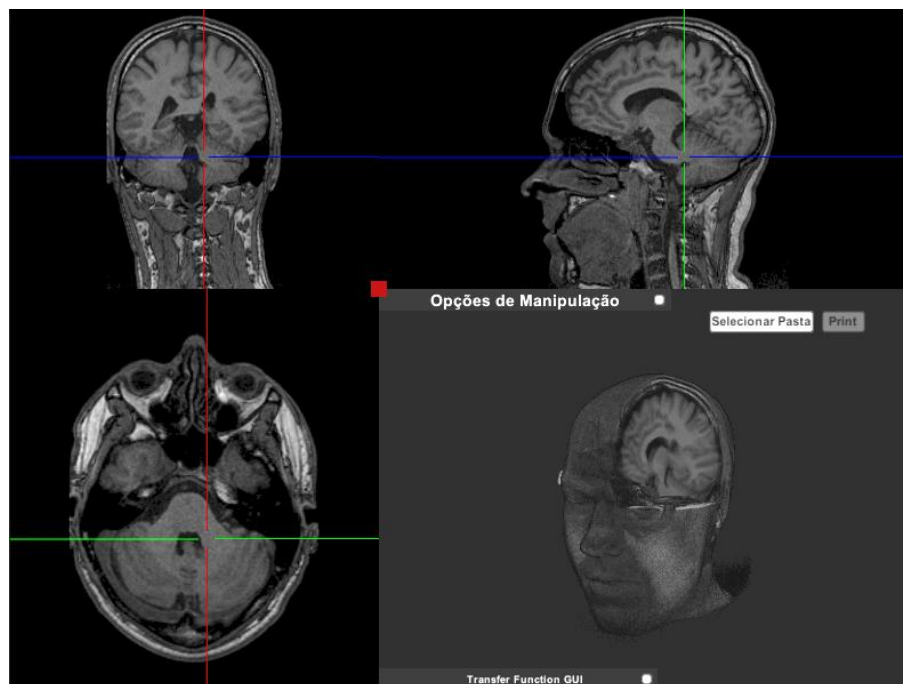
Na Figura 23 é apresentada ao usuário a interface exibida após a seleção da neuroimagem. O carregamento desta pode ser observado através do progresso da barra “Loading” e, após a conclusão deste, a imagem com formato NIfTI-1 é exibida ao usuário e este pode utilizar as opções disponíveis para manipulá-la. Uma dessas opções consiste em recortar uma parte da neuroimagem e visualizar seus diferentes pontos, como mostrado na Figura 24.

Figura 23 – Interface de Manipulação de Neuroimagem



Fonte: Autoria Própria

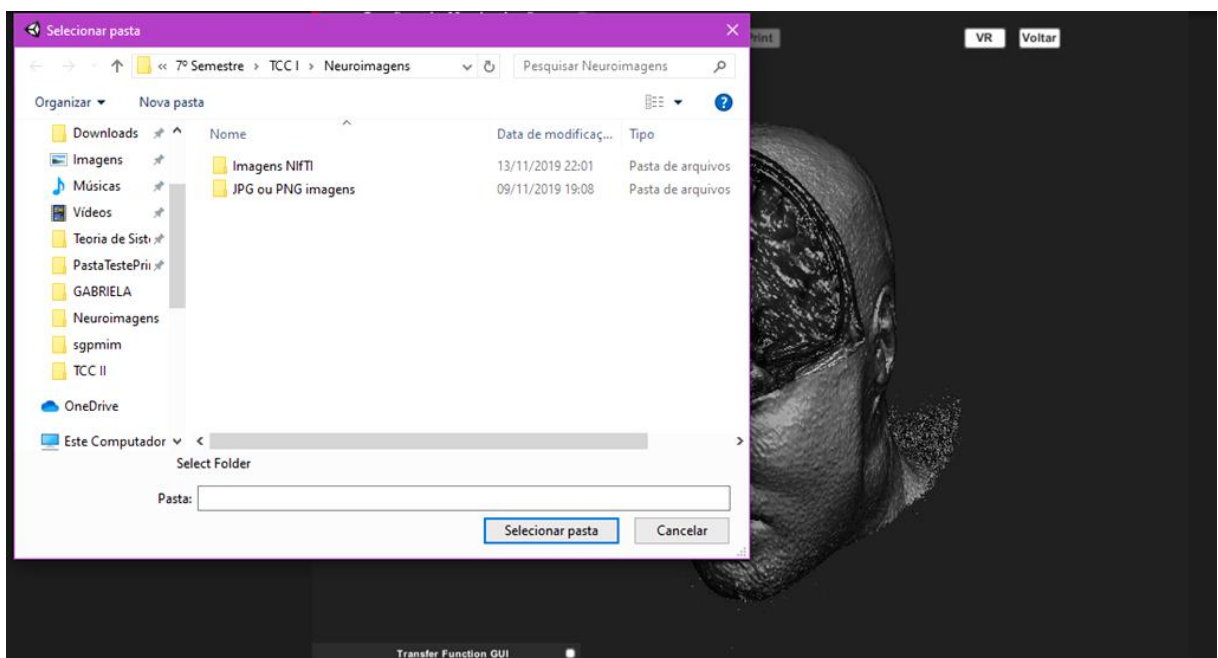
Figura 24 – Visualização de Diferentes Pontos da Neuroimagem



