

UNIVERSIDADE DO ESTADO DO RIO GRANDE DO NORTE
CAMPUS DE NATAL
DEPARTAMENTO DE COMPUTAÇÃO
CURSO DE CIÊNCIA DA COMPUTAÇÃO

SERAFIM DO NASCIMENTO JUNIOR

VISUALIZAÇÃO 3D DA DINÂMICA DE REDES NEURAIS:
UMA FERRAMENTA COMPUTACIONAL

NATAL
2015

UNIVERSIDADE DO ESTADO DO RIO GRANDE DO NORTE
CAMPUS DE NATAL
DEPARTAMENTO DE COMPUTAÇÃO
CURSO DE CIÊNCIA DA COMPUTAÇÃO

SERAFIM DO NASCIMENTO JUNIOR

VISUALIZAÇÃO 3D DA DINÂMICA DE REDES NEURAIS:
UMA FERRAMENTA COMPUTACIONAL

NATAL
2015

SERAFIM DO NASCIMENTO JUNIOR

**VISUALIZAÇÃO 3D DA DINÂMICA DE REDES NEURAIS:
UMA FERRAMENTA COMPUTACIONAL**

Trabalho de Conclusão de Curso apresentado à Universidade do Estado do Rio Grande do Norte como um dos requisitos obrigatórios para obtenção do título de Bacharel em Ciência da Computação.

ORIENTADOR: Prof. Wilfredo Blanco Figuerola

NATAL
2015

**Catlogação da Publicação na Fonte.
Universidade do Estado do Rio Grande do Norte.**

Nascimento Junior, Serafim Do

Visualização 3d da dinâmica de redes neurais: Uma ferramenta computacional /
Serafim Do Nascimento Junior – Natal, RN, 2015.

55 f.

Orientador(a): Prof. Wilfredo Blanco Figuerola

Monografia (Bacharelado) Universidade do Estado do Rio Grande do Norte. Curso
de Ciências da Computação

1. Visualização 3D. 2. Atividade - Plasticidade Sináptica. 3. Dinâmica de Redes
Neurais. DyRENE Visual. I. Wilfredo Blanco Figuerola. II. Universidade do
Estado do Rio Grande do Norte. III. Título.

UERN/ BC

CDD 004

SERAFIM DO NASCIMENTO JUNIOR

**VISUALIZAÇÃO 3D DA DINÂMICA DE REDES NEURAIS:
UMA FERRAMENTA COMPUTACIONAL**

Trabalho de Conclusão de Curso apresentado à Universidade do Estado do Rio Grande do Norte como um dos requisitos obrigatórios para obtenção do título de Bacharel em Ciência da Computação.

Aprovado em: ____ / ____ / ____.

Banca Examinadora

Professor Dr. Wilfredo Blanco Figuerola
Universidade do Estado do Rio Grande do Norte

Professora Dra. Adriana Takahashi
Universidade do Estado do Rio Grande do Norte

Professor Dr. Carlos André Guerra Fonseca
Universidade do Estado do Rio Grande do Norte

A Deus.

A meus pais.

A minha vó Cabocla.

Aos meus verdadeiros amigos.

Aos excelentes Professores que eu tive na vida.

A Neurociência Computacional.

Ao Brasil.

E claro, a World Wide Web!

AGRADECIMENTOS

Muitas são as pessoas que contribuíram, diretamente ou indiretamente, para a concretização deste trabalho as quais eu tenho a agradecer.

Primeiramente, agradeço a Deus e aos seus seres iluminados por estarem comigo durante cada segundo da minha graduação me protegendo nos bons e nos ruins momentos e por terem permitido que estes acontecessem, pois tudo que me trouxe felicidade e tristeza me ajudou a amadurecer como ser humano e a reconhecer que nós temos limites.

A meus pais, Geralda e Serafim, pelo apoio incondicional que me deram em todos os momentos da minha vida, por me amarem e me protegerem, mesmo que implicitamente, dos maus caminhos, pelos conselhos e respeito aos meus anseios.

A minha avó, Maria (vó Cabocla), que, mesmo sem condições, sempre ajudou, de alguma maneira, minha Mãe e seus filhos nos momentos mais críticos pelos quais passaram e por ela ter sido uma mulher batalhadora que não desistiu de sua vida apesar de tudo o que passou.

A UERN por disponibilizar um campus universitário em Natal ofertando cursos de graduação, principalmente, o curso de Bacharelado em Ciência da Computação.

A Direção do Campus de Natal por sempre estarem lutando por melhorias na infraestrutura, nos serviços oferecidos, na assistência estudantil, entre outros.

Ao Departamento de Computação por ser possível encontrar excelentes Professores em seu corpo docente.

Aos profissionais técnico-administrativos pelo ótimo trabalho e por sempre tratarem todos educadamente e com carisma.

A todos os amigos e todas as amigas de turma que no início eram somente colegas, mas que acabaram se tornando uma família durante a graduação. Aos que largaram o curso e foram para outras instituições: Alex Rego, Bruno Gomes, Damião Renato, Daniel Moura, Felipe Vanderlei, Fernanda Medeiros (In Memoriam), George Kennedy, Helena Paula, Henrique Carlos, Hiranilson Andrade, Idianara Santos, Jackson Santos, João Figueiredo, Jussara França, Leilane Clarisse e Renato Gomes. Aos que continuaram na turma até o fim: Ailton Luna, Antônio Porto, Jádila Ravanna, Joselita Soares, Mariêta Cunha, Renato Carlos e Rodrigo Silva. Aos que se tornaram

parte da turma: Acássio dos Anjos, Alan Santana, Ana Katarina, Carlos Eduardo (Novinho), Edilene Alves, Hércules Brito, Itamar Brito, Jhonatta Faustino, Joel Gonçalves (Jogonoli), Katiane Farias, Matheus Araújo (1º), Matheus Araújo (2º), Jessica Silva, Jucieny Dantas, Marianna Araújo, Moaberdã Gomes, Rafael Jullian (Julliannnn!), Robson Dantas (Santiago), Suzyanne Oliveira, Tamires Farias, Thaise Albuquerque, Vandeclecio Silva, Wilton Ferreira e muito(a)s outro(a)s. Aos amigos que conquistei durante a graduação sanduíche: Abiola Sunmonu, Akie Katatura, Anna Kuinsler, Antônio Carlos, Carolina Gomes (Carol), Gretchen Phillips, Haiejin Youm (Blaire), Ikumi Seki, J. Nelson Amaral, Julio Vitor, Laura Michelle, Luis Aizpuru, Luiz Drummond, Luiz Tenaglia, Mauricio Loures, Minh Vuong, Nate Chan, Preet, Remy Wu (Xiaomin Wu - like celebration), Rod Loyola, Scott MacDonald, Shuwen Tian (Amanda), Supinder Perhar (Soupy), Stephen Edge (Steve), Tatiana Tavares, Tsukasa (Anri Mone), Valéria Siciliano, Wenyu Cai, Yu Hiu Poon (Charlie) and many others. Dentre todo(a)s o(a)s citado(a)s, agradeço especialmente a Abiola, Ailton, Anna, Antônio, Blaire, Carol, Charlie, Gretchen, Helena, Idianara, Járdila, Jackson, Julio, Luis, Luiz, Luiz, Nelson, Mariêta, Mauricio, Porto, Remy, Rod, Rodrigo, Scott, Shuwen, Soupy, Steve, Tatiana, Valéria e Wenyu por me inspirarem, me fazerem feliz e estarem comigo em momentos decisivos de minha graduação.

Aos Professores e Professoras que eu tive na UERN e na UofA que contribuíram muito e de maneira saudável com o meu desenvolvimento intelectual durante a minha graduação: Ana Lúcia Dantas, Adriana Takahashi, Anderson Abner, André Gustavo, Babak Behsaz, Bartira Paraguaçu, Brismark Goes, Bruno Cruz, Camila Araújo, Catherine Kmita, Carlos Alberto, Carlos André, César Medeiros, Djalma Xavier, Diogo Henrique, Felipe Denis, Glaucia Campos, João Pires, Karla Darlene, Lídio Sânzio, Lisa Alton, Marcelo, Missilene Farias, Nilanjan Ray, Nilson, Raul Paradedda, Rodolfo Bezerra, Soraya Campos e Tiago Gameleira. Dentre o(a)s citado(a)s, agradeço especialmente: a Professora Dra. Ana Lúcia Dantas por ter me proporcionado a oportunidade de ingressar em projetos de pesquisa e ter contribuído muito na minha jornada acadêmica (só nós sabemos o quanto) e também por ela ser uma excelente Professora, Educadora, Cientista, Pesquisadora, Gestora e um maravilhoso exemplo de Ser Humano para todos os que a admiram e a conhecem. Obrigado por tudo, ♥ Ana ♥ !; a Professora Dra. Adriana Takahashi por ter ministrado algumas das disciplinas que eu mais gosto (CNC, IA e CG (PDI)) na

vida, por ser a prova viva de que é possível melhorar a maneira de ensinar e por me fazer sorrir quando eu a vejo; ao Professor Dr. Carlos André Guerra Fonseca pelo profissionalismo, pelas excelentes aulas, pelo auxílio na elaboração deste trabalho, pela dedicação que tem para com o crescimento do curso de Ciência da Computação do Campus de Natal, por ser admirável e inspirar seus alunos a serem profissionais dedicados. Obrigado, ☺ Carlos André ☺ !; ao Professor Dr. Felipe Denis Mendonça de Oliveira por ser um excelente professor e ser humano, ter contribuído com excelentes aulas, oferecer oportunidades e acreditar em seus alunos. Obrigado, ☺ Felipe ☺ !; a Professora Ma. Camila de Araújo Sena por sempre encontrar as melhores maneiras de explicar assuntos complexos a(o)s seus/suas aluno(a)s, por deixá-lo(a)s tranquilo(a)s, por ensinar muito bem e pelo seu carisma. Obrigado, ♥ Camila ♥ !; a Professora Ma. Glaucia Campos por ser uma profissional esforçada, paciente em suas aulas e companheira de seus alunos; ao Professor Me. Rodolfo Bezerra pelas aulas de Eletromagnetismo (bons tempos!); a Professora Ma. Missilene Farias pelas maravilhosas aulas de Dispositivos Semicondutores e Teoria dos Circuitos (Amo!!!); e ao Professor Dr. Wilfredo Blanco por ter ajudado nos momentos mais críticos deste trabalho. I also thank Lisa Alton for teaching me many important things for my career and life, for making me feel comfortable with my English language skills, and for believing me. Miss you, ♥ Lisa ♥ !

A(o)s funcionário(a)s da instituição: Ailton, Alda Leda, André, Andrea Linhares, Aninha Castro, Bárbara, Canindé, Charlson, Eduardo, Elias, Emmanuel, Hugo, Jairo, Laura, Neide, Neto, Patrícia, Rose, Rosinaldo, e muito(a)s outro(a)s.

A minha tia Mariquinha por ter aberto as portas sempre que eu precisei fazer algum vestibular; a minha irmã Cida; a meus amigos Gleson, Jessica, Jevan, Josuel, Samuel e muitos outros por acreditarem em mim.

A Casa do Estudante do Rio Grande do Norte (CERGN) que, apesar da situação em que se encontra, me abrigou durante a minha graduação dando o melhor suporte que pôde oferecer. Também agradeço a todos que ajudaram e ainda ajudam esta instituição a se reestabelecer, em especial ao Prof. Dr. Carlos Roberto de Miranda Gomes.

A World Wide Web e a todos que a alimentam com bons materiais de estudo.

São muitas pessoas a agradecer e tenho certeza que esqueci várias, mas desejo que elas se sintam agradecidas ao lerem estes agradecimentos.

O homem deve saber que, de nenhum outro lugar, se não do cérebro vem a alegria, o prazer, o riso e a recreação, e a tristeza, melancolia, pessimismo e as lamentações. E então, de uma maneira especial, adquirimos sabedoria e conhecimento, e vemos e ouvimos para saber o que é justo e o que não é, o que é bom e o que é ruim, o que é doce e o que é sem sabor... E pelo mesmo órgão tornamo-nos loucos e delirantes, e sentimos medo e o terror nos assola... Todas essas coisas proveem do cérebro quando este não está sadio... Dessa maneira sou da opinião de que o cérebro exerce um grande poder sobre o homem.

Hipócrates, Da Doença Sacra (IV A.C)

RESUMO

Atualmente na área de Neurociência, muitos cientistas ao redor do mundo almejam visualizar a dinâmica de redes neurais de forma mais simples. Em uma rede neural, a dinâmica pode ser descrita por meio de dois elementos: sua atividade e sua plasticidade sináptica, isto é, mediante os disparos dos neurônios e da mudança das sinapses ao longo do tempo, respectivamente. A carência de recursos computacionais para auxiliar na visualização da dinâmica de redes neurais dificulta os estudos dos neurocientistas que, para conseguirem algum resultado, utilizam métodos mais trabalhosos com o auxílio de ferramentas de visualização de grafos para tentar visualizar a dinâmica. O objetivo geral deste trabalho é propor uma ferramenta computacional para visualização da dinâmica de redes neurais em três dimensões. Para este propósito, inicialmente foi realizado um estudo sobre os conhecimentos básicos de Neurociência, especificamente, sobre Redes Neurais. Por oferecer os conhecimentos necessários para a programação da ferramenta, a Computação Gráfica também foi uma área importante estudada. A respeito da implementação, a linguagem de programação C++ e o paradigma Orientado a Objetos foram utilizados, sendo auxiliada pelo ambiente de desenvolvimento Dev-C++. A síntese de imagens foi feita com o auxílio da biblioteca OpenGL e o menu da ferramenta foi criado utilizando-se a biblioteca de funcionalidades GLUT. Um arquivo de entrada foi definido para inserção das configurações das redes neurais a serem renderizadas. Finalmente, algumas funcionalidades foram implementadas, tais como: a rotação e o zooming in/out da rede neural, assim como a exportação de suas configurações para arquivos com a extensão *wrl*. Como resultado, este trabalho propõe a ferramenta computacional DyRENE Visual para visualização 3D da dinâmica de redes neurais utilizando-se a topologia altamente conectada. Após vários testes realizados na ferramenta utilizando exemplos de redes neurais, foi possível concluir que a ferramenta é uma solução para a dificuldade abordada e pode ser aperfeiçoada em trabalhos futuros.

Palavras-chave: Atividade e Plasticidade Sináptica. Dinâmica de Redes Neurais. DyRENE Visual. Neurociência Computacional. Visualização Científica com OpenGL.

ABSTRACT

Nowadays in Neuroscience, many scientists over the world are searching for ways to visualize the dynamics of neural networks in an easier way. In a neural network, the dynamics can be described by two elements: its synaptic activity and plasticity, that is, through spike trains of neurons and synaptic changes over time, respectively. The lack of appropriate computational tools for visualization of the dynamics of neural networks makes neuroscientists research way more difficult because to find results, they have to use complicated techniques assisted by graph visualization tools to try to visualize the dynamics. The main goal of this work is to propose a computational tool for visualization of the dynamics of neural networks in three dimensions. For this purpose, first of all it was conducted a study about the basic concepts in Neuroscience, in particular, about Neural Networks. Furthermore, for providing the knowledge required for programming the tool, it was also conducted a study about Computer Graphics. Regarding the implementation of the tool, the C++ programming language, the OO paradigm, and the integrated development environment Dev-C++ were used. Image rendering was performed by the OpenGL library and the tools menu was created by using GLUT library functionalities. A specific input file was defined for inserting neural network configurations in order to be rendered by the tool. Finally, some functionalities were implemented in order to manipulate neural network configurations rendered in the screen, such as: rotation, zooming in/out, and exportation of them to *wrl* files, in case the user wants to render them by using other 3D visualization tools. As result, this work proposes the computational tool DyRENE Visual for 3D visualization of the dynamics of neural networks by using the fully connected topology. After several tests carried out by the tool by using examples of neural networks, it was possible to conclude that the tool is a solution for the difficulty approached and can also be improved in future works.

Keywords: Synaptic Activity and Plasticity. Dynamics of Neural Networks. DyRENE Visual. Computational Neuroscience. Scientific Visualization with OpenGL.

LISTA DE FIGURAS

1 - Aplicações da VC na Medicina e na exploração de petróleo	19
2 - Tipos de redes neurais.....	22
3 - Neurônio biológico	23
4 - Neurônio artificial	24
5 - Arquitetura da ferramenta	27
6 - Diagrama de classes.....	31
7 - Descrição do arquivo de entrada	32
8 - Descrição do arquivo de saída.....	34
9 - Mapas de cores aplicados nas redes renderizadas	35
10 - Apresentação da ferramenta.....	37
11 - Telas informativas da ferramenta.....	38
12 - Arquivo RedeNeural.drn.....	39
13 - Leitura do arquivo de entrada	40
14 - Renderização da rede neural	41
15 - Aplicação dos mapas de cores	42
16 - Visualização das configurações e da dinâmica da rede neural.....	43
17 - Rotação nos eixos x e y	44
18 - Operações de zooming in/out	45
19 - Exportação das configurações da rede neural para arquivos <i>wrl</i>	46

LISTA DE GRÁFICOS

1 - Modelos da atividade eletrofisiológica.....	21
---	----

LISTA DE ABREVIATURAS E SIGLAS

2D	Bidimensional
3D	Tridimensional
ASCII	American Standard Code for Information Interchange
CG	Computação Gráfica
GLUT	OpenGL Utility Toolkit
NC	Neurociência Computacional
OO	Orientado a Objetos
OpenGL	Open Graphics Library
RN	Rede Neural
RNA	Rede Neural Artificial
VC	Visualização Científica
VRML	Virtual Reality Modeling Language

SUMÁRIO

1 INTRODUÇÃO	15
1.1 OBJETIVOS	16
1.2 ESTRUTURA DO TRABALHO	17
2 FUNDAMENTAÇÃO TEÓRICA	18
2.1 COMPUTAÇÃO GRÁFICA	18
2.1.1 Visualização Científica	19
2.2 NEUROCIÊNCIA	20
2.2.1 Redes Neurais	22
3 DESENVOLVIMENTO DA FERRAMENTA	26
3.1 ARQUITETURA DA FERRAMENTA	27
3.2 DIAGRAMA DE CLASSES	28
3.2.1 Classe operacoesDaFerramenta	28
3.2.2 Classe ambienteOpenGL	29
3.2.3 Classe mapaDeCores	30
3.2.4 Classe telasDaFerramenta	30
3.3 ESTRUTURA DO ARQUIVO DE ENTRADA	31
3.4 ESTRUTURA DO ARQUIVO DE SAÍDA	33
3.5 MAPA DE CORES	34
3.6 IMPLEMENTAÇÃO	36
4 A FERRAMENTA DyRENE VISUAL	37
4.1 DyRENE VISUAL	37
4.2 ARQUIVO DE ENTRADA	39
4.3 LEITURA E RENDERIZAÇÃO DA REDE NEURAL	40
4.4 MUDANÇA DOS MAPAS DE CORES	41
4.5 VISUALIZAÇÃO DA DINÂMICA DA REDE NEURAL	43
4.6 ROTAÇÃO NOS EIXOS X E Y	44
4.7 ZOOMING IN/OUT	45
4.8 EXPORTAÇÃO DAS CONFIGURAÇÕES DA RN PARA ARQUIVOS <i>WRL</i>	46
4.9 CONCLUSÕES SOBRE A FERRAMENTA DyRENEVISUAL	47
5 CONCLUSÕES	48

5.1 TRABALHOS FUTUROS	48
REFERÊNCIAS	50
GLOSSÁRIO	53

1 INTRODUÇÃO

Um dos objetivos da Ciência da Computação é o desenvolvimento de sistemas computacionais visando à solução de problemas do mundo real em diferentes áreas da ciência e da tecnologia. Desde a invenção do primeiro computador digital, se acentuou a necessidade por ferramentas computacionais para auxiliar o ser humano na realização de tarefas complexas.

São muitas as áreas da ciência que necessitam de ferramentas computacionais para auxiliar os cientistas em suas pesquisas e, dessa maneira, facilitar a descoberta de novos conhecimentos. Muitas áreas precisam de ferramentas que possibilitem uma melhor compreensão de dados, em alguns casos, a retratação de dados através de imagens facilita o entendimento sobre o objeto de estudo.

A Computação Gráfica (CG) é uma área da Ciência da Computação que propicia o desenvolvimento de ferramentas para o processamento digital de imagens, a análise de imagens e a síntese de imagens. As ferramentas desenvolvidas podem empregar uma ou mais dessas subáreas da CG. A Síntese ou Renderização de Imagens torna possível o desenvolvimento de ferramentas de visualização científica para diversas finalidades, pois permite gerar imagens digitais, tanto bidimensionais quanto tridimensionais, a partir de modelos e dados matemáticos que as representam (AZEVEDO e CONCI, 2003).

Na área de Neurociência, ferramentas gráficas que possibilitem a visualização, em três dimensões, de fenômenos que ocorrem no cérebro dos animais podem auxiliar no entendimento mais detalhado deste complexo órgão que é vital para os seres vivos. O cérebro contém células chamadas de neurônios, as quais são encarregadas de processar e armazenar estímulos externos. Os neurônios estão conectados formando redes que processam e transmitem os estímulos entre as diferentes regiões do cérebro.

Dessa maneira, quando um neurônio A processa um estímulo e ele é transmitido para um neurônio B, a conexão entre eles muda, chamando este fenômeno de plasticidade sináptica. Em uma Rede Neural (RN), a visualização ao longo do tempo da atividade e da plasticidade sináptica dos neurônios que a compõem, também conhecida como dinâmica da RN, é uma tarefa não trivial que

muitos Neurocientistas almejam tornar mais simples. Dentro deste contexto, esta dificuldade pode ser superada com a utilização de uma ferramenta de visualização.

Devido à dificuldade e à carência de ferramentas computacionais para auxiliar neurocientistas na visualização da dinâmica de redes neurais, foi proposta a elaboração de um projeto de pesquisa visando à criação de uma ferramenta computacional para satisfazer a esta necessidade. O projeto foi executado com sucesso e neste trabalho é apresentado o processo de desenvolvimento desta ferramenta assim como a sua concepção.

Para concluir, neste trabalho se explora a Síntese de Imagens para a construção de uma ferramenta computacional, visando à geração de imagens tridimensionais das configurações sinápticas de redes neurais, com o propósito de visualizar as suas dinâmicas e, assim, auxiliar os neurocientistas em suas pesquisas.

1.1 OBJETIVOS

O objetivo principal deste trabalho é propor uma ferramenta computacional que possibilite a visualização da dinâmica de redes neurais em três dimensões, a partir de dados das configurações dessas redes obtidos ao longo de um determinado tempo.

Com o intuito de permitir que o usuário, provavelmente um(a) neurocientista, consiga inserir na ferramenta as configurações da rede neural em estudo, foi definido um padrão de arquivo em ASCII (IETF, 2015) que tem um formato específico para ser lido pela ferramenta. Este arquivo é disponibilizado com a ferramenta para que o usuário consiga inserir os dados das configurações da RN que deseja visualizar. Além deste, outros objetivos específicos deste trabalho são:

- Realizar uma breve revisão bibliográfica sobre: Neurociência, conceitos e técnicas de CG e sobre a biblioteca OpenGL, para aplicação das técnicas de síntese de imagens da ferramenta desenvolvida;
- Renderizar a RN com elementos geométricos simples;
- Mudar o mapa de cores da RN renderizada;
- Permitir a visualização de todas as suas configurações sinápticas assim como dos estados de ativação de seus neurônios em diferentes instantes de tempo;
- Possibilitar a rotação da rede em torno dos eixos x e y;

- Efetuar operações de zoom;
- Exportar o arquivo de entrada para outro formato de arquivo; e
- Fornecer ajuda ao usuário quanto ao uso da ferramenta.

1.2 ESTRUTURA DO TRABALHO

Este trabalho está organizado em cinco capítulos.

O Capítulo 2 é voltado ao embasamento teórico necessário para a compreensão dos conceitos utilizados no trabalho e conhecimento das técnicas essenciais para a sua concretização.

O Capítulo 3 denota os materiais e métodos utilizados para implementação da ferramenta computacional apresentada neste trabalho.

No Capítulo 4, todos os detalhes da ferramenta desenvolvida são apresentados claramente.

Finalmente, o Capítulo 5 mostra as conclusões obtidas com o desenvolvimento da ferramenta proposta neste trabalho assim como sugere alguns trabalhos futuros.

2 FUNDAMENTAÇÃO TEÓRICA

Este capítulo apresenta conceitos básicos da Computação Gráfica e da Neurociência que foram necessários para o desenvolvimento deste trabalho e são importantes para situar o leitor no decorrer dos demais capítulos para, dessa maneira, facilitar o seu entendimento da problemática abordada.

2.1 COMPUTAÇÃO GRÁFICA

A CG é a área da Ciência da Computação com o propósito de criar imagens digitais tanto para representação de dados e informações quanto para simulações do mundo real (AZEVEDO e CONCI, 2003). Ela foi criada por cientistas e engenheiros em centros de pesquisa americanos inicialmente como uma ferramenta de visualização (MASSON, 2015) e evoluiu tanto que hoje pode ser dividida em três subáreas, a saber: o Processamento de Imagens, a Síntese de Imagens e a Visão Computacional (também conhecida como Análise de Imagens).

A Síntese de Imagens, também conhecida como Visualização Científica ou Computacional (AZEVEDO e CONCI, 2003), possibilita gerar imagens 2D ou 3D a partir de modelos e dados matemáticos que as representam.

Segundo Gonzales e Woods (2007) imagens podem ser definidas como funções bidimensionais, onde x e y são coordenadas espaciais planas e a amplitude da função em que cada par de coordenadas (x, y) é chamada de intensidade da imagem naquela posição. Caso os valores de intensidade desta função sejam todos finitos, a imagem pode ser chamada de imagem digital. A subárea de processamento digital de imagens se refere a processar imagens digitais por meio de um computador.

A Visão Computacional tem o intuito de analisar imagens digitais para extração de características importantes para a determinação dos objetos que as compõem, a partir de suas representações geométricas, entre outras coisas.

Dentre as subáreas da CG, a Visualização Científica (VC) tem se tornado uma das mais importantes para praticamente todas as áreas da ciência. Ela é uma área que envolve tanto a síntese quanto o processamento de imagens. McCormick et al. (1987) define a VC como um método de computação que possibilita a

conversão de dados simbólicos em representações geométricas para permitir que pesquisadores observem visualmente seus cálculos e simulações e, dessa maneira, obtenham conclusões mais precisas em suas pesquisas. A seção a seguir aborda melhor a definição da VC e suas áreas de aplicação.

2.1.1 Visualização Científica

Conforme Johnson e Hansen (2005), VC é uma área da ciência dedicada a criação de imagens com o objetivo de transmitir informações importantes advindas de processos e dados representativos. “Suas técnicas envolvem algoritmos de processamento de dados que extraem os dados de interesse de uma amostra e os convertem em uma forma adequada para representação” (SCHEER et al., 2011, p. 74).

A criação e o uso de sistemas computacionais para auxiliar na VC tem se tornado cada vez mais popular devido a crescente evolução da tecnologia. Inúmeras áreas do conhecimento têm se interessado pelo desenvolvimento de ferramentas computacionais para auxiliá-las em suas pesquisas.

A VC é uma área multidisciplinar e, portanto, sua aplicabilidade abrange diversas áreas do conhecimento. Em Física, por exemplo, ela permite a visualização da dinâmica dos fluídos; em Medicina, de imagens médicas; em Astrofísica e Cosmologia, da formação de estrelas e de ondas gravitacionais; em Neurociência Computacional, de Redes Neurais (RNs); enfim, de fenômenos naturais existentes nas variadas áreas da ciência por meio de representações artificiais. A Figura 1 apresenta aplicações da VC em algumas das áreas citadas anteriormente.

Figura 1: Aplicações da VC na Medicina e na exploração de petróleo.

A) Visualização do cérebro humano.

B) Visualização 3D de um projeto para construção de plataformas de exploração de petróleo.



Fonte: A) Adaptada de (LLNL, 2015). B) (GATTASS, 2015).

Nos últimos anos, as pesquisas relacionadas à Neurociência Computacional (NC) têm evoluído muito e, conseqüentemente, ferramentas computacionais para visualização bi e/ou tridimensional de seus resultados têm surgido e se tornado cada vez mais necessárias.

A NC é uma recente área da Neurociência que tem o propósito de “explicar como os sinais elétricos e químicos nas células e redes de células interconectadas do cérebro são usadas para representar e processar informação” (ROQUE, 2011, p. 1). Segundo Roque (2011), para alcançar este objetivo, a NC utiliza modelos matemáticos e computacionais de neurônios e circuitos cerebrais com o intuito de integrar dados experimentais, adquiridos por meio das mais diferentes técnicas existentes.

Além da Psicologia, Matemática, Física, Engenharia Elétrica, Filosofia e Ciência da Computação, outras áreas do conhecimento também auxiliam a NC a tentar conquistar seus objetivos. As RNs, por exemplo, são fundamentais para esta área por permitirem uma melhor compreensão do funcionamento do cérebro por meio de modelos matemáticos.

Visando auxiliar a NC, este trabalho tem o objetivo de aplicar a VC para RNs por meio da criação de uma ferramenta computacional que permita a visualização 3D de RNs e, principalmente, de sua dinâmica, devido à carência de sistemas computacionais para este fim. A seção a seguir detalha melhor a área de RNs e sua importância em NC.

2.2 NEUROCIÊNCIA

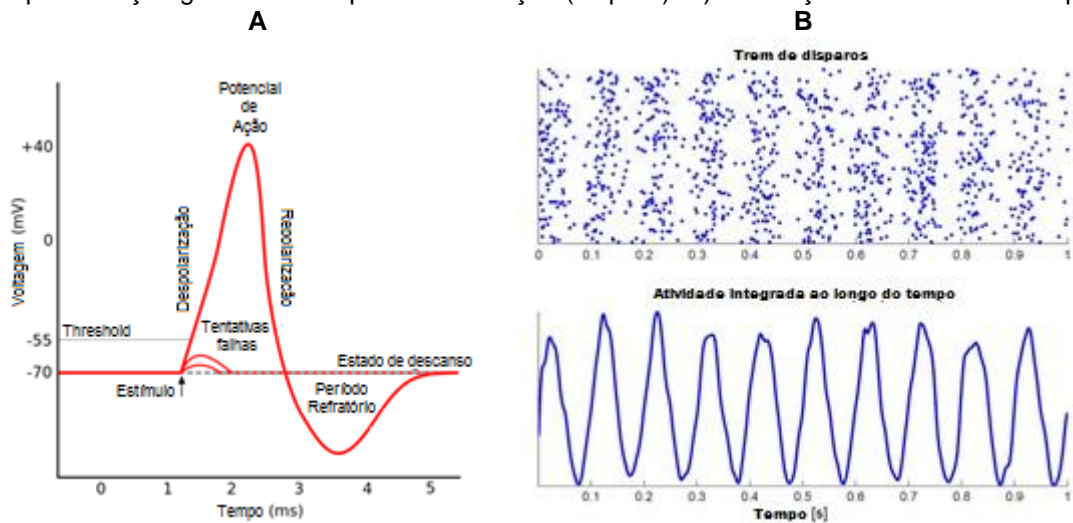
Segundo Ventura (2010, p. 123) a Neurociência “compreende o estudo do sistema nervoso e suas ligações com toda a fisiologia do organismo, incluindo a relação entre cérebro e comportamento”. O sistema nervoso da maioria dos animais é composto por um cérebro, uma medula espinhal e os nervos periféricos. Ademais, a Neurociência estuda desde a estrutura do sistema nervoso até o seu funcionamento, desenvolvimento, sua evolução e conexão entre a mente e o comportamento, assim como suas alterações.