Fonte: Autoria Própria

Observa-se também que na interface (Figura 23) são disponibilizados quatro botões: “Selecionar Pasta”, “Print”, “VR” e “Voltar”. O primeiro botão, como o nome informa, possibilita a seleção de um diretório, como apresentado na Figura 25, onde poderão ser armazenados *prints* do sistema no formato PNG quando clicado no botão “Print”, o qual é habilitado após a escolha da pasta. O botão VR possibilita que o usuário visualize a neuroimagem em Realidade Virtual e o botão “Voltar” retorna ao menu principal do sistema.

Figura 25 – Seleção da Pasta de Armazenamento de Prints



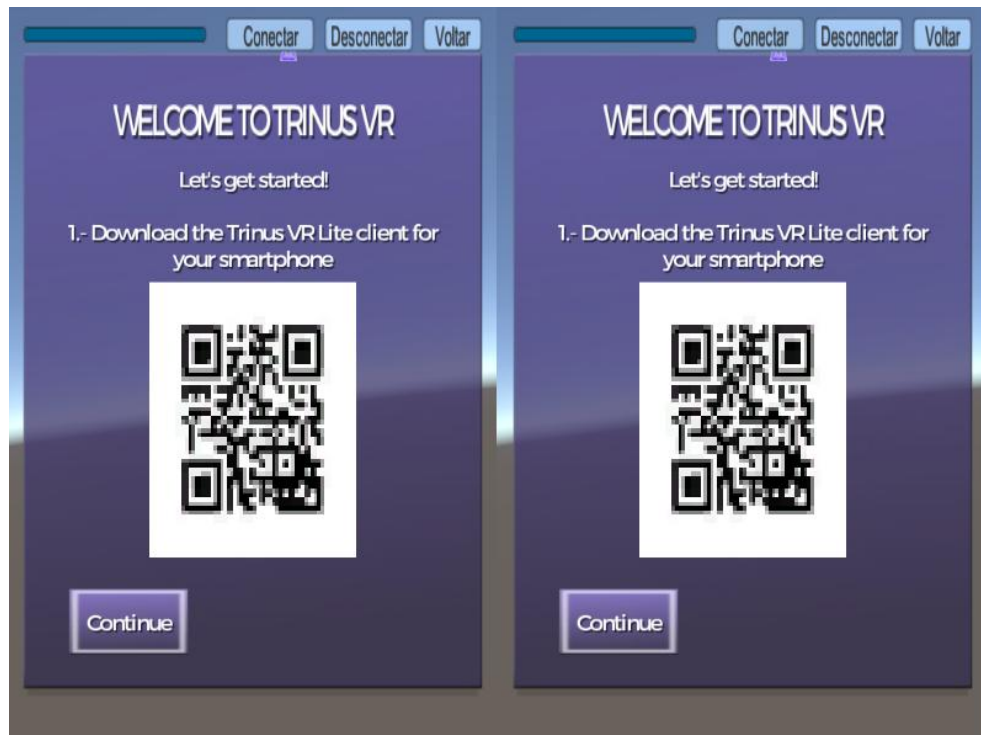
Fonte: Autoria Própria

A Figura 26 mostra a tela exibida ao selecionar o botão “VR” e clicar em “Iniciar”. Após seguir as recomendações sugeridas pela interface padrão do Trinus VR, o usuário poderá visualizar a neuroimagem em seu celular de vários ângulos (Figura 27) e, com o auxílio do HMD VRBox ou outro semelhante, imergir no ambiente virtual.