É uma Ciência bem complexa e, por isso, necessita do trabalho de profissionais de diferentes áreas para ser bem sucedida. Entre outros desafios, a

Neurociência tenta explicar os diversos comportamentos dos animais como resultado da atividade de neurônios, os quais são capazes de integrar e gerar saídas de pulsos elétricos, mais conhecidos como potenciais de ação, em inglês, *action potentials* (CHOREV et al., 2009). A eletrofisiologia é a área que nos permite capturar/registrar e estudar tais sinais que podem ser observados no Gráfico 1.

Gráfico 1: Modelos da atividade eletrofisiológica.

A) Representação gráfica de um potencial de ação (disparo). B) Simulação de um trem de disparo.



Fonte: Adaptadas de (WIKIPEDIA, 2015).

O Gráfico 1A apresenta o disparo de um neurônio de 40mV. No Gráfico 1B (superior), cada ponto azul representa um disparo individual. Por outro lado, no Gráfico 1B (inferior), é possível observar como o grupo de neurônios dispara sincronizado e sua atividade integrada (somada) é mostrada.

Neurônios normalmente estão conectados, formando parte de uma rede. Assim sendo, além de estudar os sinais eletrofisiológicos unitários de cada neurônio, é importante considerar o sinal resultante da interação de uma população de neurônios, isto é, uma RN. Estas interações, desde o nível unitário até o populacional, geram ritmos ou atividades neurais repetitivas, que comumente são chamadas de oscilações neurais.

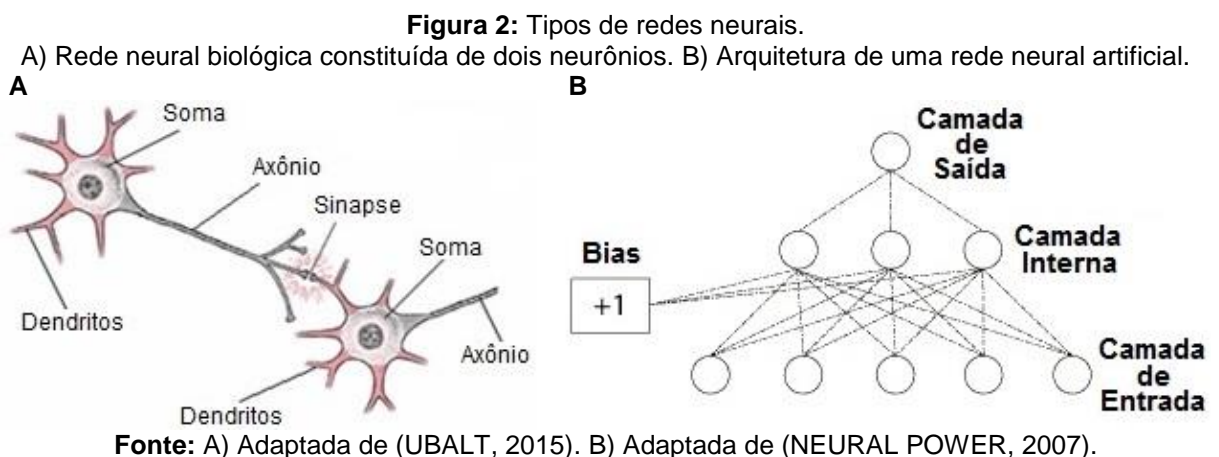
A atividade oscilatória de uma população de neurônios é, tipicamente, resultante de uma sincronização em seus padrões de disparos, como ilustrado no Gráfico 1B (superior). Geralmente, estas oscilações podem ser caracterizadas pela sua frequência, amplitude e fase, conforme observado no Gráfico 1B (inferior).

Não se pode falar de ritmos neurais sem entender como estas populações de neurônios são conectadas e como estas conexões, chamadas de sinapses, mudam ao longo do tempo. Este processo dinâmico, onde as sinapses variam dependendo da atividade (sincrônica ou não) dos neurônios conectados, é chamado de plasticidade sináptica. Em uma RN, cada neurônio pode receber vários sinais de entrada através de suas conexões, mas produz um único sinal de saída. A conexão de saída é dividida em ramos, a fim de transmitir o sinal único para as conexões de entrada de outros neurônios.

Uma forma de representar estas conexões (comumente nomeada como topologia da RN) pode ser através de uma matriz quadrada \mathbf{W} de ordem N , onde N representa o número total de neurônios. Cada elemento w_{ij} da matriz representa a força da conexão entre o neurônio i e o neurônio j . \mathbf{W} pode ser assimétrica ($w_{ij} \neq w_{ji}$) e os valores da sua diagonal principal $w_{ii}=0$, que garante que a saída de um neurônio não se conecta com si mesmo, evitando estados instáveis (HOPFIELD, 1982). A matriz \mathbf{W} é conhecida como matriz de configuração sináptica.

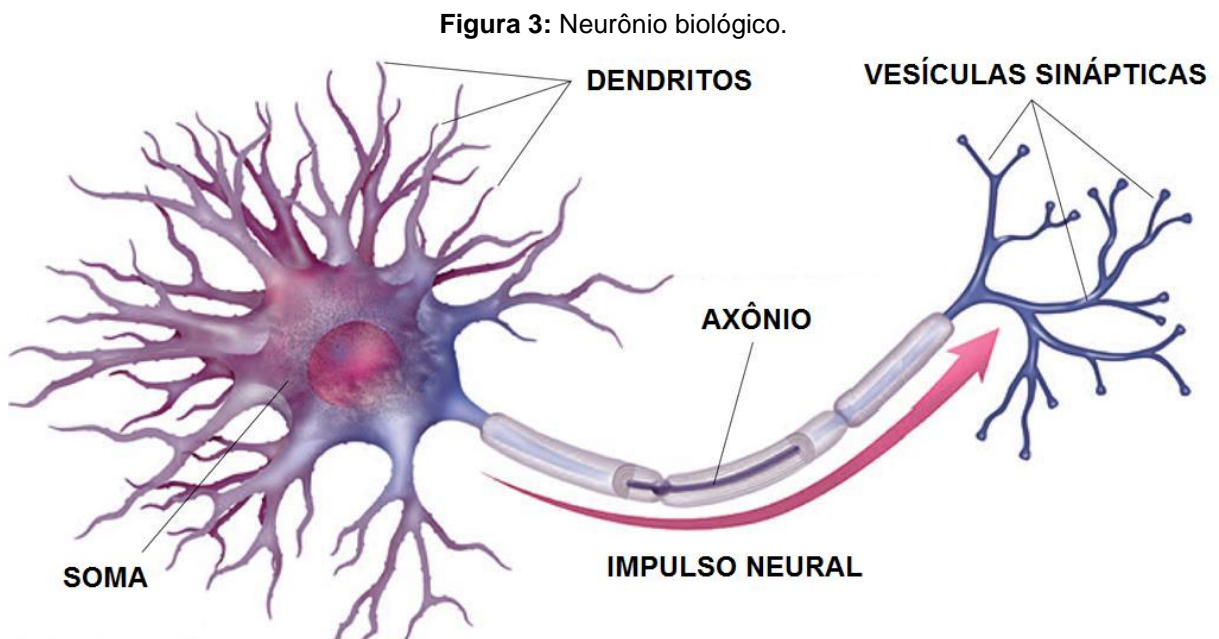
2.2.1 Redes Neurais

As Redes Neurais biológicas são, na verdade, um conjunto de neurônios do cérebro conectados entre si. Por outro lado, as Redes Neurais artificiais (RNAs) são modelos matemáticos das redes biológicas que tentam representar o que elas fazem. A Figura 2 ilustra esses dois tipos de RNs, respectivamente.



A Figura 2A apresenta uma RN simples constituída de dois neurônios dando ênfase à sinapse entre eles. A Figura 2B mostra uma RNA constituída de três camadas: a camada de entrada, a camada interna (única) e a camada de saída. Ademais, o bias também é representado.

Ludwig Jr. e Costa (2007) afirmam que os principais componentes dos neurônios biológicos são: os dendritos, a soma e o axônio. Segundo eles, os dendritos recebem os estímulos transmitidos pelos demais neurônios da rede; a soma coleta e combina informações vindas de outros neurônios através dos dendritos; e o axônio transmite o estímulo liberado pela soma para outros neurônios da RN através das vesículas sinápticas. A Figura 3 ilustra um neurônio biológico com as características descritas anteriormente e, inclusive, o impulso neural que, neste trabalho, é tratado como potencial de ação.

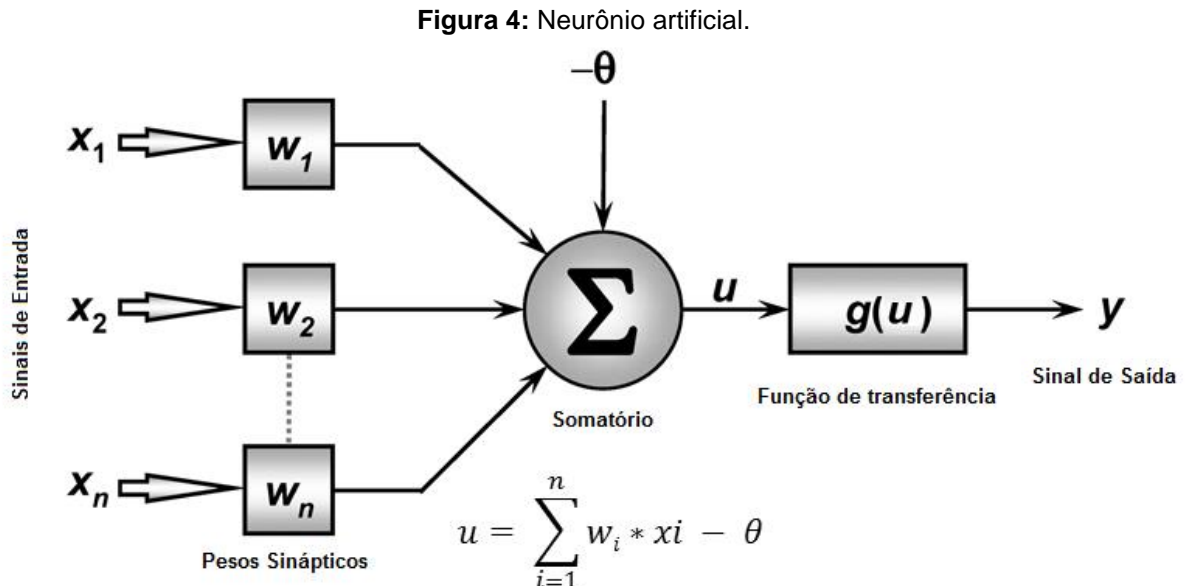


Fonte: Adaptada de (ANATOMY DIAGRAM, 2015).

Assim como o neurônio biológico, o neurônio artificial recebe um ou mais sinais de entrada e retornam um único sinal de saída. Este sinal pode ser repassado como sinal de saída da rede ou como sinal de entrada para um ou mais neurônios da próxima camada. Dendritos e axônios são representados somente pelas sinapses e a força de ligação é representada por uma grandeza denominada peso sináptico.

As entradas do neurônio são multiplicadas pelos seus pesos sinápticos correspondentes formando as entradas ponderadas. A partir daí, o neurônio totaliza

essas entradas e compara o resultado com uma função de transferência que, dependendo de qual tenha sido escolhida, decidirá se o sinal será transmitido para a outra camada ou não. A Figura 4 ilustra um neurônio artificial com as características descritas anteriormente para uma melhor compreensão.



Fonte: Adaptada de (BORGES, 2013).

A dinâmica de uma RN pode ser descrita com dois elementos: a sua atividade e sua plasticidade sináptica, respectivamente, por meio dos disparos dos neurônios e da mudança das sinapses ao longo do tempo. Em NC a visualização tridimensional da dinâmica de RNs, sejam elas biológicas ou artificiais, é de extrema importância para os pesquisadores desta área; porém, a carência de ferramentas computacionais que atendam a esta necessidade dificulta a análise deste fenômeno.

Como o objetivo deste trabalho é propor uma ferramenta computacional que possibilite a visualização desta dinâmica em três dimensões (3D), para atender a esta necessidade, a dinâmica de uma RN é representada por um conjunto constituído de todas as configurações da rede para cada momento no tempo. Cada configuração contém um vetor de ativação e uma matriz de plasticidade sináptica. O vetor de ativação guarda os estados de ativação dos neurônios, 0.0 para inativo e 1.0 para ativo. A matriz de plasticidade sináptica armazena os pesos sinápticos que representam as conexões sinápticas entre os neurônios da rede. As configurações da RN são armazenadas em um arquivo de leitura no padrão ASCII.

Esta ferramenta é importante para pesquisadores que trabalham com Neurociência Computacional especificamente os que estudam a evolução da dinâmica das RNs. O capítulo a seguir apresenta como ocorreu o desenvolvimento da ferramenta.

3 DESENVOLVIMENTO DA FERRAMENTA

O desenvolvimento deste trabalho iniciou-se com a realização de uma pesquisa bibliográfica sobre suas principais áreas do conhecimento, a Computação Gráfica e a Neurociência. O propósito desta pesquisa foi estudar, detalhadamente, essas grandes áreas para obter os conhecimentos necessários à implementação de uma ferramenta de visualização científica para simulação da dinâmica de redes neurais em três dimensões.

Posteriormente, foram concebidas as etapas de análise e modelagem da ferramenta, etapas onde foram definidas:

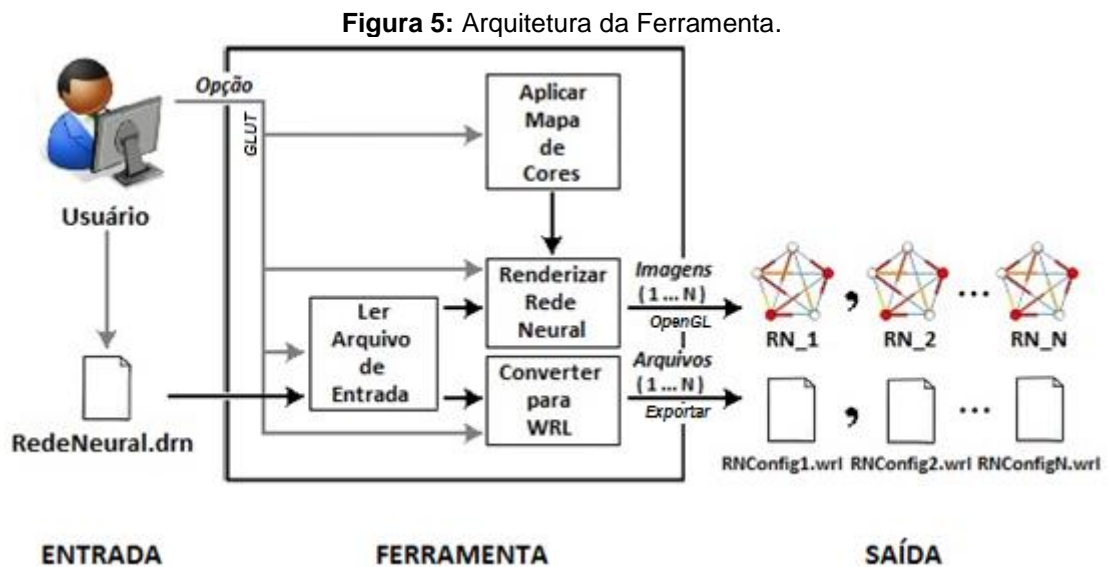
- as funcionalidades essenciais (requisitos funcionais) que foram implementadas para auxiliar na visualização da dinâmica da RN e, também, funcionalidades extras (requisitos não funcionais);
- a arquitetura da ferramenta;
- o diagrama de classes;
- a estrutura do arquivo de entrada;
- a estrutura do arquivo de saída; e
- a maneira como a ferramenta foi implementada.

Após as etapas anteriores, realizou-se a etapa mais importante deste trabalho, a construção da ferramenta, etapa onde foram implementadas todas as funcionalidades estabelecidas para a concepção da ferramenta proposta neste trabalho. Finalmente, ao se concluir a implementação, foram realizados testes com Redes Neurais hipotéticas visando a validação da ferramenta. Estes testes são apresentados no Capítulo 6 deste trabalho.

Neste capítulo, portanto, são apresentadas as etapas para se alcançar os objetivos propostos, isto é, os materiais e métodos utilizados no processo de desenvolvimento da ferramenta proposta neste trabalho. Sendo assim, ele se encontra dividido da seguinte maneira: a Seção 3.1 trata da arquitetura da ferramenta; a Seção 3.2 do diagrama de classes; a Seção 3.3 da estrutura do arquivo de entrada; a Seção 3.4 da estrutura do arquivo de saída; a Seção 3.5 do mapa de cores; e, finalmente, a Seção 3.6 trata da implementação da ferramenta.

3.1 ARQUITETURA DA FERRAMENTA

“Arquitetura é a organização fundamental de um sistema, representada por seus componentes, seus relacionamentos com o ambiente, e pelos princípios que conduzem seu design e evolução” (ISO/IEEE 1471-2000, p. 24, 2007, apud BARBOSA, 2009, p. 88). Baseando-se neste princípio e como resultado da etapa de análise dos requisitos, foi elaborada a arquitetura da ferramenta para esboçar de maneira mais intuitiva o domínio do problema a ser solucionado. Esta arquitetura é apresentada na Figura 5.



Fonte: Própria.

Como é possível perceber na arquitetura, inicialmente o usuário deve inserir no arquivo de entrada (apresentado na Seção 3.3 deste capítulo) os dados referentes à descrição da rede neural, assim como suas configurações ao longo do tempo. Em seguida, o usuário pode utilizar a ferramenta para ler os dados do arquivo de entrada e carregá-los em estruturas de dados na memória (como é descrito no diagrama de classes apresentado na próxima seção). Após executados os passos anteriores, o usuário poderá renderizar em três dimensões as configurações de sua RN.

Uma vez que a RN é visualizada, a ferramenta permite uma série de operações que fazem parte dos objetivos específicos deste trabalho. Dentre elas podem-se efetuar: a mudança do mapa de cores da rede; a navegação através das

configurações temporais da rede; e a conversão de cada uma dessas configurações para arquivos no formato *wrl*, visando a visualização da rede por meio de outras ferramentas de visualização científica.

3.2 DIAGRAMA DE CLASSES

Segundo Booch et al. (2000) os diagramas da UML (UML, 2015) encontrados com maior frequência na modelagem de sistemas orientados a objetos são os diagramas de classes. O diagrama de classes tem o propósito de definir a estrutura das classes utilizadas pelo sistema por meio da apresentação dos atributos e métodos presentes em cada classe, além de determinar a maneira como as classes se relacionam e trocam informações entre si (GUEDES, 2014).

Dessa maneira, com base na importância desses diagramas para o desenvolvimento de sistemas, foi elaborado um diagrama de classes para que fosse iniciada a implementação da ferramenta e, conseqüentemente, tornasse mais ágil seu processo de desenvolvimento. A ferramenta de modelagem UML utilizada para concepção deste artefato de software foi a Astah Community (ASTAH, 2015), uma ferramenta livre para projeto de software, por ser bastante consolidada na comunidade de desenvolvimento.

Para uma melhor compreensão de como as classes da ferramenta se encontram relacionadas, a Figura 6 apresenta o diagrama de classes criado. No diagrama, podemos observar que existem as seguintes classes: *operacoesDaFerramenta*, *ambienteOpenGL*, *mapaDeCores* e *telasDaFerramenta*. Sendo assim, nas subseções a seguir descrevemos, respectivamente, cada uma delas.

3.2.1 Classe *operacoesDaFerramenta*

A classe *operacoesDaFerramenta* é a principal classe do diagrama, pois sua função é permitir ao usuário realizar todas as operações estabelecidas para a ferramenta. Ela é composta por quatro atributos: *nNeuronios*, *nConfigsRN*, *vetorAtivacaoNeuronios* e *matrizConfigSinaptica*. Eles foram criados com o propósito

de armazenar os dados referentes às características das RNs a serem renderizadas pela ferramenta e, portanto, são descritos detalhadamente a seguir.

- **nNeuronios**: tem a finalidade de armazenar um valor inteiro referente a quantidade de neurônios da rede;
- **nConfigsRN**: tem o intuito de armazenar um valor inteiro equivalente a quantidade de configurações da rede;
- **vetorAtivacaoNeuronios**: tem o propósito de guardar todos os estados de ativação dos neurônios de cada configuração da rede; e
- **matrizConfigSinaptica**: tem o objetivo de armazenar os pesos sinápticos de todas as configurações da rede.

Esta classe se relaciona com todas as demais classes do diagrama por elas a auxiliarem em suas operações. Seu relacionamento com a classe ambienteOpenGL, por exemplo, se dá pelo fato desta classe permitir a renderização de imagens por meio da biblioteca OpenGL assim como a utilização do menu criado por meio da biblioteca GLUT. Para possibilitar a visualização da rede por meio de outros conjuntos de cores, ela precisa se relacionar com a classe mapaDeCores. Para concluir, ela também necessita da classe telasDaFerramenta para guiar o usuário quanto a utilização da ferramenta.

3.2.2 Classe ambienteOpenGL

A classe ambienteOpenGL tem o propósito de prover suporte as classes da ferramenta que necessitem de alguma funcionalidade da biblioteca OpenGL assim como da GLUT, respectivamente, para renderização de imagens e sua manipulação por meio do menu. Ela é somente composta por dois atributos: `_argv[1]` e `_argc`. Eles são necessários para dar acesso à linha de comando com a qual o programa foi chamado e são mais bem detalhados nos tópicos a seguir.

- **_argv[1]**: tem a função de armazenar a string referente ao nome da ferramenta; e
- **_argc**: tem a finalidade de armazenar a quantidade de argumentos que compõem o nome da ferramenta.

Esta classe se relaciona com as classes operacoesDaFerramenta e telasDaFerramenta. Para a classe operacoesDaFerramenta, a classe

ambienteOpenGL propicia a renderização de imagens, a realização de transformações geométricas e a criação de um menu para auxiliá-la quanto ao uso de suas operações. A classe ambienteOpenGL auxilia a classe telasDaFerramenta somente com a renderização de texto para a criação das telas informativas da ferramenta.

3.2.3 Classe mapaDeCores

A classe mapaDeCores tem o intuito de criar os mapas de cores a serem utilizados para coloração das RNs. Ela lê os três arquivos que contém os mapas de cores a serem utilizados na ferramenta e armazena-os em seus três atributos, a saber: mapaCorJetRGB, mapaCorHotRGB e mapaCorAutumnRGB. Estes atributos são matrizes de ordem 1024x3, onde 1024 indica a quantidade de cores e 3 as componentes RGB dessas cores, e são melhor explicados nos tópicos seguir.

- **mapaCorJetRGB:** é uma matriz que tem o objetivo de guardar os valores RGB do mapa de cores Jet que estão presentes no seu arquivo de entrada;
- **mapaCorHotRGB:** é uma matriz que tem o objetivo de guardar os valores RGB do mapa de cores Hot que estão presentes no seu arquivo de entrada;
- **mapaCorAutumnRGB:** é uma matriz que tem o objetivo de guardar os valores RGB do mapa de cores Autumn que estão presentes no seu arquivo de entrada.

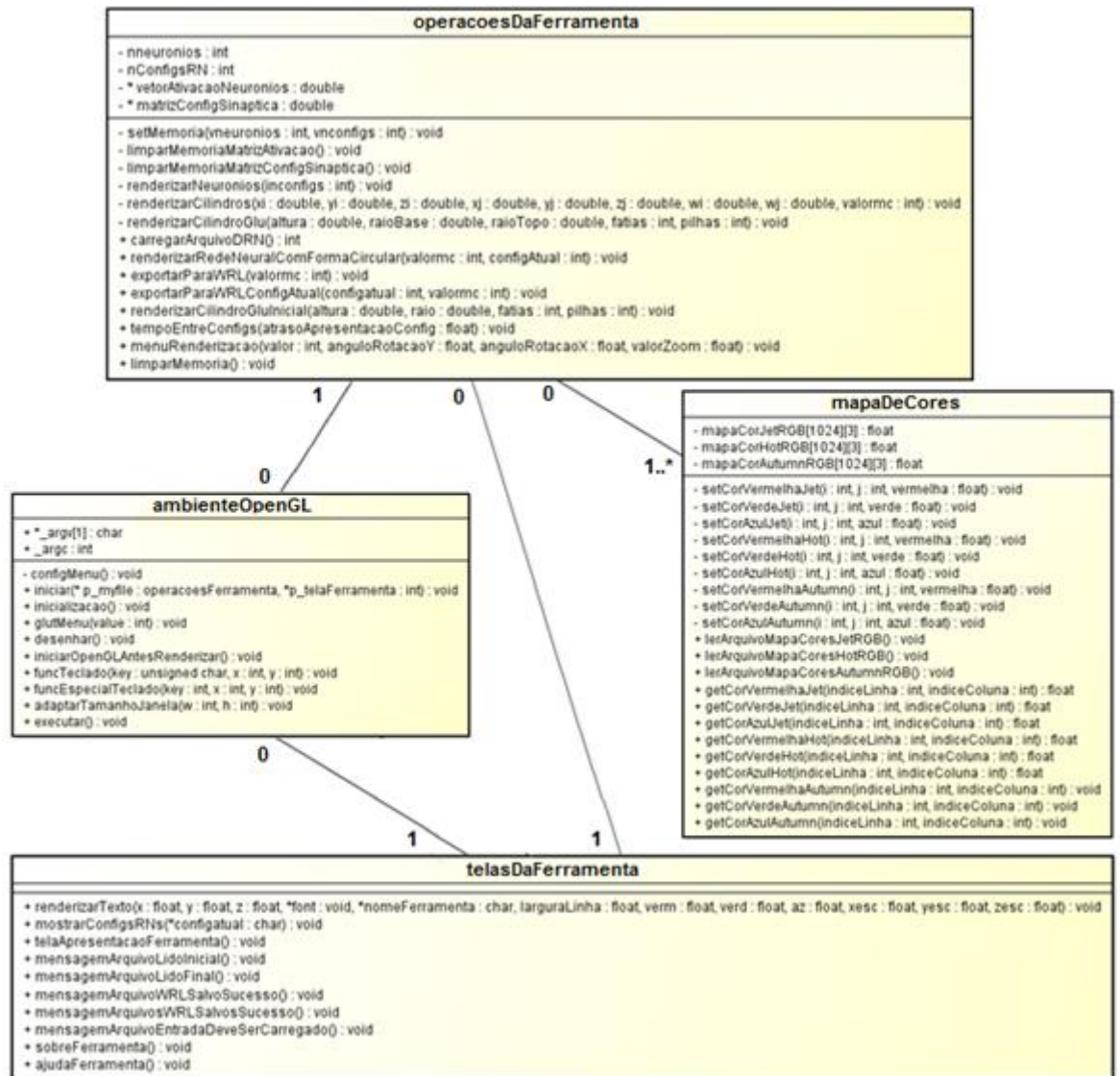
Esta classe se relaciona com a classe operacoesDaFerramenta para dar suporte na geração e consequente aplicação dos mapas de cores na RN renderizada.

3.2.4 Classe telasDaFerramenta

A classe telasDaFerramenta tem a finalidade de renderizar textos na tela para auxiliar o usuário a manusear as funcionalidades e apresentar informações relevantes da ferramenta. Ela não apresenta nenhum atributo, mas apenas métodos para renderizar informações textuais na tela a respeito do nome da ferramenta, da ajuda no manuseio, dos desenvolvedores, das mensagens, entre outras. Mantém um relacionamento com as classes ambienteOpenGL e operacoesDaFerramenta,

respectivamente, para a renderização dos textos e a escolha das operações referentes a cada um deles.

Figura 6: Diagrama de classes.



Fonte: Própria.

3.3 ESTRUTURA DO ARQUIVO DE ENTRADA

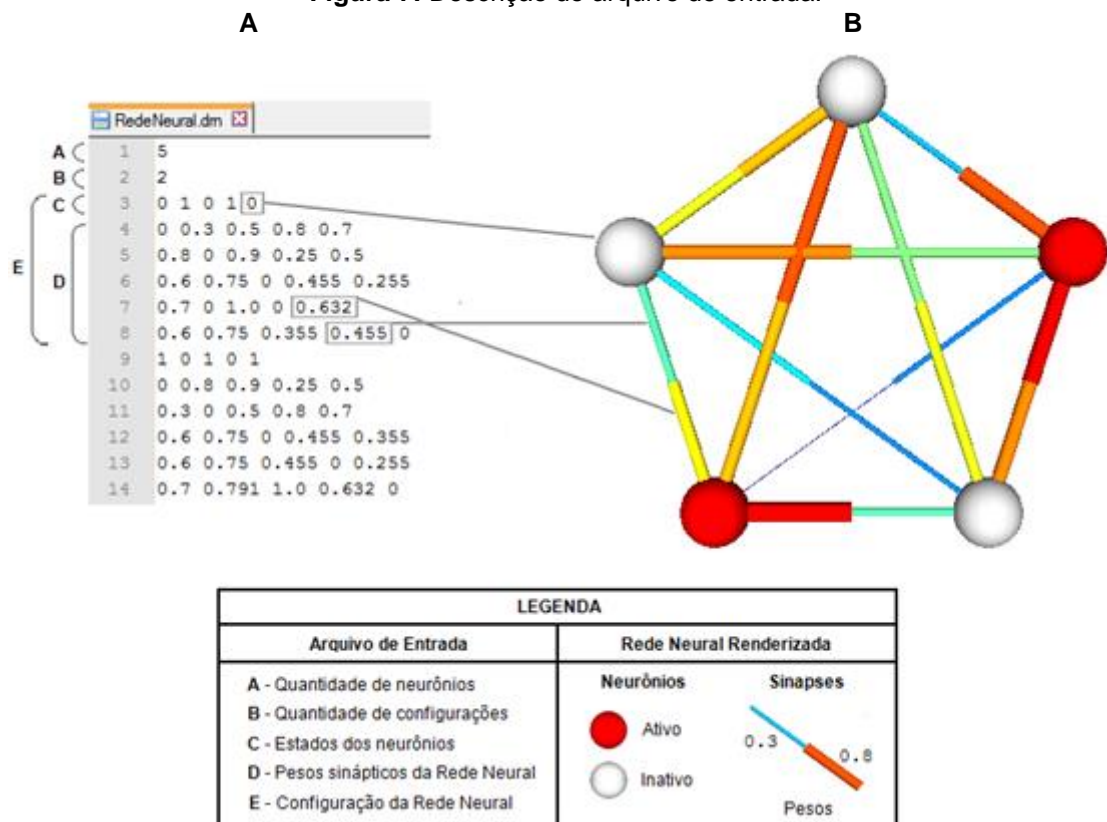
Para o desenvolvimento da ferramenta, inicialmente foi necessário realizar um estudo básico sobre RNs (apresentado no Capítulo 2) com a finalidade de entender o seu funcionamento, identificar quais tipos de estruturas de dados seriam utilizadas para representá-las como entradas do sistema e, principalmente, determinar a topologia em que elas seriam renderizadas.