A recomendação mais importante para efetivar a conexão do sistema com o Trinus CBVR, instalado no celular, é conectar um cabo USB no computador e no aparelho móvel e ativar a opção “Conexão por USB”, também conhecida como “*Tethering* USB”. Caso o usuário deseje, outra forma possível é conectar um cabo de rede no PC em que o SGPMIM

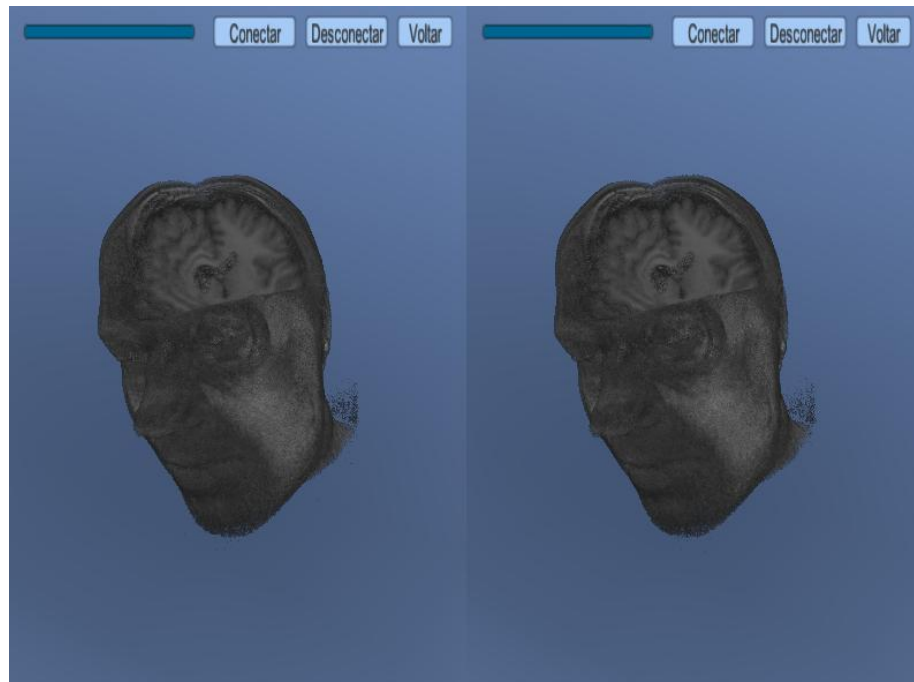
está instalado e conectar o aparelho móvel na mesma rede, via *wi-fi*. Porém, conectar tanto o computador quanto o celular em uma rede *wi-fi* fará com que ocorra perda de desempenho do sistema e, por conta disso, essa forma de conexão não é recomendada.

Figura 26 – Interface Padrão Trinus VR Para Conexão



Fonte: Autoria Própria

Figura 27 – Neuroimagem Renderizada em Realidade Virtual



Fonte: Autoria Própria

A tela mostrada na Figura 27 possui três botões que podem ser acionados pelo usuário no computador durante a exibição da neuroimagem em Realidade Virtual. O botão “Conectar” possibilita realizar a conexão do sistema com o celular novamente, caso ocorra algum empecilho durante a utilização do sistema. Já o botão “Desconectar” encerra a conexão com o celular e o “Voltar” redireciona ao menu principal.

No menu, o usuário pode selecionar também o botão “Pesquisar Paciente\Cadastrar Exame” e o sistema exibirá a tela “Pesquisar Paciente”, apresentada na Figura 28.

Figura 28 – Tela “Pesquisar Paciente”

Pesquisar Paciente

CPF do paciente ...

Nome	RG	CPF	Email	Rua
[Empty table body]				

Fonte: Autoria Própria

Esta tela possui os botões “Buscar”, “Deletar”, “Cadastrar Exame”, “Histórico Paciente” e “Voltar.” O primeiro irá exibir um paciente e seus dados, de acordo com o CPF digitado pelo usuário. O segundo botão permite excluir do banco de dados o paciente mostrado na tela. A opção “Cadastrar Exame” exibe a tela de inserção de exames no banco relacionados ao paciente digitado, conforme mostrado na Figura 29. Após um exame ser cadastrado, é aberta a tela de “Cadastro de Imagem” (Figura 30), na qual é possível cadastrar todas as imagens, nos formatos NIfTI-1, PNG e JPG, que estejam relacionadas ao exame e, caso deseje, o usuário também pode inserir um novo exame.