Como resultado deste estudo, neste trabalho uma RN será representada pela quantidade de neurônios que a compõem, quantidade de configurações e todas as suas configurações ao longo do tempo. A quantidade de neurônios N é armazenada em uma variável do tipo inteiro assim como a quantidade de configurações T da rede. Cada configuração é constituída de um vetor de ativação e de uma matriz de plasticidade sináptica sendo armazenados, respectivamente, em um vetor e uma matriz do tipo ponto flutuante, ambos de ordem N .

Visando a leitura desses dados pela ferramenta, foi definido um arquivo de entrada no padrão ASCII especificamente para inseri-los e, conseqüentemente, renderizá-los tridimensionalmente. A extensão criada para o arquivo de entrada foi a *drn*, acrônimo para dinâmicas de redes neurais. Ademais, foi escolhida a topologia altamente conectada para representação da RN em três dimensões.

Com o intuito de facilitar a compreensão da estrutura do arquivo de entrada, a Figura 7 apresenta uma ilustração deste arquivo.

Figura 7: Descrição do arquivo de entrada.



Fonte: Própria.

A Figura 7A esboça um exemplo simples do arquivo de entrada. Suas duas primeiras linhas descrevem, respectivamente, a quantidade de neurônios da RN e a quantidade de configurações desta rede ao longo do tempo. Neste caso, o usuário deseja visualizar uma RN que contém cinco neurônios e duas configurações. A partir da terceira linha do arquivo, devem ser inseridas todas as configurações da RN de acordo com a quantidade informada na segunda linha.

Cada configuração possui um vetor de ativação e uma matriz de plasticidade sináptica que servem para representar, respectivamente, os estados de ativação dos neurônios e as configurações sinápticas da rede. Como é apresentado na legenda da Figura 7, foram considerados dois estados possíveis para um neurônio: ativo (1.0) e inativo (0.0). As configurações sinápticas equivalem aos pesos sinápticos das conexões entre os neurônios. A Figura 7B apresenta um exemplo da renderização da primeira configuração da RN para que se tenha a ideia de sua representação 3D.

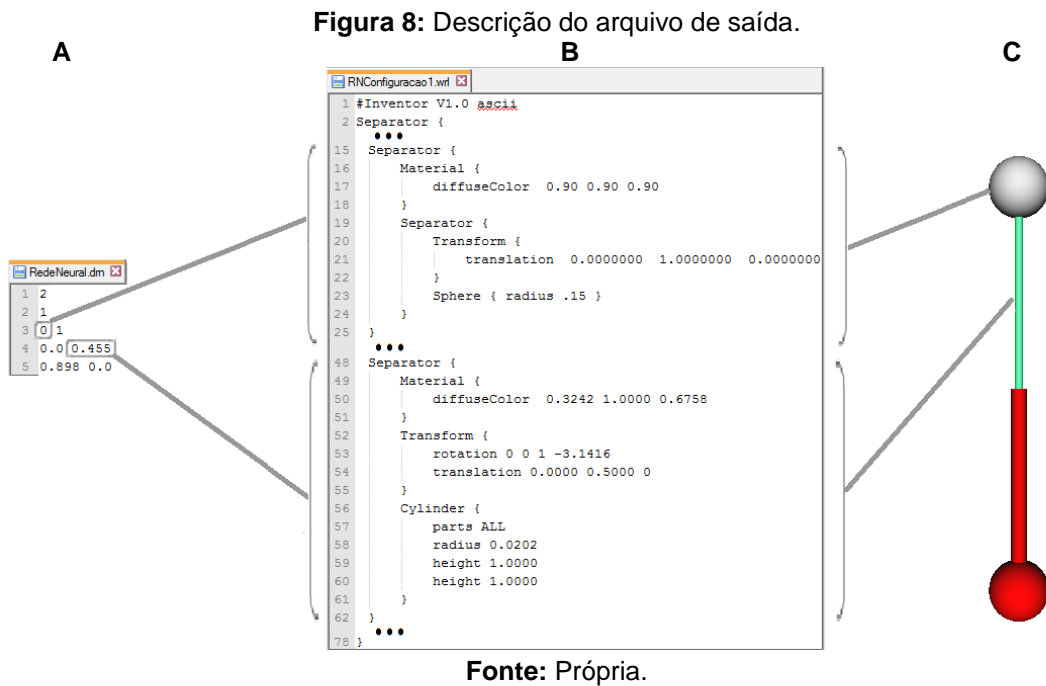
Vale salientar que a dimensão tanto do vetor de ativação quanto da matriz de plasticidade sináptica é baseada na quantidade de neurônios da RN. Neste caso, como a rede possui cinco neurônios, então o tamanho do vetor de ativação será de cinco (5) posições (Figura 7A, linhas 3 e 9) e a matriz de plasticidade sináptica será uma matriz quadrada de ordem 5 (25 conexões) para representar todas as possíveis conexões entre os neurônios. A diagonal principal dessa matriz sempre é nula, evitando a possibilidade de um neurônio ser auto conectado (Figura 7A, linhas 4-8 e linhas 10-14, primeira e segunda matriz, respectivamente).

3.4 ESTRUTURA DO ARQUIVO DE SAÍDA

Segundo Carey et al. (1997) a Virtual Reality Modeling Language (VRML), em português Linguagem de Modelagem de Realidade Virtual, é um formato de arquivo que tem o intuito de descrever mundos e objetos tridimensionais interativos. Além disso, arquivos VRML podem ser usados em diferentes áreas tais como: visualização científica e engenharia, apresentações multimídia, títulos educacionais e entretenimento, páginas da web e mundos virtuais compartilhados.

A extensão do arquivo VRML é *wrl*, acrônimo para *world*, que significa mundo em português. Com o objetivo de expandir as possibilidades de visualização das RNs por meio de outras ferramentas de visualização de modelos 3D e até mesmo

pela web, a ferramenta apresentada neste trabalho permite a exportação das configurações das RNs presentes no arquivo de entrada para arquivos individuais *wrl*. A Figura 8 apresenta o processo de exportação da configuração de uma RN para um arquivo VRML e, em seguida, sua renderização por meio da ferramenta *view3dscene*.



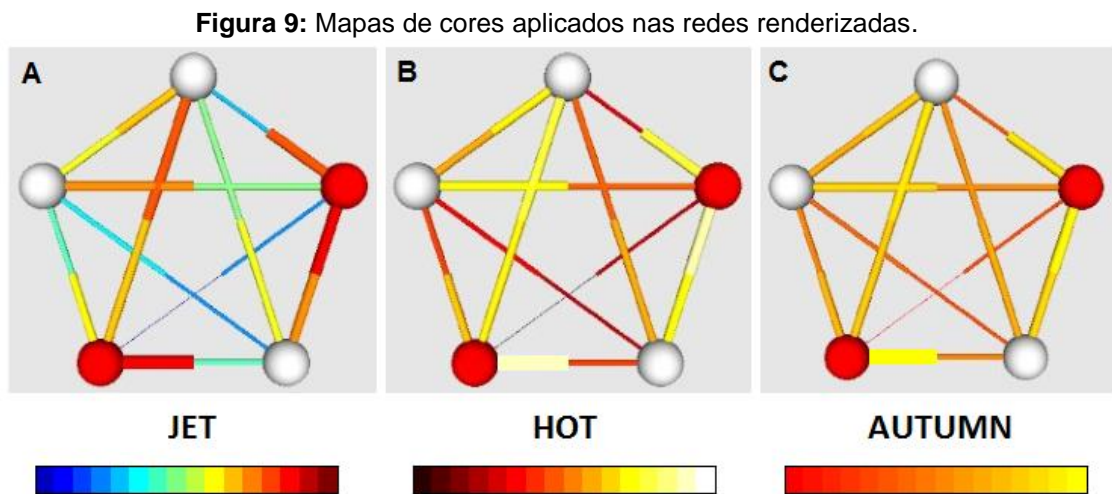
O arquivo de entrada apresentado na Figura 8A representa uma RN simples contendo somente uma configuração. A configuração é exportada para um arquivo de saída *wrl* convertendo-se os valores presentes tanto no vetor de ativação quanto na matriz de plasticidade sináptica para suas codificações equivalentes na linguagem de modelagem de realidade virtual (Figura 8B). Após a exportação, é possível utilizar uma ferramenta de visualização de arquivos *wrl* para renderizar a imagem (Figura 8C).

3.5 MAPAS DE CORES

Mapas de cores, também conhecidos como tabelas de cores, são geralmente representados por uma matriz de ordem $n \times 3$ de números reais, onde cada coluna representa um canal de cor variando de 0.0 a 1.0. A primeira, segunda e terceira colunas armazenam os valores referentes, respectivamente, à cor vermelha (Red), à

cor verde (Green) e à cor azul (Blue). Eles são conhecidos como valores RGB e servem para indicar a intensidade de uma cor em particular (PYNGL, 2015).

Com o intuito de visualizar as configurações sinápticas da RN em diferentes colorações, foram criados três mapas de cores baseados nos mapas de cores utilizados em MATLAB (MATHWORKS, 2015). Os mapas de cores escolhidos foram o Jet, o Hot e o Autumn. A Figura 9 mostra três exemplos de redes neurais que foram renderizadas utilizando-se estes três mapas de cores.



Fonte: Própria.

No mapa de cores Jet, as cores variam do azul mais escuro (cores frias) até o vermelho mais escuro (cores quentes), servindo para indicar se a intensidade da conexão entre dois neurônios é fraca ou forte, respectivamente. Como é possível observar na Figura 9A, quanto menor for o valor do peso de uma conexão (peso sináptico ou sinapse), a tonalidade da cor azul é mais escura e, dessa maneira, é considerada fraca.

A variação de cor no mapa de cores Hot acontece do vermelho mais escuro (próximo do preto) até o amarelo mais claro indicando, respectivamente, se a conexão entre dois neurônios é fraca ou forte. A Figura 9B mostra que quanto maior for a intensidade da conexão entre um neurônio e outro, as cores são mais claras tendendo para o amarelo. Caso contrário, as cores são mais escuras tendendo para o vermelho próximo de preto.

Finalmente, o mapa de cores Autumn é o mais simples de todos, pois apresenta somente tonalidades de cores que variam da laranja escura ao amarelo escuro. A Figura 9C denota que quanto mais laranja é uma conexão, mais fraca ela

é. Por outro lado, quanto mais próximo do amarelo é a cor de uma conexão, mais forte ela será.

Contudo, tendo em vista que os mapas de cores são importantes para a visualização de objetos gráficos, tais como as RNs renderizadas pela ferramenta, os exemplos apresentados na Figura 9 provam que seu uso torna mais intuitivo o estudo da dinâmica de redes neurais.

3.6 IMPLEMENTAÇÃO

Para o desenvolvimento da ferramenta, foi necessário estudar e aprender a linguagem de programação C++ (OVERLAND, 2011) visando à programação utilizando-se o paradigma Orientado a Objetos (OO). C++ foi a linguagem escolhida por ser de propósito geral, flexível, multiparadigma e, principalmente, permitir ao programador desenvolver ferramentas gráficas com o auxílio das bibliotecas do OpenGL e da GLUT (COHEN e MANSSOUR, 2006). Ademais, devido ao fato do paradigma OO (BUENO, 2003) ser o mais difundido atualmente no mercado de desenvolvimento de software, ele foi escolhido.

Dessa maneira, o ambiente de desenvolvimento Dev-C++ na versão 4.9.9.2 (BLOODSHED, 2015) associado com o compilador Mingw (MINGW, 2015) foi escolhido para implementação por apresentar uma interface simples, permitir a programação em C++ e, especialmente, ter sido o único ambiente onde se conseguiu configurar as bibliotecas OpenGL e GLUT. Depois de configurado o ambiente de desenvolvimento, foram implementadas as seguintes classes da o desenvolvimento da ferramenta: `operacoesDaFerramenta`, `ambienteOpenGL`, `mapaDeCores` e `telasDaFerramenta`.

A biblioteca OpenGL foi utilizada para renderização das imagens da RNs em 3D. As RNs renderizadas são constituídas de elementos geométricos simples, tais como esferas e cilindros que representam seus neurônios e suas conexões, respectivamente. A GLUT foi utilizada para criar o menu da ferramenta com o intuito de manipular todas as suas operações. Os mapas de cores foram criados com o auxílio do MATLAB. Finalmente, para conversão das configurações da RN em arquivos wrl foi criada um método dentro da classe `operacoesDaFerramenta` para este fim.

4 A FERRAMENTA DyRENE VISUAL

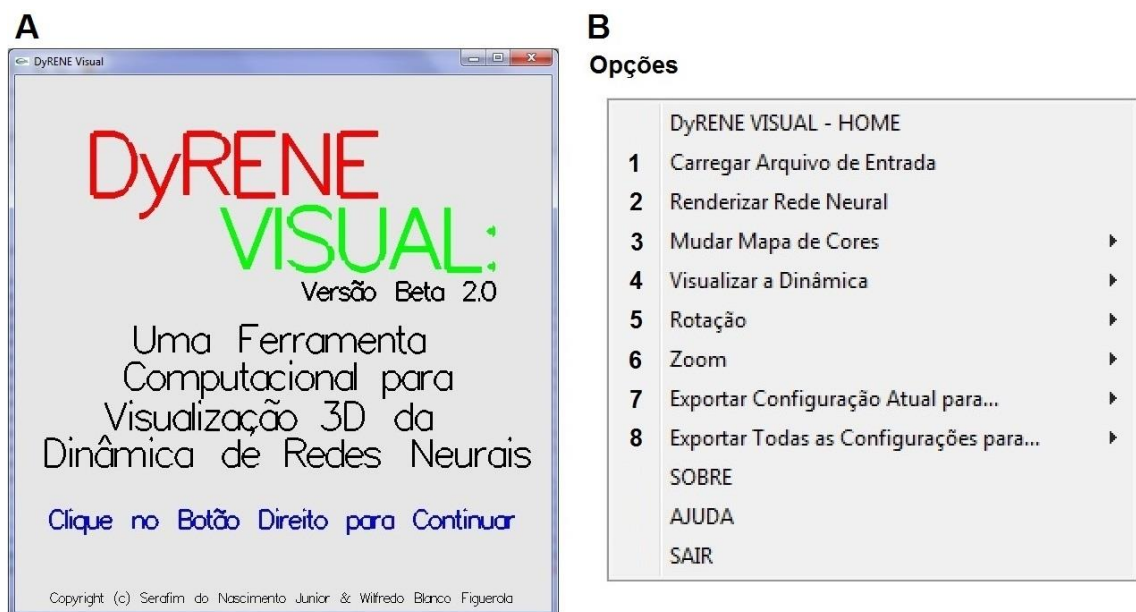
Neste capítulo a ferramenta desenvolvida é apresentada e todas as suas funcionalidades são claramente explicadas.

4.1 DyRENE VISUAL

A ferramenta desenvolvida neste trabalho foi batizada de DyRENE Visual, acrônimo para **D**inâmicas de **R**edes **N**eurais. O **i** foi substituído por **y** pelo fato da palavra dinâmica em Inglês ser “*Dynamics*”. Ao se executar a ferramenta, a tela principal é apresentada ao usuário mostrando informações básicas, tais como: o nome, a versão, o objetivo e os criadores da ferramenta. Além disso, a ferramenta orienta o usuário a clicar o botão direito do mouse para visualizar o menu e, dessa maneira, permitir que ele selecione as opções equivalentes as operações que deseja analisar.

Para entender melhor como acontece este processo, as Figuras 10A e 10B apresentam, respectivamente, a tela principal da ferramenta e o menu com as opções de operações a serem realizadas pelo usuário.

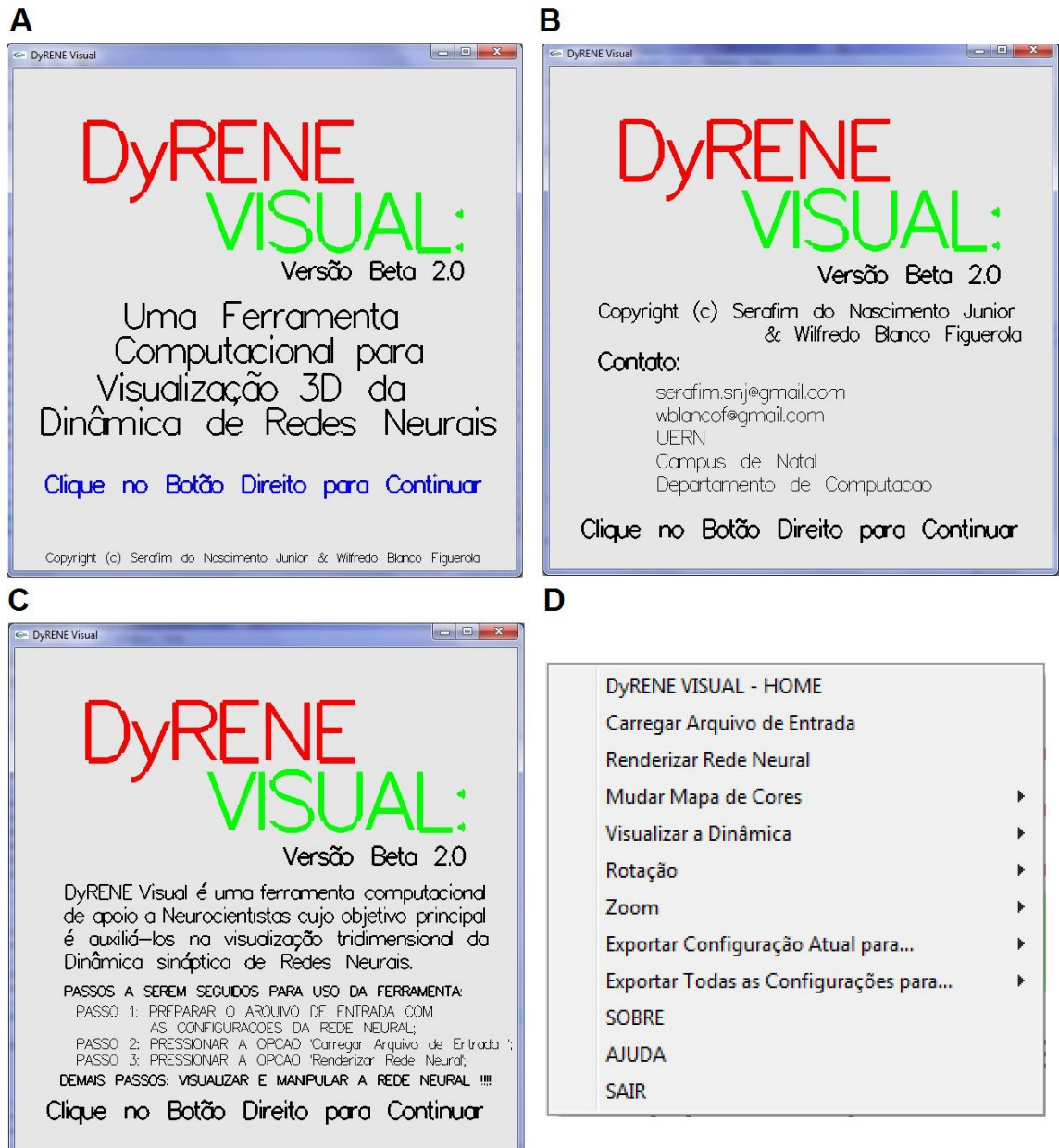
Figura 10: Apresentação da ferramenta.
A) Tela principal. B) Menu da ferramenta.



Fonte: Própria.

As opções mais simples não serão apresentadas nas seções subsequentes por somente terem a finalidade de fornecer informações ao usuário sobre a ferramenta. São elas: DyRENE VISUAL - HOME, SOBRE, AJUDA e SAIR, onde suas telas são, respectivamente, apresentadas na Figura 11.

Figura 11: Telas informativas da ferramenta.
A) Tela principal. B) Tela sobre as informações da ferramenta.
C) Tela de ajuda. D) Opção para sair.



Fonte: Própria.

Os tópicos a seguir explicam cada uma dessas funcionalidades simples da ferramenta apresentadas na Figura 11.

- DyRENE VISUAL - HOME: esta opção já foi explicada no começo desta seção e, mais uma vez, ela somente apresenta ao usuário as informações básicas da ferramenta como pode ser visto na Figura 11A;
- SOBRE: esta opção apresenta informações referentes ao desenvolvimento da ferramenta incluindo os contatos de seus idealizadores como é mostrado na Figura 11B;
- AJUDA: apresenta ao usuário a missão da ferramenta e ajuda-o a como proceder para manuseá-la (Figura 11C); e
- SAIR: esta é a mais simples de todas, pois permite que o usuário saia da ferramenta (Figura 11D).

Nas próximas seções, são apresentadas as opções mais importantes da ferramenta.

4.2 ARQUIVO DE ENTRADA

O arquivo de entrada é muito importante para a ferramenta, pois ele permite a inserção da descrição de uma determinada RN e de todas as suas configurações ao longo do tempo para que ela possa ser renderizada em 3D pela ferramenta. É o arquivo padrão para inserção dos dados das RNs a serem visualizadas e foi nomeado como RedeNeural.drn, sendo apresentado na Figura 12.

Figura 12: Arquivo RedeNeural.drn.

	Neurônios	Qtde de configurações
1	8	
2	9	
3	0 1 0 1 0 1 0 1	
4	0 0.2 0.9 0.123 0.53 0.58 0.9 0.1	Configuração 1
5	0.5 0 0.21 0.45 0.87 0.234 0.388 0.9	
6	0.3 0.325 0 0.97 0.75 0.339 0.421 0.678	
7	0.29 0 0.475 0 0.232 0.98 0.298 0.677	
8	0.231 0.25 0.55 0.121 0 0.059 0.106 0.876	
9	0.9 0.25 0.69 0.255 0.5 0 0.79 0.9	
10	0.225 0.69 0.25 0.4 0.29 0.45 0 0.899	
11	0.777 0.234 0.786 0.222 0.213 0.65 0.9 0	
	...	
75	1 1 0 0 1 1 0 0	Configuração 9
76	0 0.3 0.5 0.8 0.7 0.9 0.9 0.3	
77	0.8 0 0.9 0.25 0.5 0.9 0.8 0.3	
78	0.6 0.75 0 0.455 0.255 0.777 0.876 0.3	
79	0.7 0 0.97 0 0.632 0.499 0.677 0.3	
80	0.6 0.75 0.355 0.455 0 0.050 0.786 0.3	
81	0.6 0.75 0.23 0.455 0.255 0 0.555 0.3	
82	0.586 0.3 0.5 0.8 0.7 0.9 0 0.3	
83	0.586 0.3 0.5 0.8 0.7 0.9 0.3 0	

Fonte: Própria.

Para que o arquivo seja lido pela ferramenta, o usuário deve respeitar as seguintes regras:

- Inserir na primeira linha a quantidade de neurônios que constituem a RN;
- Inserir na segunda linha a quantidade de configurações da RN;
- Inserir nas demais linhas do arquivo todas as configurações da RN sempre colocando o vetor de ativação antes da matriz de plasticidade sináptica;

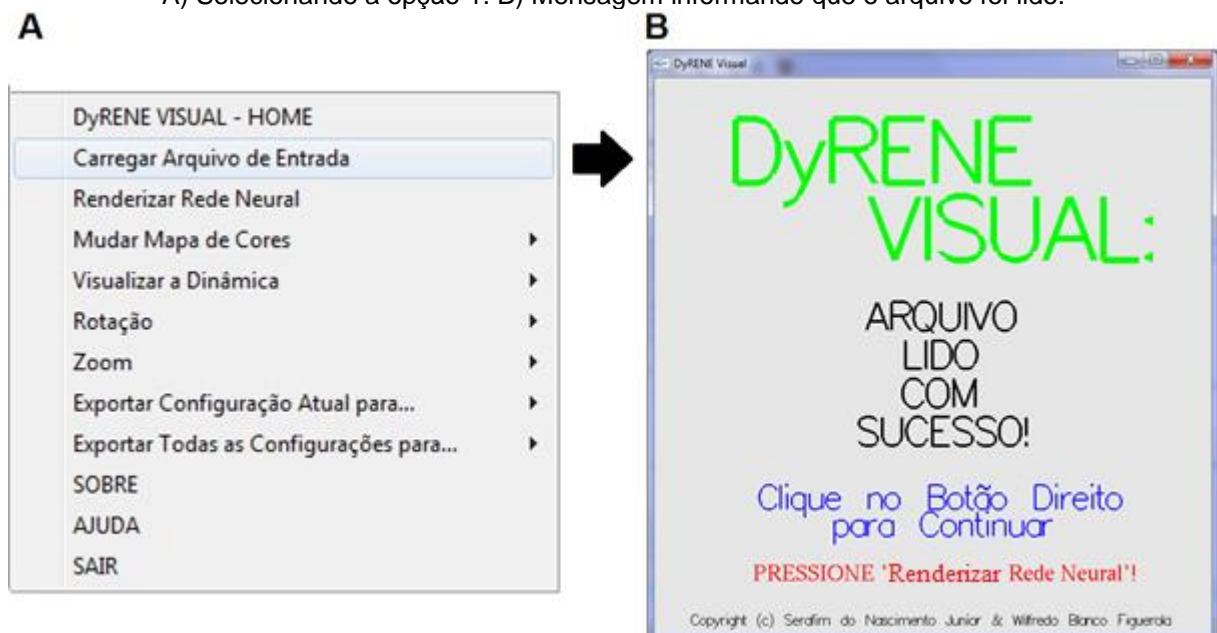
A Figura 12 apresenta um bom exemplo para guiar o usuário quanto a inserção dos dados da RN no arquivo de entrada. Os dados da RN contidos no arquivo da Figura 12 foram utilizados como um dos testes para renderizar as imagens das redes apresentadas neste capítulo nas seções a seguir.

4.3 LEITURA E RENDERIZAÇÃO DA REDE NEURAL

Esta é uma das opções mais importantes, pois é por meio dela que o arquivo de entrada, apresentado na seção anterior, é carregado pela ferramenta para que seja, posteriormente, possível a renderização em 3D das configurações da RN. A Figura 13 mostra o que acontece quando o usuário seleciona esta opção.

Figura 13: Leitura do arquivo de entrada.

A) Selecionando a opção 1. B) Mensagem informando que o arquivo foi lido.

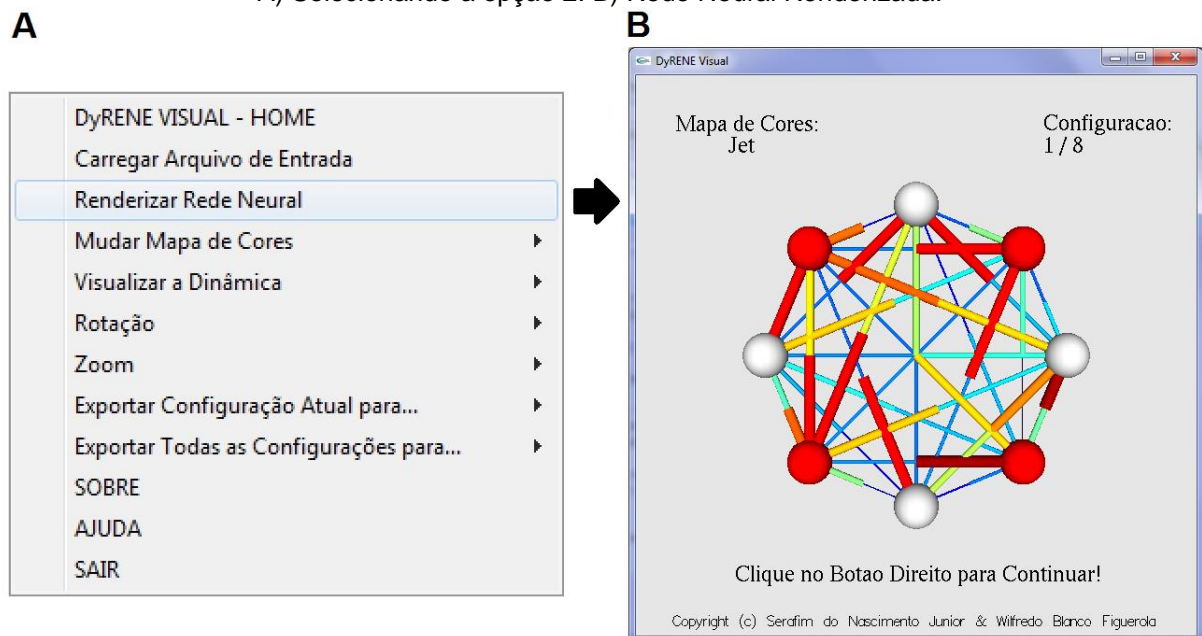


Fonte: Própria.

Ao selecionar esta opção (Figura 13A), uma mensagem é informada ao usuário afirmando que o arquivo de entrada foi lido com sucesso e que ele pode selecionar a opção 2 (Renderizar Rede Neural) para renderizar a RN (Figura 13B).

Tendo em vista que os dados das configurações da RN se encontram na memória, o usuário pode, então, selecionar a opção 2 (Figura 14A) para permitir que esses dados sejam renderizados na tela em 3D por meio da biblioteca OpenGL. A Figura 14B mostra uma RN com oito (8) neurônios que contém oito (8) configurações no tempo, sendo que a primeira configuração é renderizada como é mostrado no texto da parte superior direita da tela.

Figura 14: Renderização da rede neural.
A) Selecionando a opção 2. B) Rede Neural Renderizada.



Fonte: Própria.