Figura 29 – Tela “Cadastro de Exame”

Cadastro de Exame

Voltar

*Tipo de exame: Informações:

*Data início exame: *Resultado:

*Data fim exame: *Id Paciente: 14

Descrição:

Cadastrar

Fonte: Autoria Própria

Figura 30 – Interface de Cadastro de Imagem

Voltar Novo Exame

Exame

Tipo de exame: Exame de Sangue Id Exame: 41 Descrição:

Data início exame: 09092019 Resultado: Nenhuma

Data fim exame: 09092019 Id Paciente: 14 Informações:

Cadastro de Imagem

*Nome Imagem: *Caminho Imagem: Selecionar

*Tipo Imagem: JPG/PNG *Id Exame: 41

Cadastrar

Fonte: Autoria Própria

Na tela “Pesquisar Paciente” (Figura 28) também é possível visualizar o histórico do registro exibido através do botão “Histórico Paciente”. Ao selecioná-lo, o usuário é redirecionado para a interface mostrada na Figura 31.

Figura 31 – Tela “Histórico - Paciente”

Histórico - Paciente

Informações Paciente

Id Paciente:	Nome:	País:
Num. Pac:	E-mail:	Complem.:
RG:	Rua:	Info. Gerais:
CPF:	Número:	
Id Usuário:	Bairro:	
Nascimento:	CEP:	
Telefone:	Cidade:	
Celular:	Estado:	

Exame - Paciente

Tipo de exame:	Id Exame:	Descrição:
Data início exame:	Resultado:	

Fonte: Autoria Própria

A tela “Histórico - Paciente” proporciona ao usuário uma visão ampla dos dados do paciente selecionado, possibilitando, através do botão “Editar”, a atualização destes e a visualização de todos os seus exames e das imagens referentes a eles. As neuroimagens serão exibidas conforme a Figura 23, podendo ser manipuladas e visualizadas no ambiente virtual. Já as imagens com formato PNG ou JPG serão exibidas em uma nova tela, de maneira estática. Além disso, também é possível cadastrar novas imagens referentes aos exames mostrados.

São várias as telas que compõem o sistema, mas, apesar disso, todos os componentes exibem explicitamente suas funções, visando tornar a experiência do usuário mais interativa e intuitiva.

7 CONCLUSÕES

O principal objetivo do projeto, o desenvolvimento de uma ferramenta que possibilite o gerenciamento de informações de pacientes e a manipulação de imagens médicas com o formato NIfTI-1, foi atingido. O sistema permite ao usuário, de forma simples e intuitiva, o cadastro, a busca, alteração e exclusão de dados de pacientes, além de proporcionar a inserção e visualização de exames e imagens relacionados a este.

Foram realizados testes de “Caixa Preta”, ou seja, testes funcionais (NETO, 2015), com 25 neuroimagens. Todas foram renderizadas e puderam ser manipuladas no sistema. Além disso, foi possível visualizar todas no ambiente de Realidade Virtual. Os testes de “Caixa Branca”, que visam identificar problemas na estrutura do código (NETO, 2015), foram efetuados no decorrer do desenvolvimento.

Um dos principais desafios enfrentados pela autora durante a construção do SGPMIM foi o aprimoramento de seus conhecimentos na linguagem PHP. Além disso, pode-se citar também o entendimento da documentação e dos *scripts* em C# disponibilizados pelos *assets*, visando mesclá-los com o sistema desenvolvido.

Sugere-se, como uma das melhorias futuras possíveis, a implementação de opções mais variadas de manipulação de neuroimagens no ambiente de Realidade Virtual, as quais poderiam ser controladas através de um *joystick*, por exemplo. Além disso, o sistema também poderia permitir a criação de tipos específicos de usuários, restringindo o acesso a algumas funcionalidades dependendo do perfil criado. Por fim, outra melhoria seria a possibilidade de armazenar imagens com os formatos PNG, JPG e NIfTI-1 no próprio banco de dados do SGPMIM, sem que ocorra a perda de desempenho do sistema.

REFERÊNCIAS

CARITÁ, Edilson Carlos. **Vinculação de imagens para busca e visualização a partir de sistema de informação em radiologia (RIS)**. 2002. Disponível em: <http://www.teses.usp.br/teses/disponiveis/18/18133/tde-19032008-090917/publico/Edilson_Dissertacao_mt.pdf>. Acesso em: 25/03/19.

DRAW.IO. **Diagrams with draw.io for Confluence: Our New User Documentation Is Online**. 2016. Disponível em: <<https://about.draw.io/diagrams-with-draw-io-for-confluence-our-new-user-documentation-is-online/>>. Acesso em: 26/06/19.