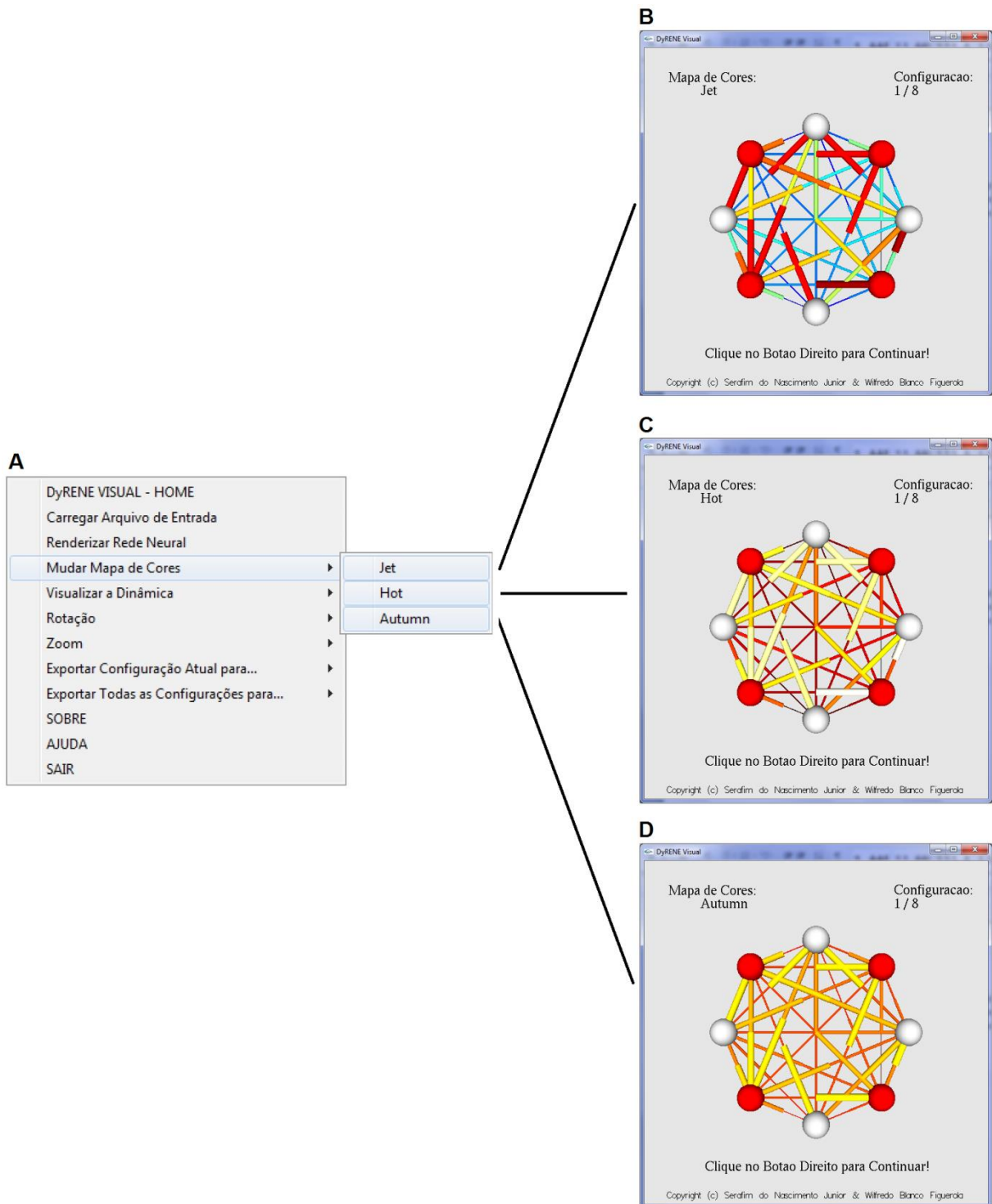
4.4 MUDANÇA DOS MAPAS DE CORES

Com o objetivo de visualizar as configurações da RN com diversidade de cores, conhecida no âmbito da visualização científica, foram implementados três diferentes mapas de cores. São eles: o Jet, o Hot e o Autumn, sendo explicados detalhadamente na Seção 3.4 do Capítulo 3. A Figura 15 ilustra um exemplo de aplicação dos três mapas de cores em uma determinada configuração da RN.

Para mudar o mapa de cores da configuração da RN, o usuário só precisa selecionar a opção 3 da ferramenta e escolher qual deles preferir aplicar na rede. A Figura 15A esboça como proceder para realizar o que foi descrito anteriormente.

Figura 15: Aplicação dos mapas de cores.

A) Selecionando os mapas de cores. B) Jet. C) Hot. D) Autumn.

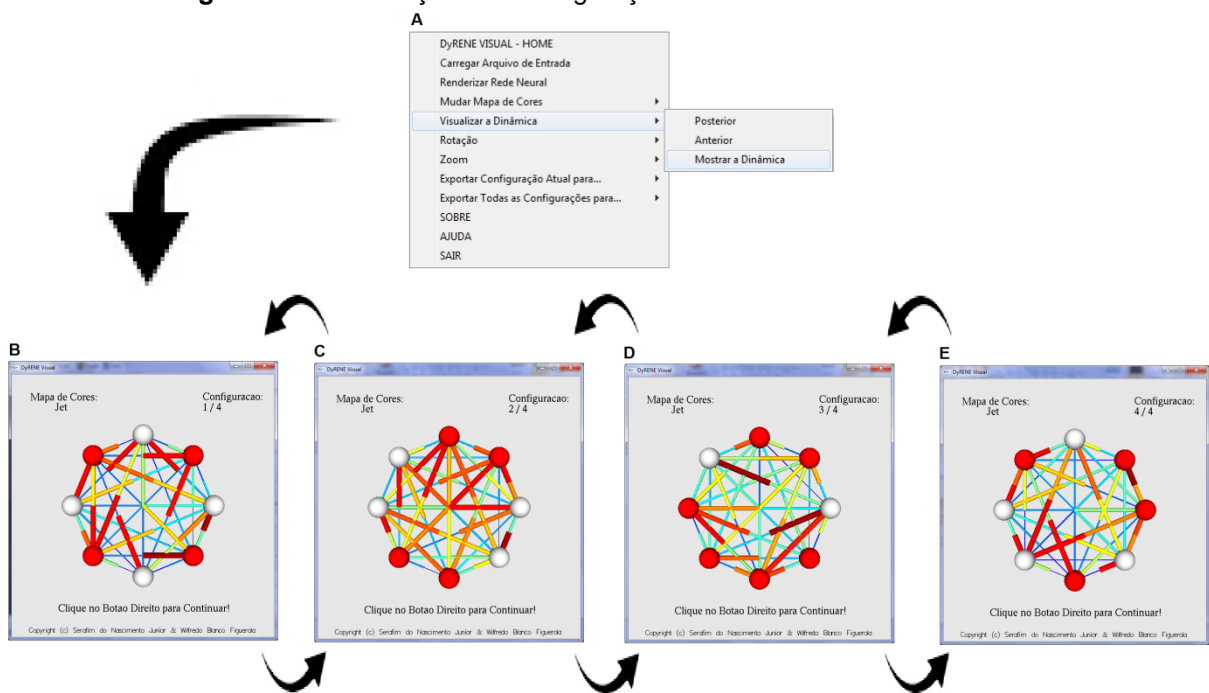


Fonte: Própria.

4.5 VISUALIZAÇÃO DA DINÂMICA DA REDE NEURAL

Esta opção é a mais importante para este trabalho, pois possibilita a visualização de cada configuração da RN assim como de sua dinâmica, dessa maneira, cumprindo com o objetivo geral deste trabalho. A navegação permite a criação de uma animação que possibilita ao usuário visualizar a dinâmica temporal da RN. A Figura 16 apresenta como o usuário deve proceder para visualizar as configurações da RN assim como sua dinâmica.

Figura 16: Visualização das configurações e da dinâmica da rede neural.



Fonte: Própria.

A opção 4 permite ao usuário realizar três operações fundamentais para a visualização das configurações da RN. São elas: Posterior, Anterior e Mostrar a Dinâmica. As operações denominadas Posterior e Anterior permitem, respectivamente, a renderização da configuração seguinte (setas da esquerda para a direita, por exemplo, partindo da Figura 16B até a Figura 16E) e anterior (setas da direita para a esquerda, por exemplo, partindo da Figura 16E até a Figura 16B) da rede. A mais importante é a operação Mostrar a Dinâmica, pois ela permite a exibição de uma animação da dinâmica da RN em estudo, iniciando da primeira configuração e apresentado todas as configurações da rede até atingir a última configuração.

4.6 ROTAÇÃO NOS EIXOS X E Y

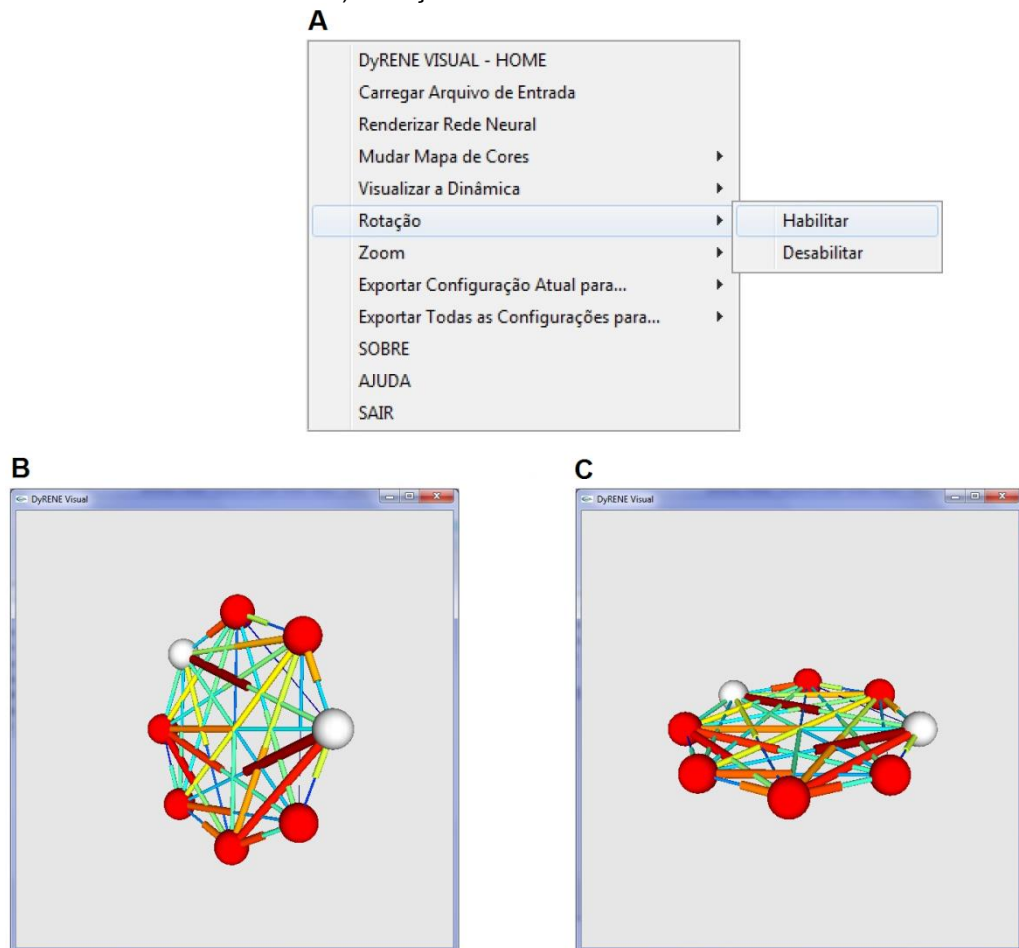
Como o nome já diz, a opção de rotação permite ao usuário rotacionar a rede neural em torno dos eixos x e y. Para realizar esta operação, o usuário deve habilitar esta função no menu da ferramenta e, em seguida, utilizar as teclas de movimentação do cursor presentes no teclado. A Figura 17 apresenta como o usuário deve proceder para rotacionar a RN.

Figura 17: Rotação nos eixos x e y.

A) Opção para habilitar a operação de rotação na rede neural.

B) Rotação ao redor do eixo y.

C) Rotação ao redor do eixo x.



Fonte: Própria.

Como poder ser visto na Figura 17A, o usuário deve habilitar a operação de rotação para poder movimentar a RN renderizada em torno dos eixos x e y. A Figura 17B ilustra a rotação em torno do eixo y e assim como a Figura 17C mostra a rotação em torno do eixo x. Para desabilitar a rotação e voltar para a tela de renderização

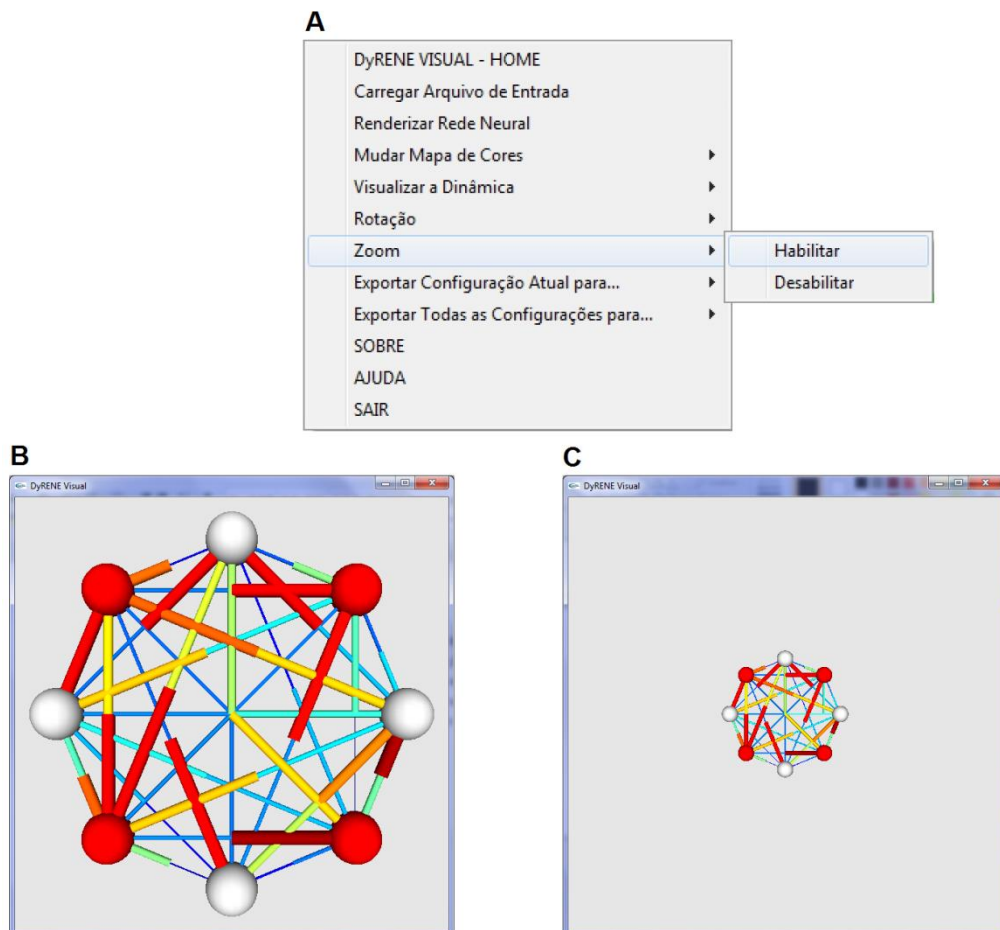
principal da ferramenta, o usuário tem que selecionar a opção Desabilitar, como também é possível observar na Figura 17A.

4.7 ZOOMING IN/OUT

A operação de Zooming também foi implementada para possibilitar a visualização da rede neural em diferentes distâncias. Após habilitar esta operação, o usuário pode efetuar as operações de *zoom out* e *zoom in* pressionando, respectivamente, as teclas *page up* e *page down* do teclado. A Figura 18 apresenta, detalhadamente, como proceder para realizar estas operações.

Figura 18: Operações de zooming in/out.

A) Opção para habilitar a operação de zooming na rede neural. B) Zoom out. C) Zoom in.



Fonte: Própria.

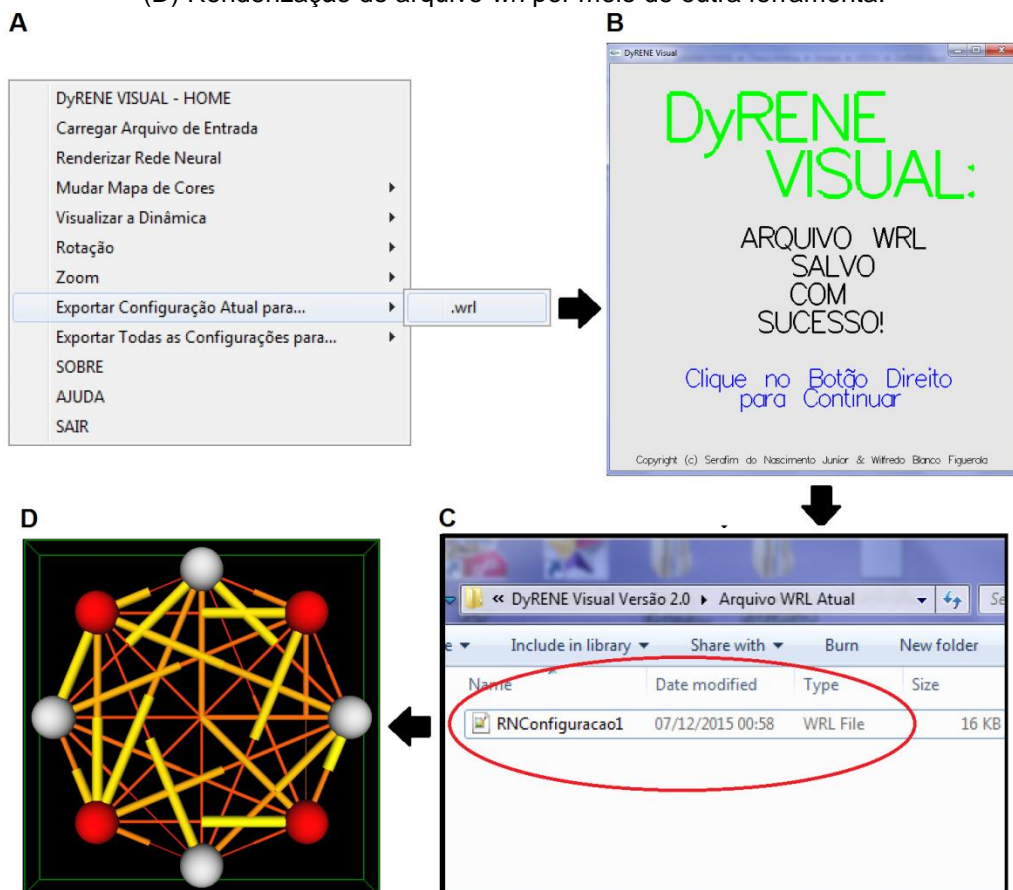
Primeiramente, o usuário precisa habilitar a operação de zooming (Figura 18A). Em seguida, basta brincar de movimentar a rede para frente (Figura 18B) e para trás (Figura 18C).

4.8 EXPORTAÇÃO DAS CONFIGURAÇÕES DA RN PARA ARQUIVOS WRL

Com o interesse de visualizar as configurações da RN através de outras ferramentas de visualização de modelos 3D, foram implementadas duas funcionalidades (opção 7 e 8) para exportar as configurações da rede presentes no arquivo de entrada para arquivos com a extensão *wrl*. A opção 7 permite que o usuário exporte a configuração atual (presente no arquivo de entrada) daquela rede neural que está renderizada na tela para um arquivo de extensão *wrl*.

A Figura 19 apresenta como o usuário deve proceder para realizar esta operação e o que ele pode fazer para visualizar o arquivo de saída de extensão *wrl*.

Figura 19: Exportação das configurações da rede neural para arquivos *wrl*.
 (A) Opção para exportar a configuração atual da RN para um arquivo *wrl*.
 (B) Mensagem de sucesso. (C) Arquivo *wrl* gerado.
 (D) Renderização do arquivo *wrl* por meio de outra ferramenta.



Fonte: Própria.

Para converter uma determinada configuração da RN em um arquivo de saída *wrl*, o usuário precisa ter executado na ferramenta a operação que permite a leitura

do arquivo de entrada RedeNeural.drn. Caso o arquivo de entrada já tenha sido lido, o usuário precisa selecionar a opção 7 do menu para exportar a configuração corrente que ele está visualizando em um arquivo *wrl*, como é visto na Figura 19A.

Após a exportação, a ferramenta apresenta uma mensagem (Figura 19B) informando o sucesso na conversão e o arquivo *wrl* é salvo no diretório denominado Arquivo WRL Atual (Figura 19C). Finalmente, o usuário pode visualizar arquivo *wrl* por meio de alguma ferramenta de visualização de modelos 3D. Neste exemplo, a ferramenta utilizada para gerar a imagem da Figura 19D foi a *view3dscene* na versão 3.15.0 (CASTLE GAME ENGINE, 2015) permitindo, assim, validar esta funcionalidade da ferramenta.

Diferentemente da opção 7, a opção 8 permite que o usuário exporte todas as configurações da RN para arquivos com a extensão *wrl*. Para efetuar esta operação na ferramenta, o usuário deve selecionar a opção 8. Devido ao processo ser praticamente o mesmo da Figura 19, não foi criada uma figura para exemplificação.

4.9 CONCLUSÕES SOBRE A FERRAMENTA DyRENE VISUAL

Como é possível perceber no decorrer deste capítulo, a ferramenta desenvolvida neste trabalho apresentou um bom desempenho quanto à execução de suas funcionalidades, pois cumpre com os objetivos definidos para a sua implementação. Após o desenvolvimento da ferramenta foram feitos vários testes com RNs hipotéticas para avaliar o seu funcionamento, dentre os quais alguns deles foram apresentados neste capítulo assim como no decorrer deste trabalho para exemplificação. Portanto, é possível concluir que a ferramenta DyRENE Visual pode ser utilizada para visualização 3D da dinâmica de RNs.

5 CONCLUSÕES

O objetivo principal do trabalho foi propor uma ferramenta de visualização científica para auxiliar neurocientistas no tocante a visualização da dinâmica de redes neurais em três dimensões. O motivo que norteou o seu desenvolvimento, foi a carência de ferramentas computacionais que possibilitem a visualização de como se comportam as sinapses e os neurônios ao longo tempo.

Com o intuito de obter os requisitos necessários para o desenvolvimento da ferramenta, foi necessário realizar uma pesquisa bibliográfica sobre as suas principais áreas do conhecimento, a Computação Gráfica e a Neurociência. Ademais, visando tentar encontrar ferramentas similares, foram realizadas pesquisas na Internet. Algumas ferramentas que auxiliam na visualização de fenômenos do cérebro foram encontradas, mas nenhuma delas se assemelha a DyRENE Visual nem possibilitam realizar as operações que ela oferece.

A implementação da ferramenta foi realizada no ambiente de desenvolvimento Dev-C++ utilizando-se a linguagem C++ Orientada a Objetos e as bibliotecas OpenGL e GLUT para renderização de imagens e criação do menu, respectivamente. Para isto, foi necessário seguir a metodologia elaborada para o seu desenvolvimento. Como resultados, a ferramenta foi apresentada por meio de exemplos de cada uma de suas funcionalidades. A validação foi feita por meio de testes com redes neurais hipotéticas que foram criadas durante a fase de desenvolvimento.

Sendo assim, baseando-se neste relato, é possível concluir que a ferramenta DyRENE Visual é uma ferramenta original que possibilita a exibição da dinâmica de redes neurais em três dimensões além de outras funcionalidades adicionais e, portanto, pode ser usada por neurocientistas em suas pesquisas.

5.1 TRABALHOS FUTUROS

Como sugestão para trabalhos futuros, pode-se citar a implementação de mais funcionalidades para a ferramenta, tais como: exportação das configurações da rede para outras extensões, operações para captura de tela e salvamento em diferentes formatos de arquivo de imagem, geração de arquivos de vídeo da

dinâmica, entre outros. Ademais, também é sugerida a implementação de uma versão desta ferramenta para a web.

REFERÊNCIAS

ANATOMY DIAGRAM. **Neuron Diagram Myelin Sheath**. Disponível em: <<http://anatomy-diagram.net/tag/neuron-human-anatomy/>>. Acesso em: 13 dez. 2015.

ASTAH. **Ferramenta para Projeto de Software Astah Community**. Disponível em: <<http://astah.net/download>>. Acesso em: 23 mai. 2015.

AZEVEDO, E.; CONCI, A. **Computação gráfica: geração de imagens**. Rio de Janeiro: Campus, 2003. 353 p.

BARBOSA, G. M. G. **Um Livro-texto para o Ensino de Projeto de Arquitetura de Software**. Campina Grande, 2009. 209 f. Dissertação (Mestrado em Ciência da Computação) - Centro de Engenharia Elétrica e Informática. Universidade Federal de Campina Grande.

BLOODSHED. **The Dev-C++ Resource Site**. Disponível em: <<http://www.bloodshed.net/dev/devcpp.html>>. Acesso em: 11 dez. 2015.

BOOCH, G.; JACOBSON, I.; RUMBAUGH, J. **The Unified Modeling Language user guide**. Redwood City: Addison Wesley Longman Publishing Co, 1999. 482 p.

BORGES, A. A. Rede neural artificial. Disponível em: <<https://www.youtube.com/watch?v=vbf4IzvXvuM>>. Acesso em: 18 dez. 2015.

BUENO, A. D. **Programação Orientada a Objeto com C++**. São Paulo: Novatec, 2003. 592 p.

BURIOL, T. M.; GEUS, K. de; SCHEER, S. Visualização Científica e Interação em Ambientes Virtuais Tridimensionais. In: SIMPÓSIO DE MÉTODOS NUMÉRICOS COMPUTACIONAIS DA UFPR, 1., 2011, Curitiba. **Resumos**. Curitiba: PPGMNE/UFPR, 2011. p. 74-75.

CAREY, R.; BELL, G.; Marrin, C. ISO/IEC 14772-1:1997 Virtual Reality Modeling Language (VRML97). Disponível em: <<http://www.vrml.org/Specifications/VRML97>>. Acesso em: 1 dez. 2015.

CASTLE GAME ENGINE. **view3dscene**. Disponível em: <http://castle-engine.sourceforge.net/view3dscene.php#section_install>. Acesso em: 7 dez. 2015.

CHOREV, E; EPSZTEIN, J.; HOUWELING, A. R.; LEE A. K.; e BRECHT, M. Electrophysiological recordings from behaving animals – going beyond spikes. **Current Opinion in Neurobiology**, Amsterdam, v. 19, n. 5, p. 513–519, out. 2009.

COHEN, M; MANSSOUR, I. H. **OpenGL: Uma abordagem prática e objetiva**. São Paulo: Novatec, 2006. 486 p.

GATTASS, M. **Material**. Disponível em: <<http://webserver2.tecgraf.puc-rio.br/~mgattass/fcg/material/>>. Acesso em: 9 dez. 2015.

GONZALEZ, R. C.; WOODS, R. E. **Digital Image Processing**: Third Edition. New Jersey: Pearson Education, Inc., 2008. 976 p.

GUEDES, G. T. A. **UML 2**: Guia Prático. São Paulo: NOVATEC, 2014. 192 p.

HOPFIELD, J. J. Neural networks and physical systems with emergent collective computational abilities. In: Proceedings of the National Academy of Sciences of the United States of America, n. 8, v. 79, 1982, USA. **Article**. USA: Proceedings of the National Academy of Sciences, 1982. p. 2554-2558.

IEEE e ISO/IEC. **Systems and Software Engineering - Recommended Practice for Architectural Description of Software-Intensive Systems**. ISO/IEC 42010 IEEE Std 1471-2000 Primeira Edição 2007-07-15, páginas c1–24, Julho 2007.

IETF. **ASCII format for Network Interchange**. Disponível em <<https://tools.ietf.org/html/rfc20>>. Acesso em: 25 mar. 2015.

JOHNSON, C. R.; HANSEN, C. D. **The Visualization Handbook**. United States of America: Elsevier, 2005. 962 p.

LLNL. **Brain Visualization**. Disponível em: <<https://wci.llnl.gov/simulation/computer-codes/visit/gallery/gallery-16>>. Acesso em: 29 abr. 2015.

LUDWIG Jr., O.; COSTA, E. M. M. **Redes Neurais**: fundamentos e aplicações com programas em C. Rio de Janeiro: Ciência Moderna Ltda, 2007. 125 p.

MASSON, T. **History of Computer Graphics (CG)**. Disponível em: <<http://www.cs.cmu.edu/~ph/nyit/masson/history.htm>>. Acesso em: 17 mar. 2015.

MATHWORKS. **MATLAB**: The Language of Technical Computing. Disponível em: <<http://www.mathworks.com/help/matlab/>>. Acesso em: 11 dez. 2015.

MCCORMICK, B. H. et al (ed), **Visualization in Scientific Computing**, Computer Graphics 21, 6 (November 1987), ACM SIGGRAPH: New York, p. 3.

MINGW. **MinGW**: Minimalist GNU for Windows. Disponível em: <<http://www.mingw.org/>>. Acesso em: 11 dez. 2015.

NEURAL POWER. **A typical artificial neural network**. Disponível em: <<http://www.neuralpower.com/technology.htm>>. Acesso em: 18 dez. 2015.

OVERLAND; B. R. **C++ without fear**: a beginner's guide that makes you feel smart. Boston: Pearson Education, Inc., 2011.

PYINGL. **Color maps**. Disponível em: <https://www.pyngl.ucar.edu/Graphics/color_maps.shtml>. Acesso em: 7 dez. 2015.

ROQUE, A. **Introdução à Neurociência Computacional**. Disponível em: <http://sisne.org/ Disciplinas/PosGrad/IntrodNeuroComput/aula0_parte1.pdf>. Acesso em: 29 abr. 2015.

UBALT. **Biological neural network**. Disponível em: <<http://home.ubalt.edu.htm>>. Acesso em: 29 abr. 2015.

UML. **Unified Modeling Language™ (UML®) Resource Page**. Disponível em: <<http://www.uml.org/>>. Acesso em: 23 mai. 2015.

VENTURA, D. F. Um Retrato da Área de Neurociência e Comportamento no Brasil. In: Psicologia: teoria e pesquisa, v. 26, n. especial, 2010, Brasília. **Artigos**. p. 123-129. Disponível em <<http://www.scielo.br/pdf/ptp/v26nspe/a11v26ns.pdf>>. Acesso em: 15 out. 2015.

WIKIPEDIA. **Action potential**. Disponível em: <http://en.m.wikipedia.org/wiki/Action_potential>. Acesso em: 26 abr. 2015.

WIKIPEDIA. **Neural oscillation**. Disponível em: <http://en.m.wikipedia.org/wiki/Neural_oscillation>. Acesso em: 26 abr. 2015.

GLOSSÁRIO

Renderização* - é o processo por meio do qual é possível gerar (sintetizar) imagens digitais a partir de modelos 2D ou 3D utilizando programas de computador.

Zooming (zoom in/out)** - fazer a imagem de algo ou alguém se apresentar maior ou mais próxima ou menor e mais distante.

* WIKIPEDIA. **Rendering (computer graphics)**. Disponível em: <[https://en.wikipedia.org/wiki/Rendering_\(computer_graphics\)](https://en.wikipedia.org/wiki/Rendering_(computer_graphics))>. Acesso em: 20 dez. 2015.

** CAMBRIDGE ADVANCED LEARNERS DICTIONARY & THESAURUS. **Zoom in/out**. Disponível em: <<http://dictionary.cambridge.org/us/dictionary/english/zoom-in-out>>. Acesso em: 28 dez. 2015