FEDERIZZI, Márcia; SOUZA, Vanessa Borba De. **Processamento digital de imagens para avaliação qualitativa em diagnósticos por imagens de ressonância magnética nuclear**. 2013. Disponível em: <<http://revistaseletronicas.pucrs.br/ojs/index.php/graduacao/article/view/17829/0>>. Acesso em: 25/03/19.

FRIENDS, Apache. **About**. 2019a. Disponível em: <<https://www.apachefriends.org/about.html>>. Acesso em: 07/11/19.

FRIENDS, Apache. **XAMPP Apache + MariaDB + PHP + Perl**. 2019b. Disponível em: <<https://www.apachefriends.org/index.html>>. Acesso em: 07/11/19.

GKNGKC. **Unity Standalone File Browser**. 2018. Disponível em: <<https://github.com/gkngkc/UnityStandaloneFileBrowser>>. Acesso em: 05/11/19.

GROUP, The PHP. **O que é o PHP?** 2019. Disponível em: <https://www.php.net/manual/pt_BR/intro-what-is.php>. Acesso em: 07/11/19.

LISCINTEC. **Volume Viewer Pro.** 2017. Disponível em: <<https://assetstore.unity.com/packages/vfx/shaders/fullscreen-camera-effects/volume-viewer-pro-83185>>. Acesso em: 25/03/19.

MICROSOFT. **Bem-vindo ao IDE do Visual Studio.** 2019a. Disponível em: <<https://docs.microsoft.com/pt-br/visualstudio/get-started/visual-studio-ide?view=vs-2019>>. Acesso em: 25/06/19.

MICROSOFT. **Visual Studio Community.** 2019b. Disponível em: <<https://visualstudio.microsoft.com/pt-br/vs/community/?rr=https%3A%2F%2Fwww.google.com%2F>>. Acesso em: 25/06/19.

NETO, Arilo Cláudio Dias. **Introdução a Teste de Software.** 2015. Disponível em: <https://edisciplinas.usp.br/pluginfile.php/3503764/mod_resource/content/3/Introducao_a_Teste_de_Software.pdf>. Acesso em: 26/03/19.

NETTO, Antonio Valerio; MACHADO, Liliane dos Santos; OLIVEIRA, Maria Cristina Ferreira De. **Realidade Virtual – Definições, Dispositivos e Aplicações.** 2002. Disponível em: <https://www.researchgate.net/publication/252019436_Realidade_Virtual_-_Definicoes_Dispositivos_e_Aplicacoes>. Acesso em: 24/06/19.

ORACLE. **General Information.** 2019a. Disponível em: <https://docs.oracle.com/cd/E17952_01/workbench-en/wb-intro.html>. Acesso em: 25/06/19.

ORACLE. **What is MySQL?** 2019b. Disponível em: <https://docs.oracle.com/cd/E17952_01/mysql-8.0-en/what-is-mysql.html>. Acesso em: 25/06/19.

PEREIRA, Miriam de Mesquita. **Realidade Virtual Aplicada ao Processo de Recrutamento**. 2018. Disponível em: <<https://iconline.ipleiria.pt/handle/10400.8/3794>>. Acesso em: 04/10/19.

RIBEIRO, Ricardo Filipe Henriques. **Desenvolvimento de uma interface de visualização de conectividade cerebral usando realidade virtual e controlo por gestos**. 2014. Disponível em: <https://run.unl.pt/bitstream/10362/14752/1/Ribeiro_2014.pdf>. Acesso em: 25/03/19.

SHEEP, Odd. **Trinus Virtual Reality**. 2019. Disponível em: <<https://www.trinusvirtualreality.com/>>. Acesso em: 07/11/19.

TECHNOLOGIES, Unity. **Unity 2019: desempenho por padrão, gráficos de alta fidelidade em tempo real e ferramentas artísticas**. 2019a. Disponível em: <<https://unity3d.com/pt/unity#engine-performance>>. Acesso em: 30/10/19.

TECHNOLOGIES, Unity. **Visual Studio C# integration**. 2019b. Disponível em: <<https://docs.unity3d.com/Manual/VisualStudioIntegration.html>>. Acesso em: 30/10/19.

TECHNOLOGIES, Unity. **WWWForm**. 2019c. Disponível em: <<https://docs.unity3d.com/ScriptReference/WWWForm.html>>. Acesso em: 19/10/19.